



# PERCONA

## Monitoring and Management

# Documentation

2.26.0 (February 8, 2022)

*Percona Technical Documentation Team*

*Percona LLC, © 2021*

# Table of contents

|                             |     |
|-----------------------------|-----|
| 1. Welcome                  | 5   |
| 1.1 Start here              | 6   |
| 1.2 Read more               | 6   |
| 2. Setting up               | 7   |
| 2.1 Setting up              | 7   |
| 2.2 Server                  | 9   |
| 2.3 Client                  | 52  |
| 3. Using                    | 126 |
| 3.1 Using                   | 126 |
| 3.2 User Interface          | 127 |
| 3.3 Integrated Alerting     | 133 |
| 3.4 Backup and Restore      | 145 |
| 3.5 Query Analytics         | 158 |
| 3.6 DBaaS                   | 171 |
| 3.7 Security Threat Tool    | 188 |
| 4. How to                   | 190 |
| 4.1 How to                  | 190 |
| 4.2 Configure               | 191 |
| 4.3 Manage users            | 204 |
| 4.4 Upgrade                 | 208 |
| 4.5 Secure                  | 210 |
| 4.6 Optimize                | 212 |
| 4.7 Annotate                | 213 |
| 4.8 Render dashboard images | 214 |
| 4.9 Extend Metrics          | 216 |
| 4.10 Troubleshoot           | 218 |
| 5. Details                  | 221 |
| 5.1 Details                 | 221 |
| 5.2 Architecture            | 222 |
| 5.3 UI components           | 226 |
| 5.4 Dashboards              | 230 |
| 5.5 Commands                | 352 |
| 5.6 API                     | 380 |
| 5.7 VictoriaMetrics         | 383 |
| 5.8 Glossary                | 385 |

|                                                                                           |     |
|-------------------------------------------------------------------------------------------|-----|
| 6. FAQ                                                                                    | 387 |
| 6.1 How can I contact the developers?                                                     | 387 |
| 6.2 What are the minimum system requirements?                                             | 387 |
| 6.3 How can I upgrade from version 1?                                                     | 387 |
| 6.4 How to control data retention?                                                        | 387 |
| 6.5 How are PMM Server logs rotated?                                                      | 388 |
| 6.6 What privileges are required to monitor a MySQL instance?                             | 388 |
| 6.7 Can I monitor multiple service instances?                                             | 388 |
| 6.8 Can I rename instances?                                                               | 388 |
| 6.9 Can I add an AWS RDS MySQL or Aurora MySQL instance from a non-default AWS partition? | 388 |
| 6.10 What resolution is used for metrics?                                                 | 389 |
| 6.11 How do I set up Alerting?                                                            | 389 |
| 6.12 How do I use a custom Prometheus configuration file?                                 | 390 |
| 6.13 How to troubleshoot an Update?                                                       | 390 |
| 6.14 What are my login credentials when I try to connect to a Prometheus Exporter?        | 390 |
| 6.15 How to provision PMM Server with non-default admin password?                         | 391 |
| 7. Release Notes                                                                          | 392 |
| 7.1 Release Notes                                                                         | 392 |
| 7.2 Percona Monitoring and Management 2.26.0                                              | 394 |
| 7.3 Percona Monitoring and Management 2.25.0                                              | 396 |
| 7.4 Percona Monitoring and Management 2.24.0                                              | 399 |
| 7.5 Percona Monitoring and Management 2.23.0                                              | 401 |
| 7.6 Percona Monitoring and Management 2.22.0                                              | 403 |
| 7.7 Percona Monitoring and Management 2.21.0                                              | 405 |
| 7.8 Percona Monitoring and Management 2.20.0                                              | 407 |
| 7.9 Percona Monitoring and Management 2.19.0                                              | 409 |
| 7.10 Percona Monitoring and Management 2.18.0                                             | 411 |
| 7.11 Percona Monitoring and Management 2.17.0                                             | 413 |
| 7.12 Percona Monitoring and Management 2.16.0                                             | 415 |
| 7.13 Percona Monitoring and Management 2.15.1                                             | 418 |
| 7.14 Percona Monitoring and Management 2.15.0                                             | 419 |
| 7.15 Percona Monitoring and Management 2.14.0                                             | 423 |
| 7.16 Percona Monitoring and Management 2.13.0                                             | 425 |
| 7.17 Percona Monitoring and Management 2.12.0                                             | 427 |
| 7.18 Percona Monitoring and Management 2.11.1                                             | 429 |
| 7.19 Percona Monitoring and Management 2.11.0                                             | 430 |
| 7.20 Percona Monitoring and Management 2.10.1                                             | 431 |
| 7.21 Percona Monitoring and Management 2.10.0                                             | 432 |

|                                              |     |
|----------------------------------------------|-----|
| 7.22 Percona Monitoring and Management 2.9.1 | 434 |
| 7.23 Percona Monitoring and Management 2.9.0 | 435 |
| 7.24 Percona Monitoring and Management 2.8.0 | 436 |
| 7.25 Percona Monitoring and Management 2.7.0 | 437 |
| 7.26 Percona Monitoring and Management 2.6.1 | 438 |
| 7.27 Percona Monitoring and Management 2.6.0 | 439 |
| 7.28 Percona Monitoring and Management 2.5.0 | 441 |
| 7.29 Percona Monitoring and Management 2.4.0 | 442 |
| 7.30 Percona Monitoring and Management 2.3.0 | 444 |
| 7.31 Percona Monitoring and Management 2.2.2 | 445 |
| 7.32 Percona Monitoring and Management 2.2.1 | 446 |
| 7.33 Percona Monitoring and Management 2.2.0 | 448 |
| 7.34 Percona Monitoring and Management 2.1.0 | 450 |
| 7.35 Percona Monitoring and Management 2.0.1 | 452 |
| 7.36 Percona Monitoring and Management 2.0.0 | 453 |

# 1. Welcome

**Percona Monitoring and Management** (PMM) is a free, open-source monitoring tool for MySQL, PostgreSQL, MongoDB, and ProxySQL, and the servers they run on.

PMM helps you improve the performance of your databases, simplify their management, and strengthen their security. It's efficient, quick to [set up](#), and easy to use.

- PMM **collects** thousands of out-of-the-box performance **metrics** from databases and their hosts.
- The PMM **web UI visualizes data** in **dashboards**.
- Additional features include checking databases for **security threats**.

This is the documentation for the latest release, **PMM 2.26.0** ([Release Notes](#)).

Here's how the home page looks on our [free, live demo system](#).

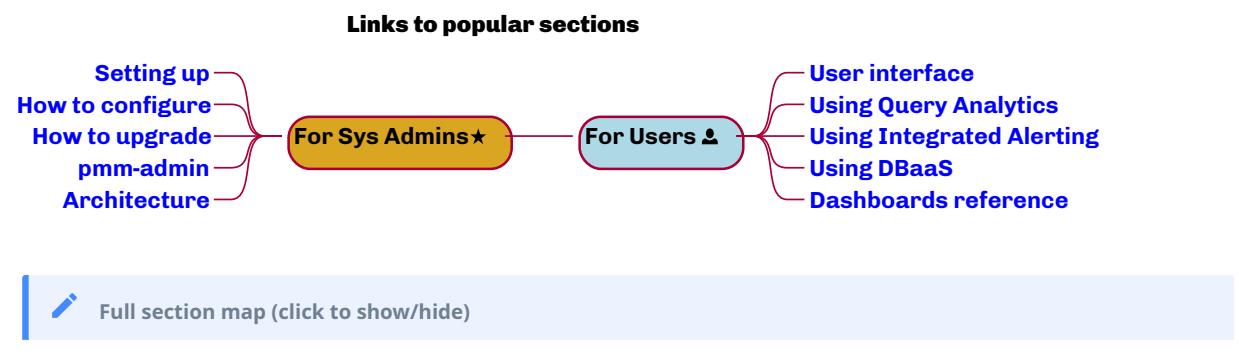
PMM runs in the cloud, on-prem, or across hybrid platforms. It's supported by our [legendary expertise](#) in open source databases, and by a vibrant developer and user [community](#).

A minimal PMM set-up comprises one [server](#) and a [client agent](#) on every system you want to monitor.

## 1.1 Start here

- An [easy install](#) script, which you download, make executable and run. The script installs Docker and runs PMM Server as a container.
- The [Quickstart install guide](#) shows how to run PMM Server as a Docker container, and how to install PMM Client on Ubuntu or Red Hat Linux hosts.
- [Setting Up](#) explains in detail how to set up PMM Server, clients, and how to add services.

## 1.2 Read more



Last update: 2022-01-18

## 2. Setting up

### 2.1 Setting up

There are three stages to installing and setting up PMM.

#### Summary

1. Set up a PMM Server.
2. Set up PMM Client(s).
3. Add services.

#### 2.1.1 Set up PMM Server

Install and run at least one PMM Server.

Choose from:

| Use               | Benefits                                        | Drawbacks                                                                         |
|-------------------|-------------------------------------------------|-----------------------------------------------------------------------------------|
| Docker            | 1. Quick.<br>2. Simple.                         | 1. Docker installation required.<br>2. Additional network configuration required. |
| Virtual appliance | 1. Easily import into Hypervisor of your choice | 1. More system resources compared to Docker footprint.                            |
| Amazon AWS        | 1. Wizard-driven install.                       | 1. Non-free solution (infrastructure costs).                                      |

#### 2.1.2 Set up PMM Client

Install and run PMM Client on every node where there is a service you want to monitor.

The choices:

- With Docker;
- Natively, installed from:
  - Linux package (installed with `apt`, `apt-get`, `dnf`, `yum`);
  - Binary package (a downloaded `.tar.gz` file).



#### Binary is only way to install PMM client without root permissions

#### 2.1.3 Add services

On each PMM Client, you configure then add to PMM Server's inventory the node or service you want to monitor.

How you do this depends on the type of service. You can monitor:

- MySQL (and variants: Percona Server for MySQL, Percona XtraDB Cluster, MariaDB);
  - MongoDB;
  - PostgreSQL;
  - ProxySQL;
  - Amazon RDS;
  - Microsoft Azure;
  - Google Cloud Platform (MySQL and PostgreSQL);
  - Linux;
  - External services;
  - HAProxy;
  - Remote instances.
- 

Last update: 2021-08-03

## 2.2 Server

### 2.2.1 Set up PMM Server

1. Check system requirements.

#### Disk

Approximately 1 GB of storage per monitored database node with data retention set to one week. By default, [retention](#) is 30 days.



#### Tip

[Disable table statistics](#) to decrease the VictoriaMetrics database size.

#### Memory

A minimum of 2 GB per monitored database node. The increase in memory usage is not proportional to the number of nodes. For example, data from 20 nodes should be easily handled with 16 GB.

#### Architecture

Your CPU must support the [SSE4.2](#) instruction set, a requirement of ClickHouse, a third-party column-oriented database used by Query Analytics. If your CPU is lacking this instruction set you won't be able to use Query Analytics.

2. Configure your [network](#).

3. Decide how you want to run PMM Server. Choose from:

- [Docker](#);
- [Virtual appliance](#);
- [Amazon AWS](#);
- Use the [easy install](#) script.

---

Last update: 2021-06-29

## 2.2.2 Network

### Ports

This is a list of ports used by the various components of PMM.

For PMM to work correctly, your system's firewall should allow TCP traffic on these ports (UDP is not needed).

Ports to expose:

| PMM component | TCP port | Direction | Description                                                                                                      |
|---------------|----------|-----------|------------------------------------------------------------------------------------------------------------------|
| PMM Server    | 80       | both      | HTTP server, used for gRPC over HTTP and web interface ( <b>insecure</b> , use with caution).                    |
| PMM Server    | 443      | both      | HTTPS server, used for gRPC over HTTPS and web interface (secure, use of SSL certificates is highly encouraged). |

Other ports:

| PMM component          | TCP port      | Direction | Description                                                                            |
|------------------------|---------------|-----------|----------------------------------------------------------------------------------------|
| PMM Server             | 7771          | both      | gRPC, used for communication between <code>pmm-agent</code> , <code>pmm-admin</code> . |
| PMM Server             | 7772          | out       | HTTP1 server, used for older links like <code>logs.zip</code> .                        |
| PMM Server             | 7773          | out       | Debugging.                                                                             |
| <code>pmm-agent</code> | 7777          | out       | Default <code>pmm-agent</code> listen port.                                            |
| <code>vm-agent</code>  | 8428          | both      | VictoriaMetrics port.                                                                  |
| <code>pmm-agent</code> | 42000 - 51999 | in        | Default range for <code>pmm-agent</code> connected agents.                             |

### Important

Depending on your architecture other ports may also need to be exposed. - For `pmm-agent`, the default listen port is 7777. - The default range for agents ports can be changed with the flag `--ports-min` and `--ports-max`, or in the configuration file.

Last update: 2021-07-07

### 2.2.3 Docker

How to run PMM Server with Docker based on our [Docker image](#).

The tags used here are for the current release. Other [tags](#) are available.

#### See also

[Easy-install script](#)

#### Before you start

- Install [Docker](#) 1.12.6 or higher.

#### Run

#### Summary

- Pull the Docker image.
- Copy it to create a persistent data container.
- Run the image.
- Open the PMM UI in a browser.

1. Pull the image.

```
docker pull percona/pmm-server:2
```

2. Create a persistent data container.

```
docker create --volume /srv \
--name pmm-data \
percona/pmm-server:2 /bin/true
```

### Important

PMM Server expects the data volume to be `/srv`. Using any other value will result in **data loss** when upgrading.

To check server and data container mount points:

```
docker inspect pmm-data | grep Destination && \
docker inspect pmm-server | grep Destination
```

3. Run the image.

```
docker run --detach --restart always \
--publish 443:443 \
--volumes-from pmm-data \
--name pmm-server \
percona/pmm-server:2
```

4. Visit `https://localhost:443` to see the PMM user interface in a web browser. (If you are accessing the docker host remotely, replace `localhost` with the IP or server name of the host.)

## Backup

### Summary

- Stop and rename the `pmm-server` container.
- Take a local copy of the `pmm-data` container's `/srv` directory.

### Important

Grafana plugins have been moved to the data volume `/srv` since the 2.23.0 version. So if you are upgrading PMM from any version before 2.23.0 and have installed additional plugins then plugins should be installed again after the upgrade.

To check used grafana plugins:

```
docker exec -it pmm-server ls /var/lib/grafana/plugins
```

1. Stop the container.

```
docker stop pmm-server
```

2. Move the image.

```
docker rename pmm-server pmm-server-backup
```

3. Create a subdirectory (e.g., `pmm-data-backup`) and move to it.

```
mkdir pmm-data-backup && cd pmm-data-backup
```

4. Backup the data.

```
docker cp pmm-data:/srv .
```

## Upgrade

### Summary

- Stop the running container.
- Backup (rename) the container and copy data.
- Pull the latest Docker image.
- Run it.

### Important

Downgrades are not possible. To go back to using a previous version you must have created a backup of it before upgrading.

 Tip

To see what release you are running, use the *PMM Upgrade* panel on the *Home Dashboard*, or run:

```
docker exec -it pmm-server \
curl -ku admin:admin https://localhost/v1/version
```

(If you are accessing the docker host remotely, replace `localhost` with the IP or server name of the host.)

1. Stop the container.

```
docker stop pmm-server
```

2. Perform a [backup](#).

3. Pull the latest image.

```
docker pull percona/pmm-server:2
```

4. Run it.

```
docker run \
--detach \
--restart always \
--publish 443:443 \
--volumes-from pmm-data \
--name pmm-server \
percona/pmm-server:2
```

5. Perform a [restore](#).

## Restore

 Summary

- Stop and remove the container.
- Restore (rename) the backup container.
- Restore saved data to the data container.
- Restore permissions to the data.

**⚠️ Important**

You must have a [backup](#) to restore from.

1. Stop the container.

```
docker stop pmm-server
```

2. Remove it.

```
docker rm pmm-server
```

3. Revert to the saved image.

```
docker rename pmm-server-backup pmm-server
```

4. Change directory to the backup directory (e.g. `pmm-data-backup`).

5. Remove Victoria Metrics data folder.

```
docker run --rm --volumes-from pmm-data -it percona/pmm-server:2 rm -r /srv/victoriametrics/data
```

6. Copy the data.

```
docker cp srv pmm-data:/
```

7. Restore permissions.

```
docker run --rm --volumes-from pmm-data -it percona/pmm-server:2 chown -R root:root /srv && \
docker run --rm --volumes-from pmm-data -it percona/pmm-server:2 chown -R pmm:pmm /srv/
alertmanager && \
docker run --rm --volumes-from pmm-data -it percona/pmm-server:2 chown -R root:pmm /srv/
clickhouse && \
docker run --rm --volumes-from pmm-data -it percona/pmm-server:2 chown -R grafana:grafana / \
srv/grafana && \
docker run --rm --volumes-from pmm-data -it percona/pmm-server:2 chown -R pmm:pmm /srv/logs \
&& \
docker run --rm --volumes-from pmm-data -it percona/pmm-server:2 chown -R postgres:postgres / \
srv/postgres && \
docker run --rm --volumes-from pmm-data -it percona/pmm-server:2 chown -R pmm:pmm /srv/
prometheus && \
docker run --rm --volumes-from pmm-data -it percona/pmm-server:2 chown -R pmm:pmm /srv/
victoriametrics && \
docker run --rm --volumes-from pmm-data -it percona/pmm-server:2 chown -R postgres:postgres / \
srv/logs/postgresql.log
```

8. Start the image.

```
docker start pmm-server
```

## Remove

### Summary

- Stop the container.
- Remove (delete) both the server and data containers.
- Remove (delete) both images.

### Caution

These steps delete the PMM Server Docker image and any accumulated PMM metrics data.

1. Stop pmm-server container.

```
docker stop pmm-server
```

2. Remove containers.

```
docker rm pmm-server pmm-data
```

3. Remove the image.

```
docker rmi $(docker images | grep "percona/pmm-server" | awk {'print $3'})
```

## Environment variables

Use the following Docker container environment variables (with `-e var=value`) to set PMM Server parameters.

| Variable                              | Description                                                                                      |
|---------------------------------------|--------------------------------------------------------------------------------------------------|
| <code>DISABLE_UPDATES</code>          | Disables a periodic check for new PMM versions as well as ability to apply upgrades using the UI |
| <code>DISABLE_TELEMETRY</code>        | Disable built-in telemetry and disable STT if telemetry is disabled.                             |
| <code>METRICS_RESOLUTION</code>       | High metrics resolution in seconds.                                                              |
| <code>METRICS_RESOLUTION_HR</code>    | High metrics resolution (same as above).                                                         |
| <code>METRICS_RESOLUTION_MR</code>    | Medium metrics resolution in seconds.                                                            |
| <code>METRICS_RESOLUTION_LR</code>    | Low metrics resolution in seconds.                                                               |
| <code>DATA_RETENTION</code>           | How many days to keep time-series data in ClickHouse.                                            |
| <code>ENABLE_VM_CACHE</code>          | Enable cache in VM.                                                                              |
| <code>ENABLE_ALERTING</code>          | Enable integrated alerting.                                                                      |
| <code>ENABLE_AZUREDISCOVER</code>     | Enable support for discovery of Azure databases.                                                 |
| <code>ENABLE_BACKUP_MANAGEMENT</code> | Enable integrated backup tools.                                                                  |
| <code>PERCONA_TEST_SAAS_HOST</code>   | SaaS server hostname.                                                                            |
| <code>PERCONA_TEST_DBaaS</code>       | Enable testing DBaaS features. (Will be deprecated in future versions.)                          |
| <code>ENABLE_DBaaS</code>             | Enable DBaaS features.                                                                           |
| <code>PMM_DEBUG</code>                | Enables a more verbose log level.                                                                |
| <code>PMM_TRACE</code>                | Enables a more verbose log level including trace-back information.                               |

## IGNORED VARIABLES

These variables will be ignored by `pmm-managed` when starting the server. If any other variable is found, it will be considered invalid and the server won't start.

| Variable                                                     | Description                                             |
|--------------------------------------------------------------|---------------------------------------------------------|
| <code>_, HOME, HOSTNAME, LANG, PATH, PWD, SHLVL, TERM</code> | Default environment variables.                          |
| <code>GF_*</code>                                            | Grafana's environment variables.                        |
| <code>SUPERVISOR_</code>                                     | <code>supervisord</code> environment variables.         |
| <code>PERCONA_TEST_</code>                                   | Unknown variable but won't prevent the server starting. |
| <code>PERCONA_TEST_DBaaS</code>                              | Deprecated. Use <code>ENABLE_DBaaS</code> .             |

## Tips

- To Disable the Home Dashboard *PMM Upgrade* panel you can either add `-e DISABLE_UPDATES=true` to the `docker run` command (for the life of the container) or navigate to *PMM* → *PMM Settings* → *Advanced Settings* and disable "Check for Updates" (can be turned back on by any admin in the UI).
- Eliminate browser certificate warnings by configuring a [trusted certificate](#).

- You can optionally enable an (insecure) HTTP connection by adding `--publish 80:80` to the `docker run` command. However, running PMM insecure is not recommended. You should also note that PMM Client *requires* TLS to communicate with the server, only working on a secure port.

#### ISOLATED HOSTS

If the host where you will run PMM Server has no internet connection, you can download the Docker image on a separate (internet-connected) host and securely copy it.

1. On an internet-connected host, download the Docker image and its checksum file.

```
wget https://downloads.percona.com/downloads/pmm2/2.26.0/docker/pmm-server-2.26.0.docker
wget https://downloads.percona.com/downloads/pmm2/2.26.0/docker/pmm-server-2.26.0.sha256sum
```

2. Copy both files to where you will run PMM Server.
3. Open a terminal on the PMM Server host.
4. (Optional) Check the Docker image file integrity.

```
shasum -ca 256 pmm-server-2.26.0.sha256sum
```

5. Load the image.

```
docker load -i pmm-server-2.26.0.docker
```

6. Create the `pmm-data` persistent data container.

```
docker create --volume /srv \
--name pmm-data percona/pmm-server:2.26.0 /bin/true
```

7. Run the container.

```
docker run \
--detach \
--restart always \
--publish 443:443 \
--volumes-from pmm-data \
--name pmm-server \
percona/pmm-server:2.26.0
```

---

Last update: 2022-02-04

## 2.2.4 Virtual Appliance

How to run PMM Server as a virtual machine.

### Summary

- Download and verify the [latest OVF file](#).
- Import it.
- Reconfigure network.
- Start the VM and get IP.
- Log into PMM UI.
- (Optional) Change VM root password.
- (Optional) Set up SSH.
- (Optional) Set up static IP.

Most steps can be done with either a user interface or on the command line, but some steps can only be done in one or the other. Sections are labeled **UI** for user interface or **CLI** for command line instructions.

### Terminology

- *Host* is the desktop or server machine running the hypervisor.
- *Hypervisor* is software (e.g. [VirtualBox](#), [VMware](#)) that runs the guest OS as a virtual machine.
- *Guest* is the CentOS virtual machine that runs PMM Server.

### OVA file details

| Item          | Value                                                                                                             |
|---------------|-------------------------------------------------------------------------------------------------------------------|
| Download page | <a href="https://www.percona.com/downloads/pmm2/2.26.0/ova">https://www.percona.com/downloads/pmm2/2.26.0/ova</a> |
| File name     | pmm-server-2.26.0.ova                                                                                             |
| VM name       | PMM2-Server-2022-02-08-N ( N =build number)                                                                       |

### VM specifications

| Component      | Value                   |
|----------------|-------------------------|
| OS             | CentOS 7.9 (64-bit)     |
| CPU            | 1                       |
| Base memory    | 4096 MB                 |
| Disk           | LVM, 2 physical volumes |
| Disk 1 ( sda ) | VMDK (SCSI, 40 GB)      |
| Disk 2 ( sdb ) | VMDK (SCSI, 400 GB)     |

## Users

| Default Username | Default password |
|------------------|------------------|
| root             | percona          |
| admin            | admin            |

## Download

### UI

1. Open a web browser.
2. [Visit the PMM Server download page.](#)
3. Choose a *Version* or use the default (the latest).
4. Click the link for `pmm-server-2.26.0.ova` to download it. Note where your browser saves it.
5. Right click the link for `pmm-server-2.26.0.sha256sum` and save it in the same place as the `.ova` file.
6. (Optional) [Verify](#).

### CLI

Download the latest PMM Server OVA and checksum files.

```
wget https://www.percona.com/downloads/pmm2/2.26.0/ova/pmm-server-2.26.0.ova
wget https://www.percona.com/downloads/pmm2/2.26.0/ova/pmm-server-2.26.0.sha256sum
```

## Verify

### CLI

Verify the checksum of the downloaded `.ova` file.

```
shasum -ca 256 pmm-server-2.26.0.sha256sum
```

## VMware

### IMPORT

#### UI

1. Select *File* → *Import*.
2. Click *Choose file...*
3. Navigate to the downloaded `.ova` file and select it.
4. Click *Open*.
5. Click *Continue*.
6. In the *Save as* dialog:
  - a. (Optional) Change the directory or file name.
  - b. Click *Save*.
7. Choose one of:
  - (Optional) Click *Finish*. This starts the virtual machine.
  - (Recommended) Click *Customize Settings*. This opens the VM's settings page without starting the machine.

**CLI**

1. Install `ovftool`. (You need to register.)
2. Import and convert the OVA file. (`ovftool` can't change CPU or memory settings during import but it can set the default interface.)

Choose one of:

- Download and import the OVA file.

```
ovftool --name="PMM Server" --net:NAT=Wi-Fi \
https://www.percona.com/downloads/pmm2/2.26.0/ova/pmm-server-2.26.0.ova \
pmm-server-2.26.0.vmx
```

- Import an already-downloaded OVA file.

```
ovftool --name="PMM Server" --net:NAT=WiFi \
pmm-server-2.26.0.ova \
pmm-server.vmx
```

**RECONFIGURE INTERFACE**

When using the command line, the interface is remapped during import.

**UI**

1. If started, shut down the virtual machine.
2. In the VMware main window, select the imported virtual machine.
3. Click *Virtual Machine* → *Settings*....
4. Click *Network Adapter*.
5. In the *Bridged Networking* section, select *Autodetect*.
6. Close the settings window.

**START GUEST AND GET IP ADDRESS****UI**

1. In the VMware main window, select the imported virtual machine.
2. Click the play button or select *Virtual Machine* → *Start Up*.
3. When the instance has booted, note the IP address in the guest console.

**CLI/UI**

1. Start the virtual machine in GUI mode. (There's no way to redirect a VMware VM's console to the host.)

```
vmlrun -gu root -gp percona start \
pmm-server.vmx gui
```

2. When the instance has booted, note the IP address in the guest console.
3. (Optional) Stop and restart the instance in headless mode.

```
vmlrun stop pmm-server.vmx
vmlrun -gu root -gp percona start \
pmm-server.vmx nogui
```

**VirtualBox****IMPORT****UI**

1. Select *File* → *Import appliance*....
2. In the *File* field, type the path to the downloaded `.ova` file, or click the folder icon to navigate and open it.
3. Click *Continue*.
4. On the *Appliance settings* page, review the settings and click *Import*.
5. Click *Start*.
6. When the guest has booted, note the IP address in the guest console.

**CLI**

1. Open a terminal and change directory to where the downloaded `.ova` file is.
2. (Optional) Do a 'dry run' import to see what values will be used.

```
VBoxManage import pmm-server-2.26.0.ova --dry-run
```

3. Import the image. Choose one of:

- With the default settings.

```
VBoxManage import pmm-server-2.26.0.ova
```

- With custom settings (in this example, Name: "PMM Server", CPUs: 2, RAM: 8192 MB).

```
VBoxManage import --vsys 0 --vmname "PMM Server" \
--cpus 2 --memory 8192 pmm-server-2.26.0.ova
```

**INTERFACE****UI**

1. Click *Settings*.
2. Click *Network*.
3. In the *Adapter 1* field, click *Attached to* and change to *Bridged Adapter*.
4. In the *Name* field, select your host's active network interface (e.g. `en0: Wi-Fi (Wireless)`).
5. Click *OK*.

**CLI**

1. Show the list of available bridge interfaces.

```
VBoxManage list bridgedifs
```

2. Find the name of the active interface you want to bridge to (one with *Status: Up* and a valid IP address). Example:  
`en0: Wi-Fi (Wireless)`
3. Bridge the virtual machine's first interface (`nic1`) to the host's `en0` ethernet adapter.

```
VBoxManage modifyvm 'PMM Server' \
--nic1 bridged --bridgeadapter1 'en0: Wi-Fi (Wireless)'
```

4. Redirect the console output into a host file.

```
VBoxManage modifyvm 'PMM Server' \
--uart1 0x3F8 4 --uartmodel file /tmp/pmm-server-console.log
```

#### GET IP

##### UI

1. Select the *PMM Server* virtual machine in the list.
2. Click *Start*.
3. When the guest has booted, note the IP address in the guest console.

##### CLI

1. Start the guest.

```
VBoxManage startvm --type headless 'PMM Server'
```

2. (Optional) Watch the log file.

```
tail -f /tmp/pmm-server-console.log
```

3. Wait for one minute for the server to boot up.

4. Choose one of:

- Read the IP address from the tailed log file.
- Extract the IP address from the log file.

```
grep -e "^IP:" /tmp/pmm-server-console.log | cut -f2 -d' '
```

5. (Optional) Stop the guest:

```
VBoxManage controlvm "PMM Server" poweroff
```

#### Log into user interface

##### UI

1. Open a web browser and visit the guest IP address.
2. The PMM **Login** screen appears.
3. Enter the default username and password in the relevant fields and click *Log in*.
  - username: **admin**
  - password: **admin**
4. (Recommended) Follow the prompts to change the default password.
5. The PMM Home Dashboard appears.

#### (Optional) Change root password

##### UI

1. Start the virtual machine in GUI mode.

2. Log in with the default superuser credentials:
  - Username: `root`
  - Password: `percona`
3. Follow the prompts to change the password.

#### (Optional) Set up SSH

UI/CLI

1. Create a key pair for the `admin` user.

```
ssh-keygen -f admin
```

2. Log into the PMM user interface.
3. Select *PMM* → *PMM Settings* → *SSH Key*.
4. Copy and paste the contents of the `admin.pub` file into the *SSH Key* field.
5. Click *Apply SSH Key*. (This copies the public key to `/home/admin/.ssh/authorized_keys` in the guest).
6. Log in via SSH (`N.N.N.N` is the guest IP address).

```
ssh -i admin admin@N.N.N.N
```

#### (Optional) Set up static IP

When the guest OS starts, it will get an IP address from the hypervisor's DHCP server. This IP can change each time the guest OS is restarted. Setting a static IP for the guest OS avoids having to check the IP address whenever the guest is restarted.

CLI

1. Start the virtual machine in non-headless (GUI) mode.
2. Log in as `root`.
3. Edit `/etc/sysconfig/network-scripts/ifcfg-eth0`
4. Change the value of `BOOTPROTO`:

```
BOOTPROTO=none
```

5. Add these values:

```
IPADDR=192.168.1.123 # replace with the desired static IP address
NETMASK=255.255.255.0 # replace with the netmask for your IP address
GATEWAY=192.168.1.1 # replace with the network gateway for your IP address
PEERDNS=no
DNS1=192.168.1.53 # replace with your DNS server IP
```

6. Restart the interface.

```
ifdown eth0 && ifup eth0
```

7. Check the IP.

```
ip addr show eth0
```

8. Preserve the network configuration across reboots.

```
echo "network: {config: disabled}" > /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg
```

## Remove

### UI

1. Stop the virtual machine: select *Close* → *Power Off*.
  2. Remove the virtual machine: select *Remove* → *Delete all files*.
- 

Last update: 2021-12-29

## 2.2.5 AWS Marketplace

You can run an instance of PMM Server hosted at AWS Marketplace.

Assuming that you have an AWS (Amazon Web Services) account, locate *Percona Monitoring and Management Server* in [AWS Marketplace](#) or use [this link](#).

The screenshot shows the AWS Marketplace listing for the Percona Monitoring and Management Server. At the top, there's a yellow 'Continue to Subscribe' button. Below it is a 'Save to List' button. To the right, a box displays the 'Typical Total Price' as '\$0.10/hr'. Below this, a note states: 'Total pricing per instance for services hosted on m4.large in US East (N. Virginia). [View Details](#)'.

Selecting a region and instance type in the *Pricing Information* section will give you an estimate of the costs involved. This is only an indication of costs. You will choose regions and instance types in later steps.

Percona Monitoring and Management Server is provided at no cost, but you may need to pay for infrastructure costs.

Disk space consumed by PMM Server depends on the number of hosts being monitored. Although each environment will be unique, you can consider the data consumption figures for the [PMM Demo](#) web site which consumes approximately 230 MB per host per day, or approximately 6.9 GB per host at the default 30 day retention period.

For more information, see our blog post [How much disk space should I allocate for Percona Monitoring and Management?](#).

1. Click *Continue to Subscribe*.
2. **Subscribe to this software:** Check the terms and conditions and click *Continue to Configuration*.
3. **Configure this software:**
  - a. Select a value for *Software Version*. (The latest is 2.26.0.)
  - b. Select a region. (You can change this in the next step.)
  - c. Click *Continue to Launch*.
4. **Launch this software:**
  - a. **Choose Action:** Select a launch option. *Launch from Website* is a quick way to make your instance ready. For more control, choose *Launch through EC2*.
  - b. **EC2 Instance Type:** Select an instance type.
  - c. **VPC Settings:** Choose or create a VPC (virtual private cloud).
  - d. **Subnet Settings:** Choose or create a subnet.
  - e. **Security Group Settings:** Choose a security group or click \*Create New Based On Seller Settings
  - f. **Key Pair Settings:** Choose or create a key pair.
  - g. Click *Launch*.

### Limiting Access to the instance: security group and a key pair

In the *Security Group* section, which acts like a firewall, you may use the preselected option `Create new based on seller settings` to create a security group with recommended settings. In the *Key Pair* select an already set up EC2 key pair to limit access to your instance.

## ▼ Key Pair



To ensure that no other person has access to your software, the software installs on an EC2 instance with an EC2 key pair that you created. Choose an existing EC2 key pair in the list.

pmm\_user

### ⚠ Important

The security group should allow communication via the the following ports: 22, 80, and 443. PMM should also be able to access port 3306 on the RDS that uses the instance.

## ▼ Security Group

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. Learn more about [Security Groups](#).

You can create a new security group based on seller-recommended settings or choose one of your existing groups.

Create new based on seller settings



A new security group will be generated by AWS Marketplace. It is based on recommended settings for Percona Monitoring and Management Server provided by Percona.

| Connection Method | Protocol | Port Range | Source (IP or Group) |
|-------------------|----------|------------|----------------------|
| SSH               | tcp      | 22 - 22    | Anywhere ▾ 0.0.0.0/0 |
| HTTP              | tcp      | 80 - 80    | Anywhere ▾ 0.0.0.0/0 |
| HTTPS             | tcp      | 443 - 443  | Anywhere ▾ 0.0.0.0/0 |

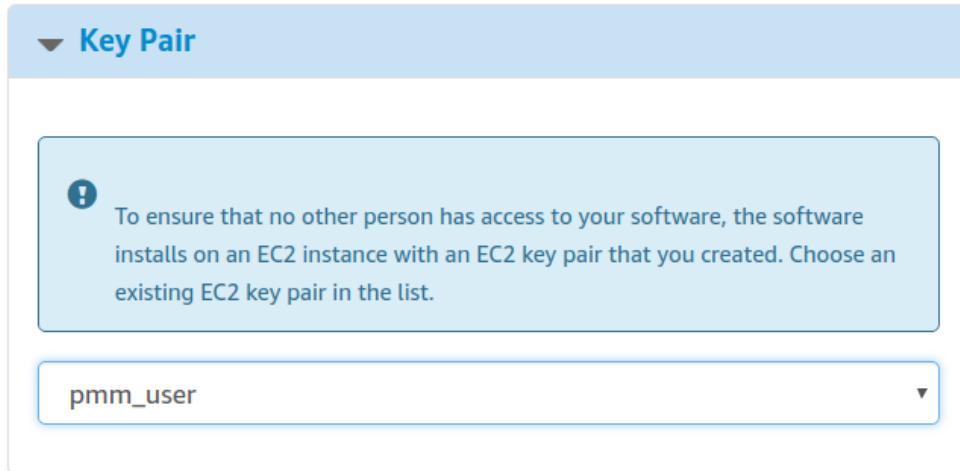


Rules with source of 0.0.0.0/0 allows all IP addresses to access your instance. We recommend limiting access to only known IP addresses.

## Applying settings

Scroll up to the top of the page to view your settings. Then, click the *Launch with 1 click* button to continue and adjust your settings in the EC2 console.

Your instance settings are summarized in a special area. Click the Launch with 1 click button to continue.



The *Launch with 1 click* button may alternatively be titled as *Accept Software Terms & Launch with 1-Click*.

## Adjusting instance settings in the EC2 Console

Your clicking the *Launch with 1 click* button, deploys your instance. To continue setting up your instance, run the EC2 console. It is available as a link at the top of the page that opens after you click the *Launch with 1 click* button.

Your instance appears in the EC2 console in a table that lists all instances available to you. When a new instance is only created, it has no name. Make sure that you give it a name to distinguish from other instances managed via the EC2 console.

|  | <a href="#">Launch Instance</a>                                                  | <a href="#">Connect</a>     | <a href="#">Actions ▾</a>     |
|--|----------------------------------------------------------------------------------|-----------------------------|-------------------------------|
|  | <input type="text"/> <a href="#">search : pmm-doc</a> <a href="#">Add filter</a> |                             |                               |
|  | <a href="#">Name</a>                                                             | <a href="#">Instance ID</a> | <a href="#">Instance Type</a> |
|  | pmm-doc                                                                          | i-0d818c28d870b8cdc         | m4.large                      |

## Running the instance

After you add your new instance it will take some time to initialize it. When the AWS console reports that the instance is now in a running state, you may continue with configuration of PMM Server.

When started the next time after rebooting, your instance may acquire another IP address. You may choose to set up an elastic IP to avoid this problem.

With your instance selected, open its IP address in a web browser. The IP address appears in the *IPv4 Public IP* column or as value of the *Public IP* field at the top of the *Properties* panel.

Instance: i-07fe5e8eb17aca0cf (pmm-doc) Public IP: 34.229.131.138

| Description       |                          | Status Checks | Monitoring | Tags | Usage Instructions |
|-------------------|--------------------------|---------------|------------|------|--------------------|
| Instance ID       | i-07fe5e8eb17aca0cf      |               |            |      |                    |
| Instance state    | running                  |               |            |      |                    |
| Instance type     | m4.large                 |               |            |      |                    |
| Elastic IPs       |                          |               |            |      |                    |
| Availability zone | us-east-1f               |               |            |      |                    |
| Security groups   | pmm · view inbound rules |               |            |      |                    |

Public DNS (IPv4) -  
IPv4 Public IP 34.229.131.138  
IPv6 IPs -  
Private DNS ip-172-30-6-213.ec2.internal  
Private IPs 172.30.6.213  
Secondary private IPs -

To run the instance, copy and paste its public IP address to the location bar of your browser. In the *Percona Monitoring and Management* welcome page that opens, enter the instance ID.

## Instance ID verification

Please provide the Amazon Instance ID (AMI ID) from the AWS Console.  
It uses the format of i-abc123def

Instance ID  Submit

[Where should I get my instance ID?](#)

You can copy the instance ID from the *Properties* panel of your instance, select the *Description* tab back in the EC2 console. Click the *Copy* button next to the *Instance ID* field. This button appears as soon as you hover the cursor of your mouse over the ID.

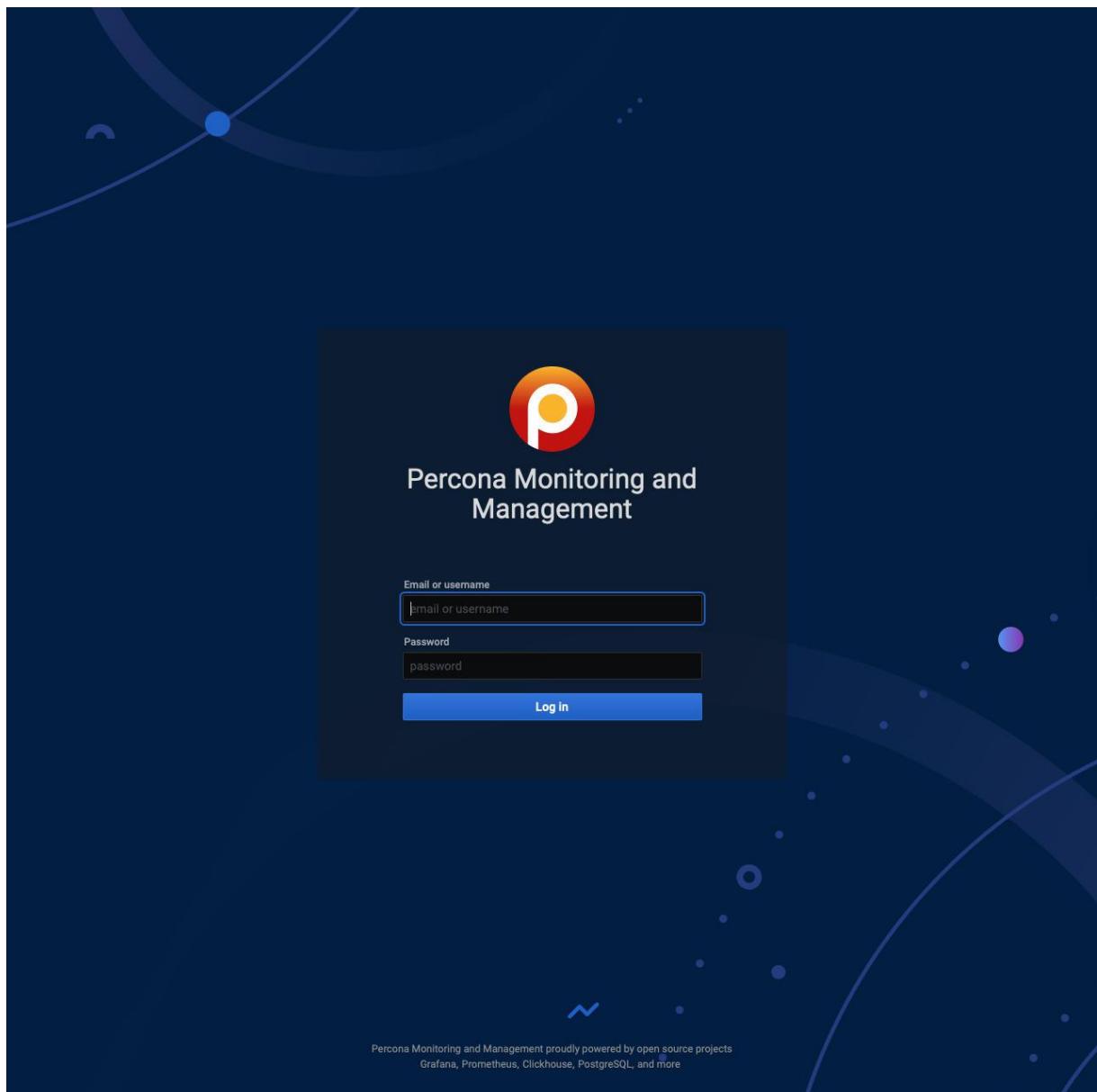
Hover the cursor over the instance ID for the Copy button to appear.

Instance: i-07fe5e8eb17aca0cf (pmm-doc) Public IP: 34.229.131.138

| Description |                     | Status Checks | Monitoring | Tags | Usage Instructions |
|-------------|---------------------|---------------|------------|------|--------------------|
| Instance ID | i-07fe5e8eb17aca0cf |               |            |      |                    |

Paste the instance in the *Instance ID* field of the *Percona Monitoring and Management* welcome page and click *Submit*.

PMM Server provides user access control, and therefore you will need user credentials to access it:



- Default user name: `admin`
- Default password: `admin`

You will be prompted to change the default password every time you log in.

The PMM Server is now ready and the home page opens.

**General information**

**Overview**  
Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB and PostgreSQL performance, and provides time-based analysis to ensure that your data works as efficiently as possible.

**Documentation**  
Please consult the official PMM documentation to learn more about PMM. Also of interest are the [Release notes](#) and [FAQ](#) for common questions about PMM.

**Community and Blogs**  
On the [PMM Community Forums](#) you will find help from Perconians and the Community at large. Further, we publish PMM announcements and use cases regularly on the [Percona Database](#).

**Percona News**

- MongoDB – Converting Replica Set to Standalone Jan 03  
One of the reasons for using MongoDB is its simplicity in scaling its structure, either as Replicaset(scale-up) or as Sharded Cluster(scale-out). Although a bit tricky in Sharded environments, the reverse process is also feasible. From the possibilities, you can: Reduce the number of shards from your Sharded Cluster as documented here. You can also reduce the [...]
- Updated Percona Distribution Jan 03 for MongoDB, Bug Fixes in Percona Server for MySQL: Release Roundup January 3

**Starred & Recently used**

- Home Dashboard Insight
- Nodes Overview OS
- MySQL Instances Overview MySQL

**Monitored nodes**  
**30**  
Failed security checks  
Insufficient access permissions.

**Monitored DB ...**  
**28**  
Insufficient access permissions.

**PMM Upgrade**  
Current version: 2.25.0 (December 13, 2021)  
Insufficient access permissions

Last check: January 03, 16:38

**Environment Overview**

| Host              | CPU B... | Mem ... | Disk R...  | Disk ...  | Netw...   | DB Co... | DB QPS  | Virtua... | RAM        | Host ...   | DB up...  |
|-------------------|----------|---------|------------|-----------|-----------|----------|---------|-----------|------------|------------|-----------|
| All               | 22.9%    | 49.2%   | 836.4 kB/s | 37.0 MB/s | 63.3 MB/s | 431      | 16.50 K | 60        | 113.41 GiB | 14.9 weeks | 1.6 days  |
| AzureDB-mysql     | N/A      | N/A     | N/A        | N/A       | N/A       | 16       | 132.45  | N/A       | N/A        | N/A        | 3.8 weeks |
| load-gen-sysbench | 64.5%    | 44.5%   | 0 B/s      | 1.8 kB/s  | 26.0 MB/s | N/A      | N/A     | 1         | 1.77 GiB   | 30.5 weeks | N/A       |

You are creating a username and password that will be used for two purposes:

1. authentication as a user to PMM - the credentials to log in to PMM.
2. authentication between PMM Server and PMM Clients - you will re-use these credentials on another host when configuring PMM Client for the first time on a server, for example (DO NOT RUN ON THIS PMM SERVER YOU JUST CREATED):

```
pmm-admin config --server-insecure-tls --server-url=https://admin:admin@<IP Address>:443
```

For instructions about how to access your instances by using an SSH client, see [Connecting to Your Linux Instance Using SSH](#)

Make sure to replace the user name `ec2-user` used in this document with `admin`.

### Resizing the EBS Volume

Your AWS instance comes with a predefined size which can become a limitation. To make more disk space available to your instance, you need to increase the size of the EBS volume as needed and then your instance will reconfigure itself to use the new size.

The procedure of resizing EBS volumes is described in the Amazon documentation: [Modifying the Size, IOPS, or Type of an EBS Volume on Linux](#).

After the EBS volume is updated, PMM Server instance will auto-detect changes in approximately 5 minutes or less and will reconfigure itself for the updated conditions.

## Upgrading PMM Server on AWS

### CHANGE PUBLIC IP ADDRESS

To assign a public IP address for an Amazon EC2 instance, follow these steps:

1. Allocate Elastic IP address

The screenshot shows the AWS EC2 console with the path: EC2 > Elastic IP addresses > Allocate Elastic IP address. The main section is titled "Elastic IP address settings". It includes a search bar for "Network Border Group" containing "us-east-1". Below it, under "Public IPv4 address pool", the "Amazon's pool of IPv4 addresses" option is selected. A note states: "Public IPv4 address that you bring to your AWS account (option disabled because no pools found) [Learn more](#)". There is also a note about "Customer owned pool of IPv4 addresses" which is disabled. A "Global static IP addresses" section is present with a note about AWS Global Accelerator. A "Create accelerator" button is at the bottom.

2. Associate Elastic IP address with a Network interface ID of your EC2 instance

If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account.

The screenshot shows the AWS EC2 console with the path: EC2 > Elastic IP addresses > 54.198.227.51 > Associate Elastic IP address. The main section is titled "Associate Elastic IP address". It asks to choose an instance or network interface to associate the IP address. A note says: "If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. [Learn more](#)". The "Resource type" section shows "Network interface" selected. The "Network interface" field contains "eni-03e9098558683d2a0". The "Private IP address" field is empty. The "Reassociation" section has a checkbox "Allow this Elastic IP address to be reassociated" which is unchecked. At the bottom are "Cancel" and "Associate" buttons.

## UPGRADING EC2 INSTANCE CLASS

Upgrading to a larger EC2 instance class is supported by PMM provided you follow the instructions from the [AWS manual](#). The PMM AMI image uses a distinct EBS volume for the PMM data volume which permits independent resizing of the EC2 instance without impacting the EBS volume.

1. Open the Amazon EC2 console.
2. In the navigation pane, choose PMM Server Instances.
3. Select the instance and choose Actions, Instance state, Stop instance.
4. In the Change instance type dialog box, select the instance type that you want.

| Name                                             | Instance ID         | Instance state | Instance type |
|--------------------------------------------------|---------------------|----------------|---------------|
| -                                                | i-0f98abbc8ec526807 | Running        | m4.large      |
| <input checked="" type="checkbox"/> adivinho-pmm | 106e                | Stopped        | m5.large      |
| <input type="checkbox"/> mongo-test-pmm2-HD-7    | a3f5                | Running        | t2.small      |
| <input type="checkbox"/> pxc-proxysql-QA-pmm-HI  | 8978                | Running        | t2.small      |
| <input type="checkbox"/> postgresql-11-pmm-HD-8  | id365               | Running        | t2.small      |

**Instance: i-0d937e08a6681106**

Details   Security   Networking

- Instance summary   Info
- Instance ID: i-0d937e08a6681106e (adivinho-pmm)
- Instance state: Stopped
- Instance type: m5.large

Instance settings ▶
 

- Attach to Auto Scaling Group
- Change instance type**
- Change Nitro Enclaves
- Change termination protection
- Change shutdown behavior
- Change credit specification
- Modify instance placement
- Modify Capacity Reservation settings
- Edit user data
- Manage tags

Networking ▶
 

- Elastic IP addresses
-

5. Choose Apply to accept the new settings and start the stopped instance.

## EXPANDING THE PMM DATA EBS VOLUME

The PMM data volume is mounted as an XFS formatted volume on top of an LVM volume. There are two ways to increase this volume size:

1. Add a new disk via EC2 console or API, and expand the LVM volume to include the new disk volume.
2. Expand existing EBS volume and grow the LVM volume.

**EXPAND EXISTING EBS VOLUME**

To expand the existing EBS volume for increased capacity, follow these steps.

1. Expand the disk from AWS Console/CLI to the desired capacity.
2. Login to the PMM EC2 instance and verify that the disk capacity has increased. For example, if you have expanded disk from 16G to 32G, `dmesg` output should look like below:

```
[ 535.994494] xvdb: detected capacity change from 17179869184 to 34359738368
```

3. You can check information about volume groups and logical volumes with the `vgs` and `lvs` commands:

```
vgs
```

```
VG #PV #LV #SN Attr VSize VFree
DataVG 1 2 0 wz--n- <16.00g 0
```

```
lvs
```

```
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
DataLV DataVG Vwi-aotz-- <12.80g ThinPool 1.74
ThinPool DataVG twi-aotz-- 15.96g 1.39 1.29
```

4. Now we can use the `lsblk` command to see that our disk size has been identified by the kernel correctly, but LVM2 is not yet aware of the new size. We can use `pvresize` to make sure the PV device reflects the new size. Once `pvresize` is executed, we can see that the VG has the new free space available.

```
lsblk | grep xvdb
```

```
xvdb 202:16 0 32G 0 disk
```

```
pvscan
```

```
PV /dev/xvdb VG DataVG lvm2 [<16.00 GiB / 0 free]
Total: 1 [<16.00 GiB] / in use: 1 [<16.00 GiB] / in no VG: 0 [0 ]
```

```
pvresize /dev/xvdb
```

```
Physical volume "/dev/xvdb" changed
1 physical volume(s) resized / 0 physical volume(s) not resized
```

```
pvs
```

```
PV VG Fmt Attr PSize PFree
/dev/xvdb DataVG lvm2 a-- <32.00g 16.00g
```

5. We then extend our logical volume. Since the PMM image uses thin provisioning, we need to extend both the pool and the volume:

```
lvs
```

```
LV      VG      Attr       LSize   Pool      Origin Data%  Meta% Move Log Cpy%Sync Convert
DataLV  DataVG Vwi-aotz-- <12.80g ThinPool          1.77
ThinPool DataVG twi-aotz--  15.96g                  1.42   1.32
```

```
lvextend /dev/mapper/DataVG-ThinPool -l 100%VG
```

Size of logical volume DataVG/ThinPool\_tdata changed from 16.00 GiB (4096 extents) to 31.96 GiB (8183 extents).  
Logical volume DataVG/ThinPool\_tdata successfully resized.

```
lvs
```

```
LV      VG      Attr       LSize   Pool      Origin Data%  Meta% Move Log Cpy%Sync Convert
DataLV  DataVG Vwi-aotz-- <12.80g ThinPool          1.77
ThinPool DataVG twi-aotz--  31.96g                  0.71   1.71
```

6. Once the pool and volumes have been extended, we need to now extend the thin volume to consume the newly available space. In this example we've grown available space to almost 32GB, and already consumed 12GB, so we're extending an additional 19GB:

```
lvs
```

```
LV      VG      Attr       LSize   Pool      Origin Data%  Meta% Move Log Cpy%Sync Convert
DataLV  DataVG Vwi-aotz-- <12.80g ThinPool          1.77
ThinPool DataVG twi-aotz--  31.96g                  0.71   1.71
```

```
lvextend /dev/mapper/DataVG-DataLV -L +19G
```

Size of logical volume DataVG/DataLV changed from <12.80 GiB (3276 extents) to <31.80 GiB (8140 extents).  
Logical volume DataVG/DataLV successfully resized.

```
lvs
```

```
LV      VG      Attr       LSize   Pool      Origin Data%  Meta% Move Log Cpy%Sync Convert
DataLV  DataVG Vwi-aotz-- <31.80g ThinPool          0.71
ThinPool DataVG twi-aotz--  31.96g                  0.71   1.71
```

7. We then expand the XFS file system to reflect the new size using `xfs_growfs`, and confirm the file system is accurate using the `df` command.

```
df -h /srv
```

| Filesystem                | Size | Used | Avail | Use% | Mounted on |
|---------------------------|------|------|-------|------|------------|
| /dev/mapper/DataVG-DataLV | 13G  | 249M | 13G   | 2%   | /srv       |

```
xfs_growfs /srv
```

```
meta-data=/dev/mapper/DataVG-DataLV isize=512    agcount=103, agsize=32752 blks
        =                                     sectsz=512  attr=2, projid32bit=1
```

```

=                               crc=1      finobt=0 spinodes=0
data   =                         bsize=4096  blocks=3354624, imaxpct=25
=                               sunit=16   swidth=16 blks
naming =version 2               bsize=4096  ascii-ci=0 ftype=1
log    =internal                bsize=4096  blocks=768, version=2
=                               sectsz=512  sunit=16 blks, lazy-count=1
realtime =none                 extsz=4096  blocks=0, rtextents=0
data blocks changed from 3354624 to 8335360

```

```
df -h /srv
```

| Filesystem                | Size | Used | Avail | Use% | Mounted on |
|---------------------------|------|------|-------|------|------------|
| /dev/mapper/DataVG-DataLV | 32G  | 254M | 32G   | 1%   | /srv       |

#### EXPAND THE AMAZON EBS ROOT VOLUME

1. Expand the disk from AWS Console/CLI to the desired capacity.
2. Login to the PMM EC2 instance and verify that the disk capacity has increased. For example, if you have expanded disk from 8G to 10G, `dmesg` output should look like below:

```
# dmesg | grep "capacity change"
[63175.044762] nvme0n1: detected capacity change from 8589934592 to 10737418240
```

3. Use the `lsblk` command to see that our disk size has been identified by the kernel correctly, but LVM2 is not yet aware of the new size.

```
# lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1       259:1   0   10G  0 disk
└─nvme0n1p1   259:2   0     8G  0 part /
...
```

4. For volumes that have a partition, such as the root volume shown in the previous step, use the `growpart` command to extend the partition.

```
# growpart /dev/nvme0n1 1
CHANGED: partition=1 start=2048 old: size=16775168 end=16777216 new: size=20969439
end=20971487
```

5. To verify that the partition reflects the increased volume size, use the `lsblk` command again.

```
# lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1       259:1   0   10G  0 disk
└─nvme0n1p1   259:2   0     10G 0 part /
...
```

6. Extend the XFS file system on the root volume by `xfs_growfs` command. I

```
# xfs_growfs -d /
meta-data=/dev/nvme0n1p1      isize=512    agcount=4, agsize=524224 blks
=                           sectsz=512   attr=2, projid32bit=1
=                           crc=1      finobt=0 spinodes=0
data   =                         bsize=4096   blocks=2096896, imaxpct=25
=                           sectsz=512   sunit=0   swidth=0 blks
naming =version 2               bsize=4096   ascii-ci=0 ftype=1
log    =internal                bsize=4096   blocks=2560, version=2
```

```
=
realtime =none
data blocks changed from 2096896 to 2621120
```

## 7. Verify that file system reflects the increased volume size

```
# df -hT /
Filesystem      Type  Size  Used  Avail Use% Mounted on
/dev/nvme0n1p1  xfs   10G   5,6G  4,5G  56% /
```

## Backup PMM Server

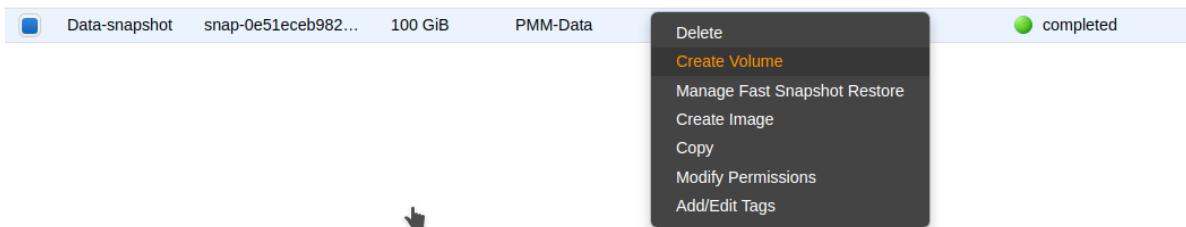
All data are stored in the `/srv` partition, so it's enough to back the PMM data volume. You can create a point-in-time snapshot of the volume and use it for data backup.

The procedure of creating a snapshot is described in the Amazon documentation: [Create Amazon EBS snapshots](#)



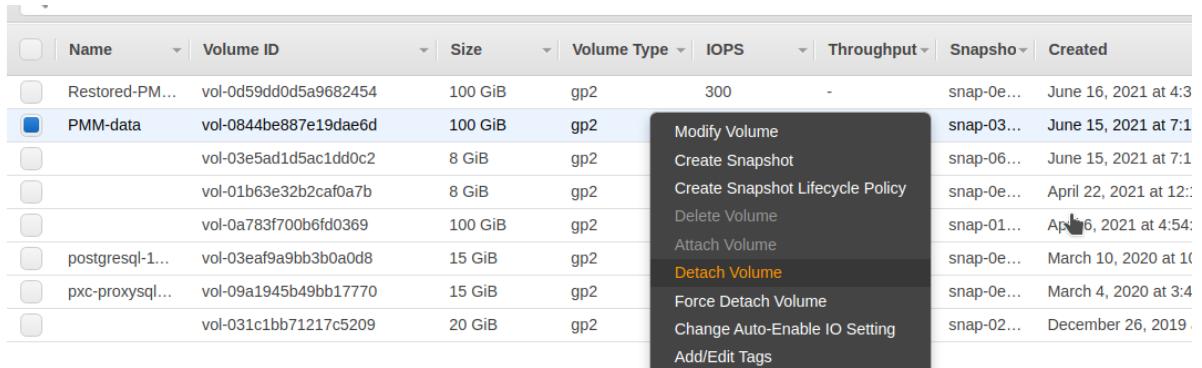
## Restore PMM Server from a backup

### 1. Create a new volume by using the latest snapshot of the PMM data volume.



### 2. Stop the PMM Server instance.

### 3. Detach the current PMM data volume.



4. Attach the new volume.

|                                     | Name                       | Volume ID             | Size    | Volume Type | IOPS | Throughput | Snapshots | Created          |
|-------------------------------------|----------------------------|-----------------------|---------|-------------|------|------------|-----------|------------------|
| <input checked="" type="checkbox"/> | Restored-PMM-Data          | vol-0d59dd0d5a9682454 | 100 GiB | gp2         | 300  |            |           | June 16, 2021 a  |
| <input type="checkbox"/>            | PMM-data                   | vol-0844be887e19dae6d | 100 GiB | gp2         | 300  |            |           | June 15, 2021 a  |
| <input type="checkbox"/>            |                            | vol-03e5ad1d5ac1dd0c2 | 8 GiB   | gp2         | 100  |            |           | June 15, 2021 a  |
| <input type="checkbox"/>            |                            | vol-01b63e32b2caf0a7b | 8 GiB   | gp2         | 100  |            |           | April 22, 2021 a |
| <input type="checkbox"/>            |                            | vol-0a783f700b6fd0369 | 100 GiB | gp2         | 300  |            |           | April 6, 2021 at |
| <input type="checkbox"/>            | postgresql-11-pmm-HD-8363  | vol-03eaf9a9bb3b0a0d8 | 15 GiB  | gp2         | 100  |            |           | March 10, 2020   |
| <input type="checkbox"/>            | pxc-proxysql-QA-pmm-HD-... | vol-09a1945b49bb17770 | 15 GiB  | gp2         | 100  |            |           | March 4, 2020 a  |
| <input type="checkbox"/>            |                            | vol-031c1bb71217c5209 | 20 GiB  | gp2         | 100  |            |           | December 26, 2   |

5. Start the PMM Server instance.

### Remove PMM Server

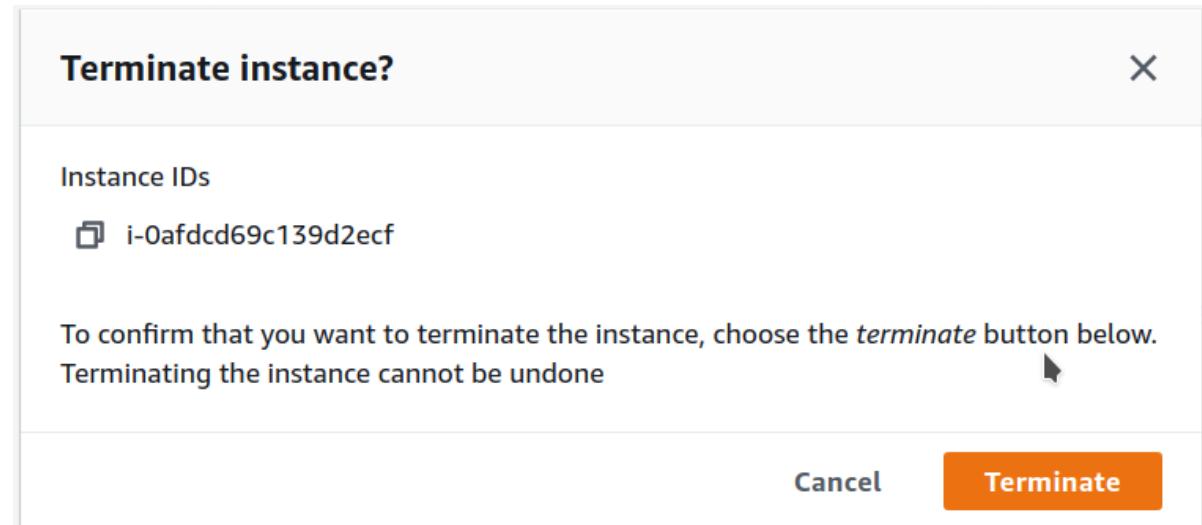
1. Find the instance in the EC2 Console

| Instances (1/1) <a href="#">Info</a>                                                                                                                    |         |                     |                         |               |                                   |                             |                   |                    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|---------|---------------------|-------------------------|---------------|-----------------------------------|-----------------------------|-------------------|--------------------|
| <a href="#">C</a> <a href="#">Connect</a> <a href="#">Instance state ▾</a> <a href="#">Actions ▾</a> <a href="#">Launch Instances</a> <a href="#">▼</a> |         |                     |                         |               |                                   |                             |                   |                    |
| <input type="text"/> Filter instances <a href="#">Clear filters</a>                                                                                     |         |                     |                         |               |                                   |                             |                   |                    |
| <input checked="" type="checkbox"/>                                                                                                                     | Name    | Instance ID         | Instance state          | Instance type | Status check                      | Alarm status                | Availability Zone | Public IPv4 DNS    |
| <input checked="" type="checkbox"/>                                                                                                                     | pmm-rds | i-0b23aebbb7268fb7c | <a href="#">Running</a> | m4.large      | <a href="#">2/2 checks passed</a> | No alarms <a href="#">+</a> | us-east-1b        | ec2-3-95-132-2.com |

2. Select "Instance state" menu and "Terminate instance"

| Instances (1/1) <a href="#">Info</a>                                                                                                                                                                                                                            |  |  |  |  |  |  |  |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--|--|--|--|--|--|
| <a href="#">C</a> <a href="#">Connect</a> <a href="#">Instance state ▾</a> <a href="#">Stop instance</a> <a href="#">Start instance</a> <a href="#">Hibernate instance</a> <a href="#">Reboot instance</a> <a href="#">Terminate instance</a> <a href="#">▼</a> |  |  |  |  |  |  |  |  |
| <a href="#">en address</a>                                                                                                                                                                                                                                      |  |  |  |  |  |  |  |  |
| <a href="#">Private IPv4 addresses</a> 10.77.1.242                                                                                                                                                                                                              |  |  |  |  |  |  |  |  |
| <a href="#">Private IPv4 DNS</a> ip-10-77-1-242.ec2.internal                                                                                                                                                                                                    |  |  |  |  |  |  |  |  |
| <a href="#">3.compute-1.amazonaws.com  </a>                                                                                                                                                                                                                     |  |  |  |  |  |  |  |  |

3. Confirm termination operation



 See also

[Improving Percona Monitoring and Management EC2 Instance Resilience Using CloudWatch Alarm Actions](#)

Last update: 2022-01-10

## 2.2.6 Easy-install script

### Caution

You can download and check `get-pmm.sh` before running it from our [github](#):

Linux or macOS

Using Curl:

```
curl -fsSL https://www.percona.com/get/pmm | /bin/bash
```

Using wget:

```
wget -O - https://www.percona.com/get/pmm | /bin/bash
```

This script:

- Installs Docker if it is not already installed on your system.
- Stops and backs up any PMM Server Docker containers that are currently running.
- Pulls and runs the latest PMM Server Docker image.
- Can run in Interactive mode to change the default settings:

```
curl -fsSL0 https://www.percona.com/get/pmm (or wget https://www.percona.com/get/pmm)
chmod +x pmm
./pmm --interactive
```

---

Last update: 2022-02-04

## 2.2.7 DBaaS

### Caution

DBaaS functionality is currently in [technical preview](#) and is subject to change.

### Software prerequisites

#### DOCKER

Red Hat, CentOS

```
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo  
yum -y install docker-ce  
usermod -a -G docker centos  
systemctl enable docker  
systemctl start docker
```

Debian, Ubuntu

```
apt-add-repository https://download.docker.com/linux/centos/docker-ce.repo  
systemctl enable docker  
systemctl start docker
```

#### MINIKUBE

Please install minikube 1.16.0

Red Hat, CentOS

```
yum -y install curl  
curl -Lo /usr/local/sbin/minikube https://github.com/kubernetes/minikube/releases/download/  
v1.16.0/minikube-linux-amd64  
chmod +x /usr/local/sbin/minikube  
ln -s /usr/local/sbin/minikube /usr/sbin/minikube  
alias kubectl='minikube kubectl --'
```

## Start PMM server and activate a DBaaS feature

- To start a fully-working 3 node XtraDB cluster, consisting of sets of 3x HAProxy, 3x PXC and 6x PMM Client containers, you will need at least 9 vCPU available for minikube. (1x vCPU for HAProxy and PXC and 0.5vCPU for each pmm-client containers).
- DBaaS does not depend on PMM Client.
- You can pass the environment variable `--env ENABLE_DBaaS=1` to force the DBaaS feature when starting up pmm-server container. **You can omit the variable and enable the feature later using PMM UI**, please follow the link in step 3. below.
- Add the option `--network minikube` if you run PMM Server and minikube in the same Docker instance. (This will share a single network and the kubeconfig will work.)
- Add the options `--env PMM_DEBUG=1` and/or `--env PMM_TRACE=1` if you need extended debug details

### 1. Start PMM server:

```
docker run --detach --publish 80:80 --publish 443:443 --name pmm-server percona/pmm-server:2
```

### 2. Change the default administrator credentials from CLI:

(This step is optional, because the same can be done from the web interface of PMM on first login.)

```
docker exec -t pmm-server bash -c 'ln -s /srv/grafana /usr/share/grafana/data; chown -R grafana:grafana /usr/share/grafana/data; grafana-cli --homepath /usr/share/grafana admin reset-admin-password <RANDOM_PASS_Goes_IN_HERE>'
```

#### **⚠ Important**

You must [activate DBaaS](#) using the PMM UI if you omitted `--env ENABLE_DBaaS=1` when starting up the container.

## Create a Kubernetes cluster

The DBaaS feature uses Kubernetes clusters to deploy database clusters. You must first create a Kubernetes cluster and then add it to PMM using `kubeconfig` to get a successful setup.

### MINIKUBE

#### 1. Configure and start minikube:

```
minikube config set cpus 16
minikube config set memory 32768
minikube config set kubernetes-version 1.20.1
minikube start
```

#### 2. Get your kubeconfig details from `minikube`. (You need these to register your Kubernetes cluster with PMM Server):

```
minikube kubectl -- config view --flatten --minify
```

You will need to copy this output to your clipboard and continue with [adding a Kubernetes cluster to PMM](#).

## AMAZON AWS EKS

1. Create your cluster via `eksctl` or the Amazon AWS interface. For example:

```
eksctl create cluster --write-kubeconfig --name=your-cluster-name --zones=us-west-2a,us-west-2b --kubeconfig <PATH_TO_KUBECONFIG>
```

2. When the cluster is running, modify your kubeconfig file, if it's not utilizing the `aws-iam-authenticator` or `client-certificate` method for authentication with Kubernetes. Here are two examples that you can use as templates to modify a copy of your existing kubeconfig:

- For the `aws-iam-authenticator` method:

```
---
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: << CERT_AUTH_DATA >>
    server: << K8S_CLUSTER_URL >>
    name: << K8S_CLUSTER_NAME >>
contexts:
- context:
    cluster: << K8S_CLUSTER_NAME >>
    user: << K8S_CLUSTER_USER >>
    name: << K8S_CLUSTER_NAME >>
current-context: << K8S_CLUSTER_NAME >>
kind: Config
preferences: {}
users:
- name: << K8S_CLUSTER_USER >>
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      command: aws-iam-authenticator
      args:
        - "token"
        - "-i"
        - "<< K8S_CLUSTER_NAME >>"
        - --region
        - << AWS_REGION >>
    env:
      - name: AWS_ACCESS_KEY_ID
        value: "<< AWS_ACCESS_KEY_ID >>"
      - name: AWS_SECRET_ACCESS_KEY
        value: "<< AWS_SECRET_ACCESS_KEY >>"
```

- For the `client-certificate` method:

```
---
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: << CERT_AUTH_DATA >>
    server: << K8S_CLUSTER_URL >>
    name: << K8S_CLUSTER_NAME >>
contexts:
- context:
    cluster: << K8S_CLUSTER_NAME >>
    user: << K8S_CLUSTER_USER >>
    name: << K8S_CLUSTER_NAME >>
current-context: << K8S_CLUSTER_NAME >>
kind: Config
preferences: {}
users:
- name: << K8S_CLUSTER_NAME >>
  user:
    client-certificate-data: << CLIENT_CERT_DATA >>
    client-key-data: << CLIENT_KEY_DATA >>
```

3. Follow the instructions on [How to add a Kubernetes cluster](#) with kubeconfig from the previous step.

If possible, the connection details will show the cluster's external IP (not possible with minikube).

#### GOOGLE GKE

1. Create your cluster either with [Google Cloud Console](#) or `gcloud` command line tool:

The command below assumes that your `gcloud` command line tool is properly configured and your user authenticated and authorized to manage GKE Clusters. This example creates a minimal zonal cluster using preemptive node machines, ideal for testing the DBaaS functionality.

```
gcloud container clusters create --zone europe-west3-c pmm-dbaas-cluster --cluster-version 1.19 --machine-type e2-standard-4 --preemptible --num-nodes=3
gcloud container clusters get-credentials pmm-dbaas-cluster --zone=europe-west3-c
kubectl create clusterrolebinding cluster-admin-binding --clusterrole=cluster-admin --user=<><your_user@your_company.com>>
```

2. Create `ServiceAccount`, `ClusterRole` and `RoleBindings` (required Roles are deployed automatically when PMM deploys Operators) using the following command:

```
cat <<EOF | kubectl apply -f -
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: percona-dbaas-cluster-operator
---
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: service-account-percona-server-dbaas-xtradb-operator
subjects:
- kind: ServiceAccount
  name: percona-dbaas-cluster-operator
roleRef:
  kind: Role
  name: percona-xtradb-cluster-operator
  apiGroup: rbac.authorization.k8s.io
---
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: service-account-percona-server-dbaas-psmdb-operator
subjects:
- kind: ServiceAccount
  name: percona-dbaas-cluster-operator
roleRef:
  kind: Role
  name: percona-server-mongodb-operator
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  name: service-account-percona-server-dbaas-admin
rules:
- apiGroups: ["*"]
  resources: ["*"]
  verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
```

```

name: service-account-percona-server-dbaas-operator-admin
subjects:
- kind: ServiceAccount
  name: percona-dbaas-cluster-operator
  namespace: default
roleRef:
  kind: ClusterRole
  name: service-account-percona-server-dbaas-admin
  apiGroup: rbac.authorization.k8s.io
EOF

```

3. Extract variables required to generate a kubeconfig:

```

name=`kubectl get serviceAccounts percona-dbaas-cluster-operator -o json | jq -r '.secrets[].name'`
certificate=`kubectl get secret $name -o json | jq -r '.data."ca.crt"'```
token=`kubectl get secret $name -o json | jq -r '.data.token' | base64 -d````
server=`kubectl cluster-info | grep 'Kubernetes master' | cut -d ' ' -f 6````

```

4. Generate your kubeconfig file (copy the output):

```

echo "
apiVersion: v1
kind: Config
users:
- name: percona-dbaas-cluster-operator
  user:
    token: $token
clusters:
- cluster:
    certificate-authority-data: $certificate
    server: $server
    name: self-hosted-cluster
contexts:
- context:
    cluster: self-hosted-cluster
    user: percona-dbaas-cluster-operator
    name: svcs-acct-context
current-context: svcs-acct-context
"

```

5. Follow the instructions on [How to add a Kubernetes cluster](#) with kubeconfig from the previous step.

## Deleting clusters

If a Public Address is set in PMM Settings, for each DB cluster an API Key is created which can be found on the page [/graph/org/apikeys](#). You should not delete them (for now, until [issue PMM-8045](#) is fixed) – once a DB cluster is removed from DBaaS, the related API Key is also removed.

For example, if you only run `eksctl delete cluster` to delete an Amazon EKS cluster without cleaning up the cluster first, there will be a lot of orphaned resources such as Cloud Formations, Load Balancers, EC2 instances, Network interfaces, etc. The same applies for Google GKE clusters.

### CLEANING UP KUBERNETES CLUSTER

1. You should delete all database clusters, backups and restores.

```

kubectl delete perconaxtradbclusterbackups.pxc.percona.com --all
kubectl delete perconaxtradbclusters.pxc.percona.com --all

```

```
kubectl delete perconaxtradbclusterrestores.pxc.percona.com --all
kubectl delete perconaservermongodbbackups.psmdb.percona.com --all
kubectl delete perconaservermongodbs.psmdb.percona.com --all
kubectl delete perconaservermongodbrestores.psmdb.percona.com --all
```

2. In the `dbaas-controller` repository, in the `deploy` directory there are manifests we use to deploy operators. Use them to delete operators and related resources from the cluster.

 **Important**

- Do NOT execute this step before all database clusters, backups and restores are deleted in the previous step. It may result in not being able to delete the namespace DBaaS lives in.
- Also be careful with this step if you are running DBaaS in more than one namespace as it deletes cluster level CustomResourceDefinitions needed to run DBaaS. This would break DBaaS in other namespaces. Delete just operators deployments in that case.

```
# Delete the PXC operator and related resources.
curl https://raw.githubusercontent.com/percona-platform/dbaas-controller/
7a5fff023994cecf6bde15705365114004b50b41/deploy/pxc-operator.yaml | kubectl delete -f -
# Delete the PSMDB operator and related resources.
curl https://raw.githubusercontent.com/percona-platform/dbaas-controller/
7a5fff023994cecf6bde15705365114004b50b41/deploy/psmdb-operator.yaml | kubectl delete -f -
```

3. Delete the name space where the DBaaS is running, this will delete all remaining name space level resources if any are left.

```
kubectl delete namespace <your-namespace>
```

4. Delete the Kubernetes cluster. The method depends on your cloud provider:

- Delete GKE cluster.
- Delete Amazon EKS cluster.

### Run PMM Server as a Docker container for DBaaS

1. Start PMM server from a feature branch:

```
docker run --detach --name pmm-server --publish 80:80 --publish 443:443 --env ENABLE_DBaaS=1
percona/pmm-server:2;
```

 **Important**

- Use `--network minikube` if running PMM Server and minikube in the same Docker instance. This way they will share single network and the kubeconfig will work.
- Use Docker variables `--env PMM_DEBUG=1 --env PMM_TRACE=1` to see extended debug details.

2. Change the default administrator credentials:

This step is optional, because the same can be done from the web interface of PMM on the first login.

```
docker exec -t pmm-server bash -c 'ln -s /srv/grafana /usr/share/grafana/data; chown -R grafana:grafana /usr/share/grafana/data; grafana-cli --homepath /usr/share/grafana admin reset-admin-password <RANDOM_PASS_Goes_IN_HERE>'
```

3. Set the public address for PMM Server in PMM settings UI
4. Follow the steps for [Add a Kubernetes cluster](#).
5. Follow the steps for [Add a DB Cluster](#).
6. Get the IP address to connect your app/service:

```
minikube kubectl get services
```

### Exposing PSMDB and XtraDB clusters for access by external clients

To make services visible externally, you create a LoadBalancer service or manually run commands to expose ports:

```
kubectl expose deployment hello-world --type=NodePort.
```

#### See also

- [DBaaS Dashboard](#)
- [Install minikube](#)
- [Setting up a Standalone MYSQL Instance on Kubernetes & exposing it using Nginx Ingress Controller](#)
- [Use a Service to Access an Application in a Cluster](#)
- [Exposing applications using services](#)

---

Last update: 2021-07-30

## 2.3 Client

### 2.3.1 Set up PMM Client

There are different ways to install PMM Client on a node and register it with PMM Server. Choose from:

- **Docker:** Run PMM Client as a Docker container, either directly or with Docker compose.
- **Package manager:**
  - On Debian or Red Hat Linux, install `percona-release` and use a Linux package manager (`apt` / `dnf`) to install PMM Client.
  - On Debian or Red Hat, download `.deb` / `.rpm` PMM Client packages and manually install them.



#### Binary is only way to install PMM client without root permissions

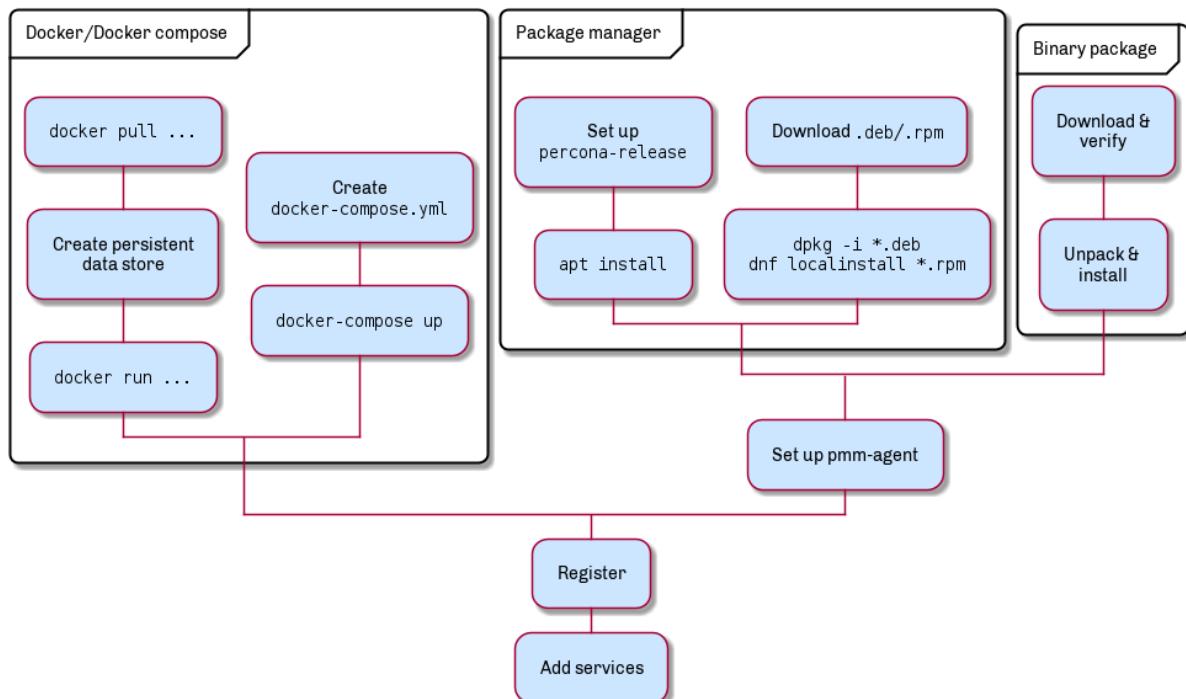
- **Binary package:** For other Linux distributions, download and unpack generic PMM Client Linux binaries.

When you have installed PMM Client, you must:

- Register the node with PMM Server.
- Configure and add services according to type.

If you need to, you can [unregister](#), [remove services](#) or [remove PMM Client](#).

Here's an overview of the choices.



## Before you start

- Set up PMM Server with a known IP address accessible from the client node.
- You have superuser (root) access on the client host.
- You have superuser access to any database servers that you want to monitor.
- These Linux packages are installed: `curl`, `gnupg`, `sudo`, `wget`.
- If using it, install Docker.
- If using it, install Docker compose.
- System requirements:
  - Operating system – PMM Client runs on any modern 64-bit Linux distribution. It is tested on supported versions of Debian, Ubuntu, CentOS, and Red Hat Enterprise Linux. (See [Percona software support life cycle](#)).
  - Disk – A minimum of 100 MB of storage is required for installing the PMM Client package. With a good connection to PMM Server, additional storage is not required. However, the client needs to store any collected data that it cannot dispatch immediately, so additional storage may be required if the connection is unstable or the throughput is low. (Caching only applies to Query Analytics data; VictoriaMetrics data is never cached on the client side.)

## Install

### DOCKER

The [PMM Client Docker image](#) is a convenient way to run PMM Client as a preconfigured Docker container.

1. Pull the PMM Client docker image.

```
docker pull \
percona/pmm-client:2
```

2. Use the image as a template to create a persistent data store that preserves local data when the image is updated.

```
docker create \
--volume /srv \
--name pmm-client-data \
percona/pmm-client:2 /bin/true
```

3. Run the container to start [PMM Agent](#) in setup mode. Set `X.X.X.X` to the IP address of your PMM Server. (Do not use the `docker --detach` option as PMM agent only logs to the console.)

```
PMM_SERVER=X.X.X.X:443
docker run \
--rm \
--name pmm-client \
-e PMM_AGENT_SERVER_ADDRESS=${PMM_SERVER} \
-e PMM_AGENT_SERVER_USERNAME=admin \
-e PMM_AGENT_SERVER_PASSWORD=admin \
-e PMM_AGENT_SERVER_INSECURE_TLS=1 \
-e PMM_AGENT_SETUP=1 \
-e PMM_AGENT_CONFIG_FILE=pmm-agent.yml \
--volumes-from pmm-client-data \
percona/pmm-client:2
```

### Tips

You can find a complete list of compatible environment variables [here](#).

1. Check status.

```
docker exec pmm-client \
pmm-admin status
```

In the PMM user interface you will also see an increase in the number of monitored nodes.

You can now add services with `pmm-admin` by prefixing commands with `docker exec pmm-client`.

### Tips

- Adjust host firewall and routing rules to allow Docker communications. ([Read more](#))
- For help: `docker run --rm percona/pmm-client:2 --help`

#### DOCKER COMPOSE

1. Copy and paste this text into a file called `docker-compose.yml`.

```
version: '2'
services:
  pmm-client:
    image: percona/pmm-client:2
    hostname: pmm-client-myhost
    container_name: pmm-client
    restart: always
    ports:
      - "42000:42000"
      - "42001:42001"
    logging:
      driver: json-file
      options:
        max-size: "10m"
        max-file: "5"
    volumes:
      - ./pmm-agent.yaml:/etc/pmm-agent.yaml
      - pmm-client-data:/srv
    environment:
      - PMM_AGENT_CONFIG_FILE=/etc/pmm-agent.yaml
      - PMM_AGENT_SERVER_USERNAME=admin
      - PMM_AGENT_SERVER_PASSWORD=admin
      - PMM_AGENT_SERVER_ADDRESS=X.X.X.X:443
      - PMM_AGENT_SERVER_INSECURE_TLS=true
    entrypoint: pmm-agent setup
  volumes:
    pmm-client-data:
```

- Check the values in the `environment` section match those for your PMM Server. (X.X.X.X is the IP address of your PMM Server.)
- Use unique hostnames across all PMM Clients (value for `services.pmm-client.hostname`).

2. Ensure a writable agent configuration file.

```
touch pmm-agent.yaml && chmod 0666 pmm-agent.yaml
```

3. Run the PMM Agent setup. This will run and stop.

```
docker-compose up
```

4. Edit `docker-compose.yml`, comment out the `entrypoint` line (insert a `#`) and save.

```
...
#       entrypoint: pmm-agent setup
```

5. Run again, this time with the Docker *detach* option.

```
docker-compose up -d
```

6. Verify.

On the command line.

```
docker exec pmm-client pmm-admin status
```

In the GUI.

- Select  *PMM Dashboards* →  *System (Node)* →  *Node Overview*.
- In the *Node Names* menu, select the new node.
- Change the time range to see data.

### Danger

`pmm-agent.yaml` contains sensitive credentials and should not be shared.

## PACKAGE MANAGER

### Tip

If you have used `percona-release` before, disable and re-enable the repository:

```
percona-release disable all
percona-release enable original release
```

### Debian-based

1. Configure repositories.

```
wget https://repo.percona.com/apt/percona-release_latest.generic_all.deb
dpkg -i percona-release_latest.generic_all.deb
```

2. Install the PMM Client package.

### Root permissions

```
apt update
apt install -y pmm2-client
```

3. Check.

```
pmm-admin --version
```

4. Register the node.

Red Hat-based

1. Configure repositories.

```
yum install -y https://repo.percona.com/yum/percona-release-latest.noarch.rpm
```

2. Install the PMM Client package.

```
yum install -y pmm2-client
```

3. Check.

```
pmm-admin --version
```

4. Register the node.

PACKAGE MANAGER – MANUAL DOWNLOAD

1. Visit the [Percona Monitoring and Management 2 download](#) page.
2. Under *Version*: select the one you want (usually the latest).
3. Under *Software*: select the item matching your software platform.
4. Click to download the package file:

- For Debian, Ubuntu: [.deb](#)
- For Red Hat, CentOS, Oracle Linux: [.rpm](#)

(Alternatively, copy the link and use `wget` to download it.)

Here are the download page links for each supported platform.

- [Debian 9 \(Stretch\)](#)
- [Debian 10 \(Buster\)](#)
- [Red Hat/CentOS/Oracle 7](#)
- [Red Hat/CentOS/Oracle 8](#)
- [Ubuntu 16.04 \(Xenial Xerus\)](#)
- [Ubuntu 18.04 \(Bionic Beaver\)](#)
- [Ubuntu 20.04 \(Focal Fossa\)](#)

Debian-based

```
dpkg -i *.deb
```

Red Hat-based

```
dnf localinstall *.rpm
```

#### BINARY PACKAGE

1. Download the PMM Client package:

```
wget https://downloads.percona.com/downloads/pmm2/2.26.0/binary/tarball/pmm2-client-2.26.0.tar.gz
```

2. Download the PMM Client package checksum file:

```
wget https://downloads.percona.com/downloads/pmm2/2.26.0/binary/tarball/pmm2-client-2.26.0.tar.gz.sha256sum
```

3. Verify the download.

```
sha256sum -c pmm2-client-2.26.0.tar.gz.sha256sum
```

4. Unpack the package and move into the directory.

```
tar xfz pmm2-client-2.26.0.tar.gz && cd pmm2-client-2.26.0
```

5. Choose one of these two commands (depends on your permissions):

#### Without root permissions

```
export PMM_DIR=YOURPATH
```

where YOURPATH replace with you real path, where you have required access.

#### With root permissions

```
export PMM_DIR=/usr/local/percona/pmm2
```

6. Run the installer.

#### Root permissions (if you skipped step 5 for non root users)

```
./install_tarball
```

7. Change the path.

```
PATH=$PATH:$PMM_DIR/bin
```

8. Set up the agent (pick the command for you depending on permissions)

### Root permissions

```
pmm-agent setup --config-file=/usr/local/percona/pmm2/config/pmm-agent.yaml --server-address=192.168.1.123 --server-insecure-tls --server-username=admin --server-password=admin
```

### Non root users

```
pmm-agent setup --config-file=${PMM_DIR}/config/pmm-agent.yaml --server-address=192.168.1.123 --server-insecure-tls --server-username=admin --server-password=admin --paths-tempdir=${PMM_DIR}/tmp --paths-base=${PMM_DIR}
```

9. Run the agent.

```
pmm-agent --config-file=${PMM_DIR}/config/pmm-agent.yaml
```

10. Open a new terminal and check.

```
pmm-admin status
```

!!! hint PMM-Agent can be updated from tarball:

- Download tar.gz with pmm2-client.
- Extract it.
- Run ./install\_tarball script with the “-u” flag.

The configuration file will be overwritten if you do not provide the “-u” flag while the pmm-agent is updated.

## Register

Register your client node with PMM Server.

```
pmm-admin config --server-insecure-tls --server-url=https://admin:admin@X.X.X.X:443
```

- X.X.X.X is the address of your PMM Server.
- 443 is the default port number.
- admin / admin is the default PMM username and password. This is the same account you use to log into the PMM user interface, which you had the option to change when first logging in.

### Important

Clients *must* be registered with the PMM Server using a secure channel. If you use http as your server URL, PMM will try to connect via https on port 443. If a TLS connection can't be established you will get an error and you must use https along with the appropriate secure port.

## EXAMPLES

Register on PMM Server with IP address 192.168.33.14 using the default admin/admin username and password, a node with IP address 192.168.33.23, type generic, and name mynode.

```
pmm-admin config --server-insecure-tls --server-url=https://admin:admin@192.168.33.14:443 192.168.33.23 generic mynode
```

## Add services

You must configure and adding services according to the service type.

- [MySQL](#) (and variants Percona Server for MySQL, Percona XtraDB Cluster, MariaDB)
- [MongoDB](#)
- [PostgreSQL](#)
- [ProxySQL](#)
- [Amazon RDS](#)
- [Microsoft Azure](#)
- [Google Cloud Platform](#) (MySQL and PostgreSQL)
- [Linux](#)
- [External services](#)
- [HAProxy](#)
- [Remote instances](#)



### Tip

To change the parameters of a previously-added service, remove the service and re-add it with new parameters.

## Remove

How to remove (uninstall) PMM Client.

DOCKER



### Caution

These steps delete the PMM Client Docker image and client services configuration data.

1. Stop pmm-client container.

```
docker stop pmm-client
```

2. Remove containers.

```
docker rm pmm-client
```

3. Remove the image.

```
docker rmi $(docker images | grep "percona/pmm-client" | awk {'print $3'})
```

4. Remove the volume.

```
docker volume rm pmm-client-data
```

**PACKAGE MANAGER****Debian-based distributions**

1. Uninstall the PMM Client package.

```
apt remove -y pmm2-client
```

2. Remove the Percona repository

```
dpkg -r percona-release
```

**Red Hat-based distributions**

1. Uninstall the PMM Client package.

```
yum remove -y pmm2-client
```

2. Remove the Percona repository

```
yum remove -y percona-release
```

**Unregister**

How to unregister PMM Client from PMM Server.

```
pmm-admin unregister --force
```

All services monitored by this node will be removed from monitoring.

**Remove services**

You must specify the service type and service name to remove services from monitoring.

```
pmm-admin remove <service-type> <service-name>
```

**service-type**

One of `mysql`, `mongodb`, `postgresql`, `proxysql`, `haproxy`, `external`.

 **See also**

- [Percona release](#)
- [PMM Client architecture](#)
- Thanks to [paskal](#) for original Docker compose files

---

Last update: 2022-02-08

## 2.3.2 MySQL

How to set up PMM to monitor a MySQL or MySQL-based database instance.

PMM Client collects metrics from [MySQL](#), [Percona Server for MySQL](#), [Percona XtraDB Cluster](#), and [MariaDB](#). (Amazon RDS is also supported and explained in a [separate section](#).)

### Summary

- Create PMM account and set permissions.
- Choose a data source:
  - Slow query log, or,
  - Performance Schema.
- Configure:
  - Query response time,
  - Tablestats,
  - User statistics.
- Add service.
- Check service.

### Before you start

Check that:

- [PMM Server is installed](#) and running with a known IP address accessible from the client node.
- [PMM Client is installed](#) and the [node is registered with PMM Server](#).
- You have superuser (root) access on the client host.

### Create a database account for PMM

It is good practice to use a non-superuser account to connect PMM Client to the monitored database instance. This example creates a database user with name `pmm`, password `pass`, and the necessary permissions.

```
CREATE USER 'pmm'@'localhost' IDENTIFIED BY 'pass' WITH MAX_USER_CONNECTIONS 10;
GRANT SELECT, PROCESS, REPLICATION CLIENT, RELOAD, BACKUP_ADMIN ON *.* TO 'pmm'@'localhost';
```

### Choose and configure a source

Decide which source of metrics to use, and configure your database server for it. The choices are [Slow query log](#) and [Performance Schema](#).

While you can use both at the same time we recommend using only one—there is some overlap in the data reported, and each incurs a small performance penalty. The choice depends on the version and variant of your MySQL instance, and how much detail you want to see.

Here are the benefits and drawbacks of *Slow query log* and *Performance Schema* metrics sources.

|                           | <b>Benefits</b>                                                                                        | <b>Drawbacks</b>                                                                                                                           |
|---------------------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Slow query log</b>     | 1. More detail.<br>2. Lower resource impact (with query sampling feature in Percona Server for MySQL). | 1. PMM Client must be on same host as database server or have access to slow query log.<br>2. Log files grow and must be actively managed. |
| <b>Performance Schema</b> | 1. Faster parsing.<br>2. Enabled by default on newer versions of MySQL.                                | 1. Less detail.                                                                                                                            |

#### DATA SOURCE RECOMMENDATIONS

| <b>Database server</b>   | <b>Versions</b> | <b>Recommended source</b> |
|--------------------------|-----------------|---------------------------|
| MySQL                    | 5.1-5.5         | Slow query log            |
| MySQL                    | 5.6+            | Performance Schema        |
| MariaDB                  | 10.0+           | Performance Schema        |
| Percona Server for MySQL | 5.7, 8.0        | Slow query log            |
| Percona XtraDB Cluster   | 5.6, 5.7, 8.0   | Slow query log            |

#### SLOW QUERY LOG

This section covers how to configure a MySQL-based database server to use the *slow query log* as a source of metrics.

#### APPLICABLE VERSIONS

| <b>Server</b>            | <b>Versions</b>  |
|--------------------------|------------------|
| MySQL                    | 5.1-5.5          |
| MariaDB                  | 10.1.2+          |
| Percona Server for MySQL | 5.7.10+, 8.0.12+ |
| Percona XtraDB Cluster   | 5.6, 5.7, 8.0    |

The *slow query log* records the details of queries that take more than a certain amount of time to complete. With the database server configured to write this information to a file rather than a table, PMM Client parses the file and sends aggregated data to PMM Server via the Query Analytics part of PMM Agent.

## SETTINGS

| Variable                               | Value  | Description                                                                                                                                                                                                      |
|----------------------------------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>slow_query_log</code>            | ON     | Enables the slow query log.                                                                                                                                                                                      |
| <code>log_output</code>                | 'FILE' | Ensures the log is sent to a file. (This is the default on MariaDB.)                                                                                                                                             |
| <code>long_query_time</code>           | 0      | The slow query threshold in seconds. In heavily-loaded applications, many quick queries can affect performance more than a few slow ones. Setting this value to <code>0</code> ensures all queries are captured. |
| <code>log_slow_admin_statements</code> | ON     | Includes the logging of slow administrative statements.                                                                                                                                                          |
| <code>log_slow_slave_statements</code> | ON     | Enables logging for queries that have taken more than <code>long_query_time</code> seconds to execute on the replica.                                                                                            |

## Examples

- Configuration file.

```
slow_query_log=ON
log_output=FILE
long_query_time=0
log_slow_admin_statements=ON
log_slow_slave_statements=ON
```

- Session.

```
SET GLOBAL slow_query_log = 1;
SET GLOBAL log_output = 'FILE';
SET GLOBAL long_query_time = 0;
SET GLOBAL log_slow_admin_statements = 1;
SET GLOBAL log_slow_slave_statements = 1;
```

## Slow query log – extended

Some MySQL-based database servers support extended slow query log variables.

## Applicable versions

| Server                   | Versions         |
|--------------------------|------------------|
| Percona Server for MySQL | 5.7.10+, 8.0.12+ |
| Percona XtraDB Cluster   | 5.6, 5.7, 8.0    |
| MariaDB                  | 10.0             |

## Settings

| Variable                                       | Value   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>log_slow_rate_limit</code>               | 100     | Defines the rate of queries captured by the <i>slow query log</i> . A good rule of thumb is 100 queries logged per second. For example, if your Percona Server instance processes 10,000 queries per second, you should set <code>log_slow_rate_limit</code> to <code>100</code> and capture every 100 <sup>th</sup> query for the <i>slow query log</i> . Depending on the amount of traffic, logging could become aggressive and resource consuming. This variable throttles the level of intensity of the data capture without compromising information. |
| <code>log_slow_rate_type</code>                | 'query' | Set so that it applies to queries, rather than sessions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>slow_query_log_always_write_time</code>  | 1       | Specifies which queries should ignore sampling. With query sampling this ensures that queries with longer execution time will always be captured by the slow query log, avoiding the possibility that infrequent slow queries might not get captured at all.                                                                                                                                                                                                                                                                                                |
| <code>log_slow_verbosity</code>                | 'full'  | Ensures that all information about each captured query is stored in the slow query log.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>slow_query_log_use_global_control</code> | 'all'   | Configure the slow query log during runtime and apply these settings to existing connections. (By default, slow query log settings apply only to new sessions.)                                                                                                                                                                                                                                                                                                                                                                                             |

## Examples

- Configuration file (Percona Server for MySQL, Percona XtraDB Cluster).

```
log_slow_rate_limit=100
log_slow_rate_type='query'
slow_query_log_always_write_time=1
log_slow_verbosity='full'
slow_query_log_use_global_control='all'
```

- Configuration file (MariaDB).

```
log_slow_rate_limit=100
```

- Session (Percona Server for MySQL, Percona XtraDB Cluster).

```
SET GLOBAL log_slow_rate_limit = 100;
SET GLOBAL log_slow_rate_type = 'query';
SET GLOBAL slow_query_log_always_write_time = 1;
SET GLOBAL log_slow_verbosity = 'full';
SET GLOBAL slow_query_log_use_global_control = 'all';
```

## Slow query log rotation

Slow query log files can grow quickly and must be managed.

When adding a service with the command line use the `pmm-admin` option `--size-slow-logs` to set at what size the slow query log file is rotated. (The size is specified as a number with a suffix. See `pmm-admin add mysql`.)

When the limit is reached, PMM Client will:

- remove the previous `.old` slow log file,
- rename the current file by adding the suffix `.old`,
- execute the MySQL command `FLUSH LOGS`.

Only one `.old` file is kept. Older ones are deleted.

You can manage log rotation yourself, for example, with `logrotate`. If you do, you can disable PMM Client's log rotation by providing a negative value to `--size-slow-logs` option when adding a service with `pmm-admin add`.

#### PERFORMANCE SCHEMA

This section covers how to configure a MySQL-based database server to use *Performance Schema* as a source of metrics.

##### Applicable versions

| Server                   | Versions      |
|--------------------------|---------------|
| Percona Server for MySQL | 5.6, 5.7, 8.0 |
| Percona XtraDB Cluster   | 5.6, 5.7, 8.0 |
| MariaDB                  | 10.3+         |

PMM's [MySQL Performance Schema Details](#) dashboard charts the various `performance_schema` metrics.

To use *Performance Schema*, set these variables.

| Variable                                                   | Value                         | Description                                                                               |
|------------------------------------------------------------|-------------------------------|-------------------------------------------------------------------------------------------|
| <code>performance_schema</code>                            | <code>ON</code>               | Enables <i>Performance Schema</i> metrics. This is the default in MySQL 5.6.6 and higher. |
| <code>performance-schema-instrument</code>                 | <code>'statement/%=ON'</code> | Configures Performance Schema instruments.                                                |
| <code>performance-schema-consumer-statements-digest</code> | <code>ON</code>               | Configures the <code>statements-digest</code> consumer.                                   |
| <code>innodb_monitor_enable</code>                         | <code>all</code>              | Enables InnoDB metrics counters.                                                          |

##### Examples

- Configuration file.

```
performance_schema=ON
performance-schema-instrument='statement/%=ON'
performance-schema-consumer-statements-digest=ON
innodb_monitor_enable=all
```

- Session.

( `performance_schema` cannot be set in a session and must be set at server start-up.)

```
UPDATE performance_schema.setup_consumers
SET ENABLED = 'YES' WHERE NAME LIKE '%statements%';
SET GLOBAL innodb_monitor_enable = all;
```

#### MariaDB 10.5.7 or lower

There is no *Explain* or *Example* data shown by default in Query Analytics when monitoring MariaDB instances version 10.5.7 or lower. A workaround is to set this variable.

| Variable                                          | Value                      | Description                          |
|---------------------------------------------------|----------------------------|--------------------------------------|
| <code>performance_schema.setup_instruments</code> | <code>'statement/%'</code> | List of instrumented object classes. |

- Session.

```
UPDATE performance_schema.setup_instruments SET ENABLED = 'YES', TIMED = 'YES' WHERE NAME LIKE 'statement/%';
UPDATE performance_schema.setup_consumers SET ENABLED = 'YES' WHERE NAME LIKE '%statements%';
```

- Transactions

MariaDB doesn't implement queries history for transactions. All queries executed within a transaction won't have query examples since PMM relies on the `performance_schema.events_statements_history` to grab the query example but that table won't have any query executed as part of a transaction.

This behavior is because MariaDB doesn't implement these consumers:

```
events_transactions_current
events_transactions_history
events_transactions_history_long
```

#### Query response time

*Query time distribution* is a chart in the [Details tab of Query Analytics](#) showing the proportion of query time spent on various activities. It is enabled with the `query_response_time_stats` variable and associated plugins.

##### APPLICABLE VERSIONS

| Server                   | Versions                                                  |
|--------------------------|-----------------------------------------------------------|
| Percona Server for MySQL | 5.7 ( <a href="#">not Percona Server for MySQL 8.0.</a> ) |
| MariaDB                  | 10.0.4                                                    |

Set this variable to see query time distribution charts.

| Variable                               | Value | Description                                                                                  |
|----------------------------------------|-------|----------------------------------------------------------------------------------------------|
| <code>query_response_time_stats</code> | ON    | Report <i>query response time distributions</i> . (Requires plugin installation. See below.) |

- Configuration file.

```
query_response_time_stats=ON
```

You must also install the plugins.

- Session.
  - a. Check that `/usr/lib/mysql/plugin/query_response_time.so` exists.
  - b. Install the plugins and activate.

For MariaDB 10.3:

```
INSTALL PLUGIN QUERY_RESPONSE_TIME_AUDIT SONAME 'query_response_time.so';
INSTALL PLUGIN QUERY_RESPONSE_TIME SONAME 'query_response_time.so';
SET GLOBAL query_response_time_stats = ON;
```

For Percona Server for MySQL 5.7:

```
INSTALL PLUGIN QUERY_RESPONSE_TIME_AUDIT SONAME 'query_response_time.so';
INSTALL PLUGIN QUERY_RESPONSE_TIME SONAME 'query_response_time.so';
INSTALL PLUGIN QUERY_RESPONSE_TIME_READ SONAME 'query_response_time.so';
INSTALL PLUGIN QUERY_RESPONSE_TIME_WRITE SONAME 'query_response_time.so';
SET GLOBAL query_response_time_stats = ON;
```

## Tablestats

Some table metrics are automatically disabled when the number of tables exceeds a default limit of 1000 tables. This prevents PMM Client from affecting the performance of your database server.

The limit can be changed [when adding a service on the command line](#) with the two `pmm-admin` options:

| <code>pmm-admin</code> option             | Description                                                                                                                                                                                       |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>--disable-tablestats</code>         | Disables tablestats collection when the default limit is reached.                                                                                                                                 |
| <code>--disable-tablestats-limit=N</code> | Sets the number of tables ( <code>N</code> ) for which tablestats collection is disabled. 0 means no limit. A negative number means tablestats is completely disabled (for any number of tables). |

## User statistics

### APPLICABLE VERSIONS

User activity, individual table and index access details are shown on the [MySQL User Details](#) dashboard when the `userstat` variable is set.

| Server                   | Versions      |
|--------------------------|---------------|
| Percona Server for MySQL | 5.6, 5.7, 8.0 |
| Percona XtraDB Cluster   | 5.6, 5.7, 8.0 |
| MariaDB                  | 5.2.0+        |

### EXAMPLES

- Configuration file.

```
userstat=ON
```

- Session.

```
SET GLOBAL userstat = ON;
```

## Add service

When you have configured your database server, you can add a MySQL service with the user interface or on the command line.

When adding a service with the command line, you must use the `pmm-admin --query-source=SOURCE` option to match the source you've chosen and configured the database server for.

With the PMM user interface, you select *Use performance schema*, or deselect it to use *slow query log*.

### WITH THE USER INTERFACE

1. Select *Configuration* →  *PMM Inventory* →  *Add Instance*.
2. Select *MySQL - Add a remote instance*.
3. Enter or select values for the fields.
4. Click *Add service*.

The screenshot shows the 'Add Instance' configuration dialog for a MySQL instance. The interface is dark-themed with light-colored input fields.

**Main details**

- Hostname**: Hostname
- Service name**: Service name (default: Hostname)
- Port**: 3306
- Username**: Username
- Password**: Password

**Labels**

- Environment**: Environment
- Region**: Region
- Availability Zone**: Availability Zone
- Replication set**: Replication set
- Cluster**: Cluster

**Custom labels**

"Custom labels Format: key1:value1 key2:value2"

**Additional options**

- Skip connection check
- Use TLS for database connections
- Skip TLS certificate and hostname validation

**Table statistics limit**

- Disabled
- Default
- Custom

1000

Use performance schema

**Add service** | [Return to menu](#)

If your MySQL instance is configured to use TLS, click on the *Use TLS for database connections* check box and fill in your TLS certificates and key.

### Additional options

Skip connection check

Use TLS for database connections

TLS CA ⓘ

TLS certificate key ⓘ

TLS certificate ⓘ

Skip TLS certificate and hostname validation

#### ON THE COMMAND LINE

Add the database server as a service using one of these example commands. If successful, PMM Client will print MySQL Service added with the service's ID and name. Use the `--environment` and `-custom-labels` options to set tags for the service to help identify them.

#### EXAMPLES

##### TLS connection

```
pmm-admin add mysql --username=user --password=pass --tls --tls-skip-verify --tls-ca=pathtoca.pem --tls-cert=pathtocert.pem --tls-key=pathtocertkey.pem --server-url=http://admin:admin@127.0.0.1 --query-source=perfschema name localhost:3306
```

#### Slow query log

Default query source (`slowlog`), service name (`{node_name}-mysql`), and service address/port (`127.0.0.1:3306`), with database server account `pmm` and password `pass`.

```
pmm-admin add mysql --username=pmm --password=pass
```

Slow query log source and log size limit (1 gigabyte), service name (`MYSQL_NODE`) and service address/port (`192.168.1.123:3306`).

```
pmm-admin add mysql --query-source=slowlog --size-slow-logs=1GiB --username=pmm --password=pass MYSQL_NODE 192.168.1.123:3306
```

Slow query log source, disabled log management (use `logrotate` or some other log management tool), service name (`MYSQL_NODE`) and service address/port (`192.168.1.123:3306`).

```
pmm-admin add mysql --query-source=slowlog --size-slow-logs=-1GiB --username=pmm --password=pass MYSQL_NODE 192.168.1.123:3306
```

Default query source (`slowlog`), service name (`{node}-mysql`), connect via socket.

```
pmm-admin add mysql --username=pmm --password=pass --socket=/var/run/mysqld/mysqld.sock
```

#### Performance Schema

Performance schema query source, service name (`MYSQL_NODE`) and default service address/port (`127.0.0.1:3306`).

```
pmm-admin add mysql --query-source=perfschema --username=pmm --password=pass MYSQL_NODE
```

Performance schema query source, service name (`MYSQL_NODE`) and default service address/port (`127.0.0.1:3306`) specified with flags.

```
pmm-admin add mysql --query-source=perfschema --username=pmm --password=pass --service-name=MYSQL_NODE --host=127.0.0.1 --port=3306
```

#### Identifying services

Default query source (`slowlog`), environment labeled `test`, custom labels setting `source` to `slowLog`. (This example uses positional parameters for service name and service address.)

```
pmm-admin add mysql --environment=test --custom-labels='source=slowlog' --username=root --password=password --query-source=slowlog MyQLSlowLog localhost:3306
```

#### Check the service

##### PMM USER INTERFACE

1. Select  Configuration →  PMM Inventory → Inventory list.
2. Look in the Services tab for a matching Service Type (MySQL), Service name, Addresses, and any other details entered in the form.

3. Look in the *Agents* tab to check the desired data source is being used.

#### COMMAND LINE

Look for your service in the output of this command.

```
pmm-admin inventory list services --service-type=mysql
```

#### CHECK DATA

1. Open the *MySQL Instance Summary* dashboard.
2. Set the *Service Name* to the newly-added service.

#### Percona Server for MySQL, MariaDB

If query response time plugin was installed, check for data in the *MySQL Query Response Time Details* dashboard or select a query in *PMM Query Analytics* to see the *Query time distribution* bar.

#### Percona XtraDB Cluster

Open the *PXC/Galera Cluster Summary* dashboard.



#### See also

- [Percona Server for MySQL – Slow Query Log Extended](#)
- [Percona Server for MySQL – User Statistics](#)
- [MariaDB – Slow Query Log Overview](#)
- [MariaDB – Slow Query Log Extended Statistics](#)
- [MariaDB – User Statistics](#)
- [Percona Blog – PERFORMANCE\\_SCHEMA vs Slow Query Log](#)
- [Percona Blog – MySQL’s INNODB\\_METRICS table](#)
- [Percona Blog – Rotating MySQL Slow Logs Safely](#)
- [Percona Blog – Impact of logging on MySQL’s performance](#)
- [Percona Blog – Running Custom MySQL Queries in Percona Monitoring and Management](#)

---

Last update: 2021-12-14

### 2.3.3 MongoDB

How to set up PMM to monitor a MongoDB or Percona Server for MongoDB database instance.

#### Summary

- Create PMM account and set permissions.
- Configure profiling.
- Add service.
- Check service.

#### Before you start

Check that:

- PMM Server is installed and running with a known IP address or hostname accessible from the client node.
- PMM Client is installed and the node is registered with PMM Server.
- You have superuser (root) access on the client host.
- You have `adminUserAnyDatabase` or superuser role privilege to any database servers that you want to monitor.
- Your MongoDB server is version 4.0 or higher.

#### Create PMM account and set permissions

We recommend using a dedicated account to connect PMM Client to the monitored database instance.

This example creates a new custom role with the privileges needed by the Query Analyzer, and adds a database user with that role plus the built-in `clusterMonitor` role.

Values for username (`user`) and password (`pwd`) are examples. Replace them before using this code.

Run this in a `mongo` session.

```
db.getSiblingDB("admin").createRole({
  role: "explainRole",
  privileges: [
    {
      resource: {
        db: "",
        collection: "system.profile"
      },
      actions: [
        "listIndexes",
        "listCollections",
        "dbStats",
        "dbHash",
        "collStats",
        "find"
      ]
    }
  ],
  roles:[]
})

db.getSiblingDB("admin").createUser({
  user: "pmm_mongodb",
  ...
})
```

```

    pwd: "password",
    roles: [
      { role: "explainRole", db: "admin" },
      { role: "clusterMonitor", db: "admin" },
      { role: "read", db: "local" }
    ]
  )
}

```

## Profiling

To use PMM Query Analytics, you must turn on MongoDB's [profiling feature](#).

You can set profiling:

- permanently, by editing the MongoDB configuration file and restarting the database instance (recommended);
- when starting MongoDB, by passing arguments to `mongod` on the command line;
- until the next database instance restart, by running a command in a `mongo` session.

Profiling is turned off by default as it can adversely affect the performance of the database server.

### SET PROFILING IN THE CONFIGURATION FILE

1. Edit the configuration file (usually `/etc/mongod.conf`).
2. Create or add this to the `operationProfiling` section. ([Read more](#).)

```

operationProfiling:
  mode: all
  slowOpThresholdMs: 200
  rateLimit: 100 # (Only available with Percona Server for MongoDB.)

```

#### Important

This is a [YAML](#) file. Indentation matters.

3. Restart the `mongod` service. (Example for `systemd`.)

```
systemctl restart mongod
```

### SET PROFILING ON THE COMMAND LINE

```
mongod --dbpath=DATABASEDIR --profile 2 --slowms 200 --rateLimit 100
```

- `--dbpath`: The path to database files (usually `/var/lib/mongo`).
- `--profile`: The MongoDB profiling level. A value of `2` tells the server to collect profiling data for *all* operations. To lower the load on the server, use a value of `1` to only record slow operations.
- `--slowms`: An operation is classified as *slow* if it runs for longer than this number of milliseconds.
- `--rateLimit`: (Only available with Percona Server for MongoDB.) The sample rate of profiled queries. A value of `100` means sample every 100<sup>th</sup> fast query. ([Read more](#).)

#### Caution

Smaller values improve accuracy but can adversely affect the performance of your server.

#### SET PROFILING IN A MONGO SESSION

In a `mongo` session, the profiler should be enabled **per** database. For example, to enable the profiler in the `testdb`, run this:

```
use testdb
db.setProfilingLevel(2, {slowms: 0})
```

If you have already [added a service](#), you should remove it and re-add it after changing the profiling level.

#### Add service

When you have configured your database server, you can add a MongoDB service with the user interface or on the command line.

##### WITH THE USER INTERFACE

1. Select `Configuration` →  `PMM Inventory` →  `Add Instance`.
2. Select `MongoDB - Add a remote instance`.
3. Enter or select values for the fields.
4. Click `Add service`.

The screenshot shows the 'Add Instance' screen for a 'remote MongoDB Instance'. The interface is dark-themed with light-colored input fields.

**Main details**

- Hostname**: Hostname
- Service name**: Service name (default: Hostname)
- Port**: 27017
- Username**: Username
- Password**: Password

**Labels**

- Environment**: Environment
- Region**: Region
- Availability Zone**: Availability Zone
- Replication set**: Replication set
- Cluster**: Cluster

**Custom labels**

"Custom labels Format: key1:value1 key2:value2"

**Additional options**

- Skip connection check
- Use TLS for database connections
- Skip TLS certificate and hostname validation
- Use QAN MongoDB Profiler

**Buttons**

- Add service** (blue button)
- Return to menu**

## ON THE COMMAND LINE

Use `pmm-admin` to add the database server as a service using one of these example commands.

When successful, PMM Client will print `MongoDB Service added` with the service's ID and name. Use the `--environment` and `-custom-labels` options to set tags for the service to help identify them.

### Tips

- When adding nodes of a sharded cluster, add each node separately using the `--cluster mycluster` option for the MongoDB Cluster Summary dashboard to populate correctly.
- Atlas doesn't support direct connections. When connecting to an Atlas instance, use the `pmm-admin` option `--direct-connection=false`. (Doing so will prevent replicaset status from working and the MongoDB Overview dashboard widget will show invalid values.)

#### EXAMPLES

```
pmm-admin add mongodb \
--username=pmm_mongodb --password=password \
--query-source=profiler --cluster=mycluster
```

```
pmm-admin add mongodb \
--username=pmm_mongodb --password=password \
mongo 127.0.0.1:27017
```

```
pmm-admin add mongodb \
--username=pmm_mongodb --password=password \
--service-name=mymongosvc --host=127.0.0.1 --port=27017
```

#### Connect via UNIX socket

```
pmm-admin add mongodb --socket=/tmp/mongodb-27017.sock
```

#### Connecting via SSL/TLS

```
pmm-admin add mongodb --tls \
--tls-certificate-key-file=PATHTOCERT \
--tls-certificate-key-file-password=IFPASSWORDTOCERTISSET \
--tls-ca-file=PATHTOCACERT \
--authentication-mechanism=AUTHENTICATION-MECHANISM \
--authentication-database=AUTHENTICATION-DATABASE
```

where:

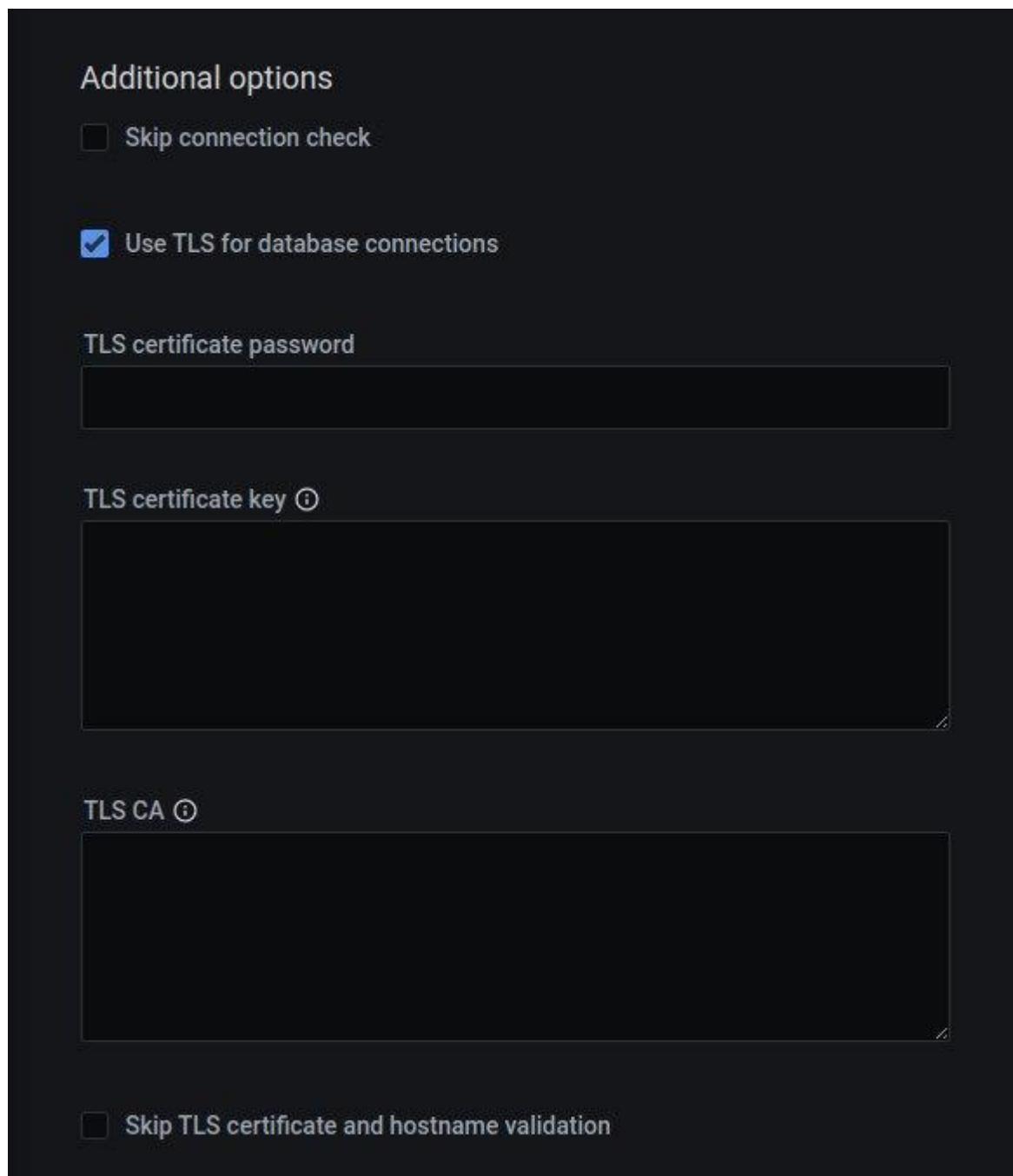
- `PATHTOCERT` : Path to TLS certificate file.
- `IFPASSWORDTOCERTISSET` : Password for TLS certificate file.
- `PATHTOCACERT` : Path to certificate authority file.
- `AUTHENTICATION-MECHANISM` : Authentication mechanism. Default is empty. Use `MONGODB-X509` for SSL certificates.
- `AUTHENTICATION-DATABASE` : Authentication database. Default is empty. Use `$external` for SSL certificates.

#### Check the service

##### WITH THE USER INTERFACE

1. Select `Configuration` →  `PMM Inventory` →  `Inventory list`.
2. Look in the `Services` tab for a matching `Service Type` (MongoDB), `Service name`, `Addresses`, and any other values used when adding the service.

3. Look in the *Agents* tab to check the desired data source is being used.
4. If your MongoDB instance is configured to use TLS, click on the **Use TLS for database connection** check box and fill in TLS certificates and keys.
  - a. If you use TLS, the authentication mechanism is automatically set to `MONGODB-X509`.



#### ON THE COMMAND LINE

Look for your service in the output of this command.

```
pmm-admin inventory list services --service-type=mongodb
```

#### CHECK DATA

1. Open the *MongoDB Instances Overview* dashboard.

2. Set the *Service Name* to the newly-added service.

#### Query Analytics

1. Open *PMM Query Analytics*.

2. In the *Filters* panel:

a. Under *Service Name*, select your service.

b. Under *Service Type* select `mongodb`.

#### Remove service

##### WITH THE USER INTERFACE

1. Select *Configuration* → *PMM Inventory* → *Inventory List*.

2. In the first column, click the tick box for the service you want to remove.

3. Click *Delete*.

4. On the *Confirm action* dialog window:

a. (Optional) Select *Force mode* to also delete associated agents.

b. Click *Proceed*.

##### ON THE COMMAND LINE

```
pmm-admin remove mongodb SERVICE_NAME
```

- `SERVICE_NAME` : The name the service was added as. (Find it with `pmm-admin list`.)

#### See also

- [pmm-admin add mongodb](#)
- [Troubleshooting connection difficulties](#)

Last update: 2022-01-20

## 2.3.4 PostgreSQL

How to set up PMM to monitor a [PostgreSQL](#) or [Percona Distribution for PostgreSQL](#) database instance.

### Summary

- Create PMM account and set permissions.
- Choose, install and configure an extension:
  - `pg_stat_statements`, or,
  - `pg_stat_monitor`.
- Add service.
- Check service.

### Before you start

Check that:

- [PMM Server is installed](#) and running with a known IP address accessible from the client node.
- [PMM Client is installed](#) and the [node is registered with PMM Server](#).
- You have superuser (root) access on the client host.
- You have superuser access to any database servers that you want to monitor.

(PMM follows PostgreSQL's end-of-life policy. For specific details on supported platforms and versions, see [Percona's Software Platform Lifecycle page](#).)

## Create a database account for PMM

We recommend creating a PMM database account that can connect to the `postgres` database with the `SUPERUSER` role.

1. Create a user. This example uses `pmm`. (Replace `*****` with a strong password of your choice.)

```
CREATE USER pmm WITH SUPERUSER ENCRYPTED PASSWORD '*****';
```

If your database runs on Amazon RDS:

```
CREATE USER pmm WITH rds_superuser ENCRYPTED PASSWORD '*****';
```

Optionally, you can also set up a connection limit (only if the user is not a SUPERUSER):

```
ALTER USER pmm CONNECTION LIMIT 10;
```

2. PMM must be able to log in locally as this user to the PostgreSQL instance. To enable this, edit the `pg_hba.conf` file.

If not already enabled by an existing rule, add:

| local  | all      | pmm  | md5     |        |
|--------|----------|------|---------|--------|
| # TYPE | DATABASE | USER | ADDRESS | METHOD |

(Ignore the second line. It is a comment to show field alignment.)

3. Reload the configuration:

```
su - postgres
psql -c "select pg_reload_conf()"
```

4. Check local login.

```
psql postgres pmm -c "\conninfo"
```

5. Enter the password for the `pmm` user when prompted.

## Choose and configure an extension

Decide which database extension to use, and configure your database server for it. The choices are:

1. `pg_stat_statements`, the original extension created by PostgreSQL, part of the `postgresql-contrib` package available on Linux.
2. `pg_stat_monitor` is a new extension created by Percona. It is based on and compatible with `pg_stat_statements`. `pg_stat_monitor` has all the features of `pg_stat_statements`, but adds *bucket-based data aggregation*.

We recommend choosing only one of these. **If you use both, you will get duplicate metrics.**

### Caution

While we recommend use of the newer `pg_stat_monitor` extension, be aware it is currently in beta phase and unsupported.

Here are the benefits and drawbacks of each.

|                                 | Benefits                                                                            | Drawbacks                                                           |
|---------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| <code>pg_stat_statements</code> | 1. Part of official <code>postgresql-contrib</code> package.                        | 1. No aggregated statistics or histograms.<br>2. No Query Examples. |
| <code>pg_stat_monitor</code>    | 1. Builds on <code>pg_stat_monitor</code> features.<br>2. Bucket-based aggregation. | 1. Beta software.                                                   |



### Bucket-based data aggregation

`pg_stat_monitor` collects statistics and aggregates data in a data collection unit called a *bucket*. These are linked together to form a *bucket chain*.

You can specify:

- the number of buckets (the length of the chain);
- how much space is available for all buckets;
- a time limit for each bucket's data collection (the *bucket expiry*).

When a bucket's expiration time is reached, accumulated statistics are reset and data is stored in the next available bucket in the chain.

When all buckets in the chain have been used, the first bucket is reused and its contents are overwritten.

If a bucket fills before its expiration time is reached, data is discarded.

## PG\_STAT\_STATEMENTS

### Install

- Debian/Ubuntu



#### Root permissions

```
apt install -y postgresql-contrib
```

- Red Hat/CentOS



#### Root permissions

```
yum install -y postgresql-contrib
```

### Configure

1. Add these lines to your `postgresql.conf` file:

```
shared_preload_libraries = 'pg_stat_statements'
track_activity_query_size = 2048 # Increase tracked query string size
pg_stat_statements.track = all    # Track all statements including nested
track_io_timing = on              # Capture read/write stats
```

2. Restart the database server. After the restart, the extension starts capturing statistics from every database.
3. Install the extension.

```
psql postgres postgres -c "CREATE EXTENSION pg_stat_statements SCHEMA public"
```

This command creates the view where you can access the collected statistics.

We recommend that you create the extension for the `postgres` database. In this case, you receive access to the statistics collected from every database.

You can now add the service.

#### PG\_STAT\_MONITOR

##### Caution

`pg_stat_monitor` is currently in beta phase and is unsupported.

`pg_stat_monitor` has been tested with:

- PostgreSQL versions 11, 12, 13.
- Percona Distribution for PostgreSQL versions 11, 12, 13.

#### Install

- If you use *Percona Distribution for PostgreSQL*, you can install the extension with your Linux package manager. See [Installing Percona Distribution for PostgreSQL](#).
- If you use *PostgreSQL* you can install by downloading and compiling the source code. See [Installing pg\\_stat\\_monitor](#).

#### Configure

1. Set or change the value for `shared_preload_library`.

In your `postgresql.conf` file:

```
shared_preload_libraries = 'pg_stat_monitor'
```

##### Caution

If you use both `pg_stat_statements` and `pg_stat_monitor`, set `pg_stat_monitor` **after** `pg_stat_statements`:

```
shared_preload_libraries = 'pg_stat_statements, pg_stat_monitor'
```

2. Set configuration values.

In your `postgresql.conf` file:

```
pg_stat_monitor.pgsm_query_max_len = 2048
```

### Caution

It is important to set maximal length of query to 2048 characters or more for PMM to work properly.

You can get a list of other available settings with `SELECT * FROM pg_stat_monitor_settings;`.

See [pg\\_stat\\_monitor GitHub repository](#) for details about available parameters.

3. Start or restart your PostgreSQL instance. The extension starts capturing statistics from every database.

4. In a `psql` session:

```
CREATE EXTENSION pg_stat_monitor;
```

This command creates the view where you can access the collected statistics.

We recommend that you create the extension for the `postgres` database. In this case, you receive the access to the statistics, collected from every database.

5. Check the version.

```
SELECT pg_stat_monitor_version();
```

### Add service

When you have configured your database server, you can add a PostgreSQL service with the user interface or on the command line.

#### WITH THE USER INTERFACE

1. Select `Configuration` →  `PMM Inventory` →  `Add Instance`.
2. Select `PostgreSQL – Add a remote instance`.
3. Enter or select values for the fields.
4. Click `Add service`.

**Add Instance**

[Return to menu](#)

**Main details**

**Hostname**

**Service name**  (default: Hostname)

**Port**  5432

**Username**

**Password**

**Labels**

**Environment**  Environment

**Region**  Region

**Availability Zone**  Availability Zone

**Replication set**  Replication set

**Cluster**  Cluster

**Custom labels**  
"Custom labels Format: key1:value1 key2:value2"

**Additional options**

Skip connection check

Use TLS for database connections

Skip TLS certificate and hostname validation

**Stat tracking options**

Don't track    PG Stat Statements    PG Stat Monitor

**Add service** [Return to menu](#)

If your PostgreSQL instance is configured to use TLS, click on the *Use TLS for database connections* check box and fill in your TLS certificates and key.

### Additional options

Skip connection check

Use TLS for database connections

TLS CA ⓘ

TLS certificate key ⓘ

TLS certificate ⓘ

Skip TLS certificate and hostname validation

#### Note

For TLS connection to work SSL needs to be configured in your PostgreSQL instance. Make sure SSL is enabled in the server configuration file `postgresql.conf`, and that hosts are allowed to connect in the client authentication configuration file `pg_hba.conf`. (See PostgreSQL documentation on [Secure TCP/IP Connections with SSL](#).)

#### ON THE COMMAND LINE

Add the database server as a service using one of these example commands. If successful, PMM Client will print `PostgreSQL Service added` with the service's ID and name. Use the `--environment` and `-custom-labels` options to set tags for the service to help identify them.

#### EXAMPLES

Add instance with default node (`<node>-postgresql`) and service name.

```
pmm-admin add postgresql \
--username=pmm \
--password=password \
--server-url=https://admin:admin@X.X.X.X:443 \
--server-insecure-tls
```

- `<user name>` : The PostgreSQL PMM user
- `<password>` : The PostgreSQL user credentials.

The service name and service ID will be automatically chosen.

Add instance with specified node and service name.

```
pmm-admin add postgresql \
--username=pmm \
--password=password \
--server-url=https://admin:admin@X.X.X.X:443 \
--server-insecure-tls \
```

Add instance to connect with a UNIX socket.

```
pmm-admin add postgresql --socket=/var/run/postgresql
```

#### Connecting via SSL/TLS

```
pmm-admin add postgresql --tls \
--tls-cert-file=PATHTOCERT \
--tls-ca-file=PATHTOCACERT \
--tls-key-file=PATHTOKEY \
--host=HOST \
--port=PORT \
--username=USER \
--service-name=SERVICE
```

where:

- `PATHTOCERT` : Path to client certificate file.
- `PATHTOCACERT` : Path to certificate authority file.
- `PATHTOKEY` : Path to client key file.
- `HOST` : Instance hostname or IP.
- `PORT` : PostgreSQL service port number.
- `USER` : Database user allowed to connect via TLS. Should match the common name (CN) used in the client certificate.
- `SERVICE` : Name to give to the service within PMM.

## Check the service

### CHECK SERVICE - PMM USER INTERFACE

1. Select *Configuration* →  *PMM Inventory* →  *Inventory list*.
2. Look in the *Services* tab for a matching *Service Type* (PostgreSQL), *Service name*, *Addresses*, and any other details entered in the form.
3. Look in the *Agents* tab to check the desired data source is being used.

### CHECK SERVICE - COMMAND LINE

Look for your service in the output of this command.

```
pmm-admin inventory list services
```

If using Docker, use `docker exec pmm-client pmm-admin inventory list services`

## CHECK DATA

1. Open the *PostgreSQL Instance Summary* dashboard.
2. Set the *Service Name* to the newly-added service.

### RUNNING CUSTOM QUERIES

The Postgres exporter can run custom queries to add new metrics not provided by default.

Those custom queries must be defined in the `/usr/local/percona/pmm2/collectors/custom-queries/postgresql` in the same host where the exporter is running. There are 3 directories inside it: - high-resolution/ - every 5 seconds - medium-resolution/ - every 10 seconds - low-resolution/ - every 60 seconds

Depending on the desired resolution for your custom queries, you can place a file with the queries definition. The file is a yaml where each query can have these fields:

```
query_name: query: the query definition master: boolean to specify if the query should be executed only in the master
metrics: - metric name: usage: GAUGE, LABEL, COUNTER, MAPPEDMETRIC or DURATION description: a human
readable description
```

#### Example

```
pg_postmaster_uptime: query: "select extract(epoch from current_timestamp - pg_postmaster_start_time()) as
seconds" master: true metrics: - seconds: usage: "GAUGE" description: "Service uptime"
```

Check the see also section for a more detailed description on MySQL custom queries with more examples about how to use custom queries in dashboards.

#### See also

- [pmm-admin man page for pmm-admin add postgresql](#)
- [Configuring Percona Repositories with percona-release](#)
- [Percona Blog – Running Custom MySQL Queries in Percona Monitoring and Management](#)

---

Last update: 2021-12-22

### 2.3.5 ProxySQL

Use the `proxysql` alias to enable ProxySQL performance metrics monitoring.

#### USAGE

```
pmm-admin add proxysql --username=pmm --password=pmm
```

where `username` and `password` are credentials for the administration interface of the monitored ProxySQL instance. You should configure a read-only account for monitoring using the `admin-stats_credentials` variable in ProxySQL

Additionally, two positional arguments can be appended to the command line flags: a service name to be used by PMM, and a service address. If not specified, they are substituted automatically as `<node>-proxysql` and `127.0.0.1:6032`.

The output of this command may look as follows:

```
pmm-admin add proxysql --username=pmm --password=pmm
```

```
ProxySQL Service added.  
Service ID : /service_id/f69df379-6584-4db5-a896-f35ae8c97573  
Service name: ubuntu-proxysql
```

Beside positional arguments shown above you can specify service name and service address with the following flags: `--service-name`, and `--host` (the hostname or IP address of the service) and `--port` (the port number of the service), or `--socket` (the UNIX socket path). If both flag and positional argument are present, flag gains higher priority. Here is the previous example modified to use these flags for both host/port or socket connections:

```
pmm-admin add proxysql --username=pmm --password=pmm --service-name=my-new-proxysql --host=127.0.0.1 --port=6032  
pmm-admin add proxysql --username=pmm --password=pmm --service-name=my-new-proxysql --socket=/tmp/proxysql_admin.sock
```

---

Last update: 2022-02-04

## 2.3.6 Amazon RDS

### Required settings

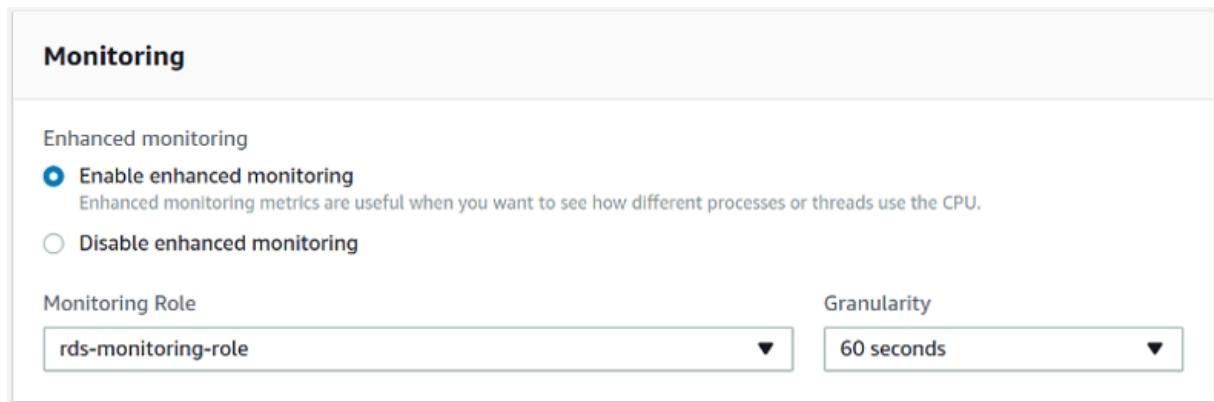
It is possible to use PMM for monitoring [Amazon RDS](#). In this case, the PMM Client is not installed on the host where the database server is deployed. By using the PMM web interface, you connect to the Amazon RDS DB instance. You only need to provide the IAM user access key (or assign an IAM role) and PMM discovers the Amazon RDS DB instances available for monitoring.

First of all, ensure that there is the minimal latency between PMM Server and the Amazon RDS instance.

Network connectivity can become an issue for VictoriaMetrics to scrape metrics with 1 second resolution. We strongly suggest that you run PMM Server on AWS (Amazon Web Services) in the same availability zone as Amazon RDS instances.

It is crucial that *enhanced monitoring* be enabled for the Amazon RDS DB instances you intend to monitor.

Set the *Enable Enhanced Monitoring* option in the settings of your Amazon RDS DB instance.



### Creating an IAM user with permission to access Amazon RDS DB instances

It is recommended that you use an IAM user account to access Amazon RDS DB instances instead of using your AWS account. This measure improves security as the permissions of an IAM user account can be limited so that this account only grants access to your Amazon RDS DB instances. On the other hand, you use your AWS account to access all AWS services.

The procedure for creating IAM user accounts is well described in the Amazon RDS documentation. This section only goes through the essential steps and points out the steps required for using Amazon RDS with Percona Monitoring and Management.

The first step is to define a policy which will hold all the necessary permissions. Then, you need to associate this policy with the IAM user or group. In this section, we will create a new user for this purpose.

### Creating a policy

A policy defines how AWS services can be accessed. Once defined it can be associated with an existing user or group.

To define a new policy use the IAM page at AWS.

Search IAM

**Dashboard**

- Groups
- Users
- Roles
- Policies
- Identity providers
- Account settings
- Credential report

---

Encryption keys

◀ Welcome to Identity and Access Management

IAM users sign-in link:  
<https://119175775298.signin.aws.amazon.com/console> 

| Customize

**IAM Resources**

|                              |                       |
|------------------------------|-----------------------|
| Users: 13                    | Roles: 16             |
| Groups: 2                    | Identity Providers: 1 |
| Customer Managed Policies: 2 |                       |

**Security Status**  2 out of 4 complete.

|                                                                                                                     |                                                                                     |
|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|  Activate MFA on your root account |  |
|  Create individual IAM users       |  |
|  Use groups to assign permissions  |  |
|  Apply an IAM password policy      |  |

1. Select the *Policies* option on the navigation panel and click the *Create policy* button.
2. On the *Create policy* page, select the JSON tab and replace the existing contents with the following JSON document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1508404837000",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch>ListMetrics",
        "Resource": ["*"]
      ],
      "Sid": "Stmt1508410723001",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeLogStreams",
        "logs:GetLogEvents",
        "logs:FilterLogEvents"
      ],
      "Resource": [ "arn:aws:logs:*log-group:RDSOSMetrics:*" ]
    }
  ]
}
```

3. Click *Review policy* and set a name to your policy, such as `AmazonRDSforPMMPolicy`. Then, click the *Create policy* button.

[Create policy](#)

---

[Review policy](#)

| Name*                                                              | AmazonRDSforPMMPolicy                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                        |                   |  |  |         |              |          |                   |                                                              |  |  |  |            |                          |               |      |                 |                     |                                        |      |     |               |               |      |
|--------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|-------------------|--|--|---------|--------------|----------|-------------------|--------------------------------------------------------------|--|--|--|------------|--------------------------|---------------|------|-----------------|---------------------|----------------------------------------|------|-----|---------------|---------------|------|
| Use alphanumeric and '+_,@_-' characters. Maximum 128 characters.  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                        |                   |  |  |         |              |          |                   |                                                              |  |  |  |            |                          |               |      |                 |                     |                                        |      |     |               |               |      |
| Description                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                        |                   |  |  |         |              |          |                   |                                                              |  |  |  |            |                          |               |      |                 |                     |                                        |      |     |               |               |      |
| Maximum 1000 characters. Use alphanumeric and '+_,@_-' characters. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                        |                   |  |  |         |              |          |                   |                                                              |  |  |  |            |                          |               |      |                 |                     |                                        |      |     |               |               |      |
| Summary                                                            | <table border="1"> <thead> <tr> <th colspan="4">Q Filter</th> </tr> <tr> <th>Service</th> <th>Access level</th> <th>Resource</th> <th>Request condition</th> </tr> </thead> <tbody> <tr> <td colspan="4">Allow (3 of 133 services) <a href="#">Show remaining 130</a></td> </tr> <tr> <td>CloudWatch</td> <td>Full: List Limited: Read</td> <td>All resources</td> <td>None</td> </tr> <tr> <td>CloudWatch Logs</td> <td>Limited: List, Read</td> <td>arn:aws:logs:*log-group:RDSOSMetrics:*</td> <td>None</td> </tr> <tr> <td>RDS</td> <td>Limited: List</td> <td>All resources</td> <td>None</td> </tr> </tbody> </table> | Q Filter                               |                   |  |  | Service | Access level | Resource | Request condition | Allow (3 of 133 services) <a href="#">Show remaining 130</a> |  |  |  | CloudWatch | Full: List Limited: Read | All resources | None | CloudWatch Logs | Limited: List, Read | arn:aws:logs:*log-group:RDSOSMetrics:* | None | RDS | Limited: List | All resources | None |
| Q Filter                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                        |                   |  |  |         |              |          |                   |                                                              |  |  |  |            |                          |               |      |                 |                     |                                        |      |     |               |               |      |
| Service                                                            | Access level                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Resource                               | Request condition |  |  |         |              |          |                   |                                                              |  |  |  |            |                          |               |      |                 |                     |                                        |      |     |               |               |      |
| Allow (3 of 133 services) <a href="#">Show remaining 130</a>       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                        |                   |  |  |         |              |          |                   |                                                              |  |  |  |            |                          |               |      |                 |                     |                                        |      |     |               |               |      |
| CloudWatch                                                         | Full: List Limited: Read                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | All resources                          | None              |  |  |         |              |          |                   |                                                              |  |  |  |            |                          |               |      |                 |                     |                                        |      |     |               |               |      |
| CloudWatch Logs                                                    | Limited: List, Read                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | arn:aws:logs:*log-group:RDSOSMetrics:* | None              |  |  |         |              |          |                   |                                                              |  |  |  |            |                          |               |      |                 |                     |                                        |      |     |               |               |      |
| RDS                                                                | Limited: List                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | All resources                          | None              |  |  |         |              |          |                   |                                                              |  |  |  |            |                          |               |      |                 |                     |                                        |      |     |               |               |      |

\* Required

[Cancel](#) [Previous](#) [Create policy](#)

### Creating an IAM user

Policies are attached to existing IAM users or groups. To create a new IAM user, select *Users* on the Identity and Access Management page at AWS. Then click *Add user* and complete the following steps:

The screenshot shows the AWS IAM 'Users' page. On the left, there's a navigation pane with options like Dashboard, Groups, Users (which is selected and highlighted in orange), Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main area has tabs for 'Add user' (highlighted in blue) and 'Delete user'. Below these are buttons for 'Find users by username or access key' and 'User name' (with a dropdown arrow). A table header includes columns for 'User name' (with a checkbox), 'Groups', 'Access key age', 'Password age', 'Last activity', 'MFA', and 'Access k'. There are also three small icons in the top right corner.

1. On the *Add user* page, set the user name and select the *Programmatic access* option under *Select AWS access type*. Set a custom password and then proceed to permissions by clicking the *Permissions* button.
2. On the *Set permissions* page, add the new user to one or more groups if necessary. Then, click *Review*.
3. On the *Add user* page, click *Create user*.

#### Creating an access key for an IAM user

To discover an Amazon RDS DB instance in PMM, you either need to use the access key and secret access key of an existing IAM user or an IAM role. To create an access key for use with PMM, open the IAM console and click *Users* on the navigation pane. Then, select your IAM user.

To create the access key, open the *Security credentials* tab and click the *Create access key* button. The system automatically generates a new access key ID and a secret access key that you can provide on the *PMM Add Instance* dashboard to have your Amazon RDS DB instances discovered.

In case, the PMM Server and Amazon RDS DB instance were created by using the same AWS account, you do not need to create the access key ID and secret access key manually. PMM retrieves this information automatically and attempts to discover your Amazon RDS DB instances.

#### Attaching a policy to an IAM user

The last step before you are ready to create an Amazon RDS DB instance is to attach the policy with the required permissions to the IAM user.

First, make sure that the Identity and Access Management page is open and open *Users*. Then, locate and open the IAM user that you plan to use with Amazon RDS DB instances. Complete the following steps, to apply the policy:

1. On the *Permissions* tab, click the *Add permissions* button.
2. On the *Add permissions* page, click *Attach existing policies directly*.
3. Using the *Filter*, locate the policy with the required permissions (such as `AmazonRDSforPMMPolicy`).
4. Select a check-box next to the name of the policy and click *Review*.
5. The selected policy appears on the *Permissions summary* page. Click *Add permissions*.

The `AmazonRDSforPMMPolicy` is now added to your IAM user.

## Add permissions to pmm\_user

1 2

### Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

### Creating an IAM role

Instead of creating an IAM user you can create an IAM role for a service, to discover Amazon RDS DB instances automatically without the need for access and secret keys. (But this only works if you are running PMM through AWS.)

To create an IAM role open the IAM console and click *Roles* on the navigation pane.

1. Click the *Create role* button.
2. Select *AWS service* and select *EC2* for the use case.
3. Click the *Next: Permissions* button.
4. Find the policy created previously and select it.
5. Click the *Next: Tags* button.
6. (Optional) Add a metadata tag to the role.
7. Click the *Next: Review* button.
8. Fill the role name and description.
9. Click the *Create role* button

After the role is created EC2 instances running PMM will have permissions to discover RDS DB instances.

It's also possible to create an IAM role to delegate permissions to an IAM user or to add permissions to a user belonging to another AWS account. See the [official AWS documentation on creating IAM roles](#).

### Setting up the Amazon RDS DB Instance

Query Analytics requires Configuring Performance Schema as the query source, because the slow query log is stored on the AWS (Amazon Web Services) side, and QAN agent is not able to read it. Enable the `performance_schema` option under `Parameter Groups` in Amazon RDS.

#### **Important**

Enabling Performance Schema on T2 instances is not recommended because it can easily run the T2 instance out of memory.

When adding a monitoring instance for Amazon RDS, specify a unique name to distinguish it from the local instance. If you do not specify a name, it will use the client's host name.

Create the `pmm` user with the following privileges on the Amazon RDS instance that you want to monitor:

```
CREATE USER 'pmm'@'%' IDENTIFIED BY 'pass';
GRANT SELECT, PROCESS, REPLICATION CLIENT ON *.* TO 'pmm'@'%';
ALTER USER 'pmm'@'%' WITH MAX_USER_CONNECTIONS 10;
GRANT SELECT, UPDATE, DELETE, DROP ON performance_schema.* TO 'pmm'@'%';
```

#### Adding an Amazon RDS, Aurora or Remote Instance

The preferred method of adding an Amazon RDS database instance to PMM is via the [Configuration](#) →  [PMM](#) [Inventory](#) →  [Add Instance](#) menu option.

This method supports Amazon RDS database instances that use Amazon Aurora, MySQL, or MariaDB engines, as well as any remote PostgreSQL, ProxySQL, MySQL and MongoDB instances.

The following steps are needed to add an Amazon RDS database instance to PMM:

1. In the PMM web interface, go to **Configuration** → **PMM Inventory** → **Add Instance**.
2. Select *Amazon RDS – Add a remote instance*.

**Add Instance**

Amazon RDS access key

Amazon RDS secret key

No data

- [Where do I get the security credentials for my Amazon RDS DB instances](#)
- [How to configure IAM role to automatically discover Amazon RDS DB instances](#)

3. Enter the access key ID and the secret access key of your IAM user or leave these fields empty if an IAM role was created.
4. Click the *Discover* button for PMM to retrieve the available Amazon RDS instances.

| Region    | Availability Zone | Engine                        | Instance ID           | Address                                                        | Action                  |
|-----------|-------------------|-------------------------------|-----------------------|----------------------------------------------------------------|-------------------------|
| us-east-2 | us-east-2c        | MySQL 5.7.mysql_aurora.2.07.2 | myauroradb-instance-1 | myauroradb-instance-1.cnzi7ua01o3d.us-east-2.rds.amazonaws.com | <b>Start monitoring</b> |
| us-east-2 | us-east-2b        | MySQL 8.0.20                  | mydbinstance          | mydbinstance.cnzi7ua01o3d.us-east-2.rds.amazonaws.com          | <b>Start monitoring</b> |

5. For the instance that you would like to monitor, select the *Start monitoring* button.
6. You will see a new page with the number of fields. The list is divided into the following groups: *Main details*, *RDS database*, *Labels*, and *Additional options*. Some already known data, such as already entered *AWS access key*, are filled in automatically, and some fields are optional.

### Main details

**Hostname ⓘ**

**Service name ⓘ**

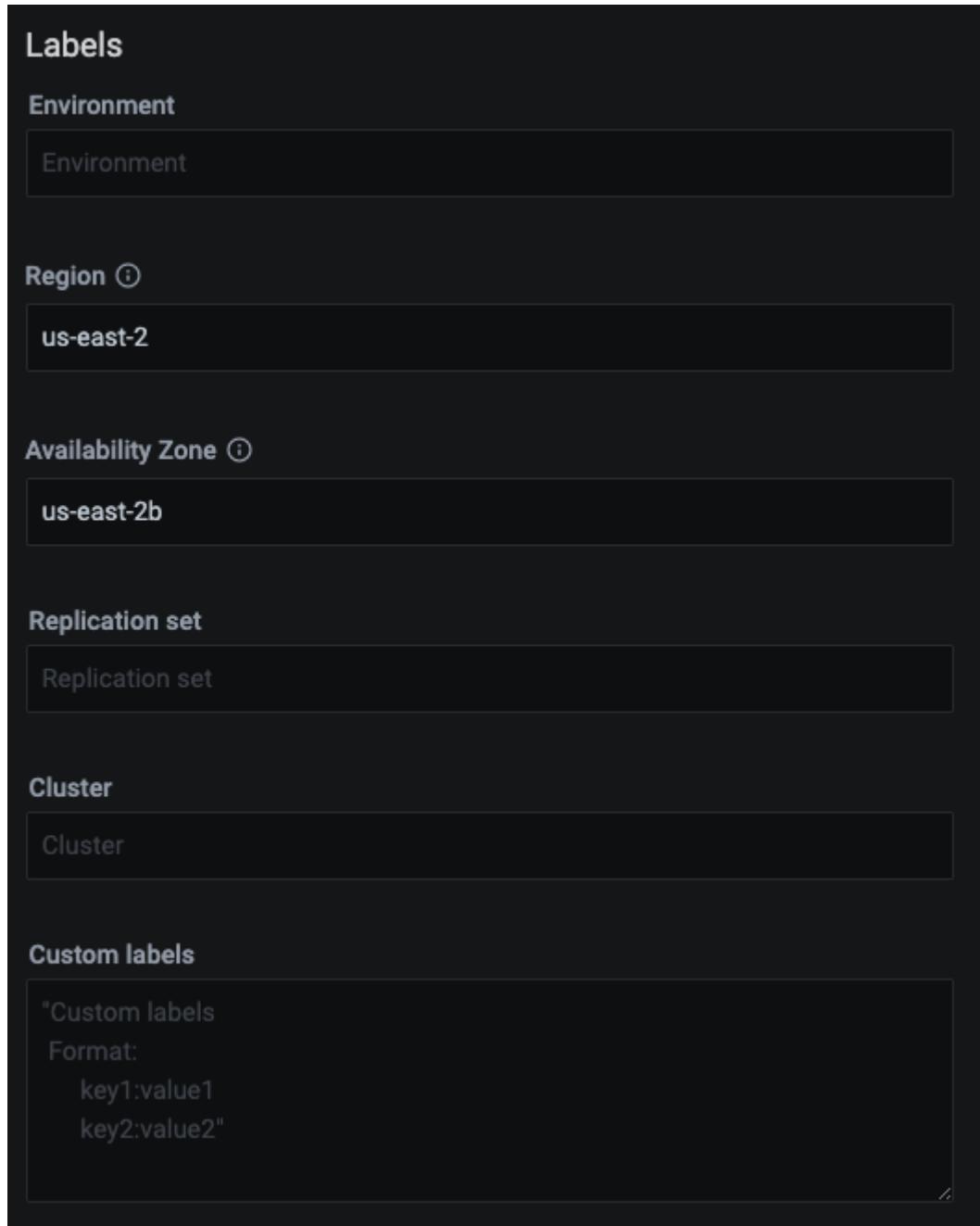
**Port ⓘ**

**Username ⓘ**

**Password ⓘ**



The *Main details* section allows you to specify the DNS hostname of your instance, the service name to use within PMM, the port your service is listening on, and the database user name and password.



The *Labels* section allows you to specify labels for the environment, the AWS region and availability zone to be used, the Replication set and Cluster names and also it allows you to set the list of custom labels in a key:value format.

**Additional options**

**Skip connection check**

**Use TLS for database connections**

**Skip TLS certificate and hostname validation**

**Table statistics limit**

|          |         |        |
|----------|---------|--------|
| Disabled | Default | Custom |
|----------|---------|--------|

1000

**Use performance schema**

**Disable Basic Metrics**

**Disable Enhanced Metrics**

**Add service**    **Return to menu**

The *Additional options* section contains specific flags which allow you to tune the RDS monitoring. They can allow you to skip connection check, to use TLS for the database connection, not to validate the TLS certificate and the hostname, as well as to disable basic and/or enhanced metrics collection for the RDS instance to reduce costs.

Also this section contains a database-specific flag, which would allow Query Analytics for the selected remote database:

- when adding some remote MySQL, AWS RDS MySQL or Aurora MySQL instance, you will be able to choose using performance schema for the database monitoring;
- when adding a PostgreSQL instance, you will be able to activate using `pg_stat_statements` extension;
- when adding a MongoDB instance, you will be able to choose using Query Analytics MongoDB profiler.

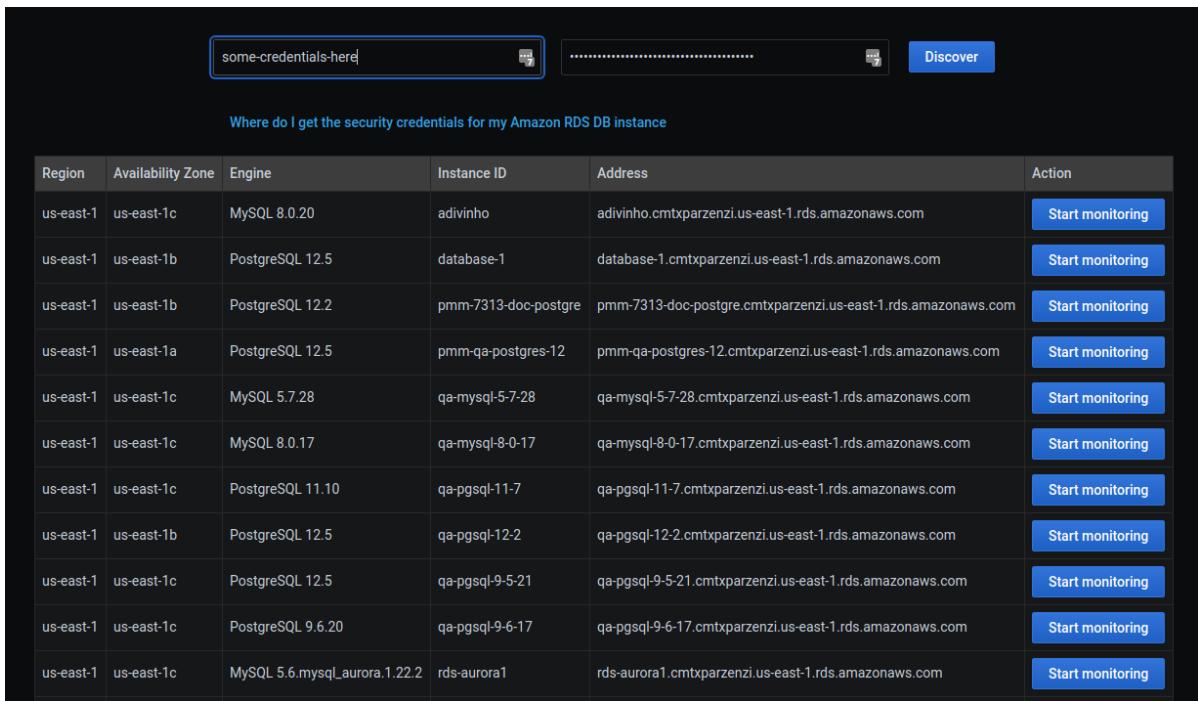
7. Finally press the *Add service* button to start monitoring your instance.

**Adding an Amazon RDS PostgreSQL instance**

For PostgreSQL, use the same method described above.

1. In the PMM web interface, go to [Configuration](#) →  [PMM Inventory](#) →  [Add Instance..](#)
2. Select *Amazon RDS – Add a remote instance.*

At the moment of writing this guide, the Add button doesn't mention PostgreSQL but the discovery function already supports it.



| Region    | Availability Zone | Engine                        | Instance ID          | Action                           |
|-----------|-------------------|-------------------------------|----------------------|----------------------------------|
| us-east-1 | us-east-1c        | MySQL 8.0.20                  | adivinho             | <a href="#">Start monitoring</a> |
| us-east-1 | us-east-1b        | PostgreSQL 12.5               | database-1           | <a href="#">Start monitoring</a> |
| us-east-1 | us-east-1b        | PostgreSQL 12.2               | pmm-7313-doc-postgre | <a href="#">Start monitoring</a> |
| us-east-1 | us-east-1a        | PostgreSQL 12.5               | pmm-qa-postgres-12   | <a href="#">Start monitoring</a> |
| us-east-1 | us-east-1c        | MySQL 5.7.28                  | qa-mysql-5-7-28      | <a href="#">Start monitoring</a> |
| us-east-1 | us-east-1c        | MySQL 8.0.17                  | qa-mysql-8-0-17      | <a href="#">Start monitoring</a> |
| us-east-1 | us-east-1c        | PostgreSQL 11.10              | qa-pgsql-11-7        | <a href="#">Start monitoring</a> |
| us-east-1 | us-east-1b        | PostgreSQL 12.5               | qa-pgsql-12-2        | <a href="#">Start monitoring</a> |
| us-east-1 | us-east-1c        | PostgreSQL 12.5               | qa-pgsql-9-5-21      | <a href="#">Start monitoring</a> |
| us-east-1 | us-east-1c        | PostgreSQL 9.6.20             | qa-pgsql-9-6-17      | <a href="#">Start monitoring</a> |
| us-east-1 | us-east-1c        | MySQL 5.6.mysql_aurora.1.22.2 | rds-aurora1          | <a href="#">Start monitoring</a> |

3. Follow steps 4 to 6 as in the previous section. Fill the form and remember to select `PG Stat Statement` to enable Query Analytics.

To get queries for Query Analytics, you need to enable `pg_stat_statements` in `postgres` database of your instance by running:

```
CREATE EXTENSION pg_stat_statements SCHEMA public;
```

### Add remote PostgreSQL Instance

Main details

Hostname ⓘ  
pmm-7313-doc-postgre.cmtxparzenzi.us-east-1.rds.amazonaws.com

Service name ⓘ  
pmm-7313-doc-postgre

Port ⓘ  
5432

Username ⓘ  
Username

Password ⓘ  
Password

Stat tracking options

---

Last update: 2021-12-20

## 2.3.7 Microsoft Azure

### **⚠ Caution**

Microsoft Azure functionality is currently in [technical preview](#) and is subject to change.

### Activate Microsoft Azure

The Microsoft Azure feature is turned off by default. To turn it on:

1. Go to *Configuration* → *Settings* → *Advanced Settings*.
2. Click the  toggle in the *Technical preview features* section of the page.

### Required settings

It is possible to use PMM for monitoring [Azure](#) database instances like other remote instances. In this case, the PMM Client is not installed on the host where the database server is deployed. By using the PMM web interface, you connect to the Azure DB instance. Discovery is not yet implemented in PMM but it is possible to add known instances by providing the connection parameters.

First of all, ensure that there is the minimal latency between PMM Server and the Azure instance.

Second, add a firewall rule to enable access from PMM Client like this:

The screenshot shows the AWS Firewall rules configuration page. It includes sections for Firewall rules and SSL settings. In the Firewall rules section, a new rule is being created with the following details:

| Firewall rule name                 | Start IP      | End IP        |
|------------------------------------|---------------|---------------|
| ClientIPAddress_2021-2-15_16-37-38 | 186.122.18.18 | 186.122.18.18 |
| Firewall rule name                 | Start IP      | End IP        |

In the SSL settings section, the 'Enforce SSL connection' toggle is set to 'ENABLED'.

### Setting up a MySQL instance

Query Analytics requires you to configure *Performance Schema* as the query source, because the slow query log is stored on the Azure side, and QAN agent is not able to read it. Enable the `performance_schema` option under **Parameter Groups** in Azure MySQL databases.

When adding a monitoring instance for Azure, specify a unique name to distinguish it from the local MySQL instance. If you do not specify a name, it will use the client's host name.

Create the `pmm` user with the following privileges on the Azure MySQL database instance that you want to monitor:

```
CREATE USER 'pmm'@'%' IDENTIFIED BY 'pass';
GRANT SELECT, PROCESS, REPLICATION CLIENT ON *.* TO 'pmm'@'%';
ALTER USER 'pmm'@'%' WITH MAX_USER_CONNECTIONS 10;
```

## Adding an Azure Instance

Follow the instructions for remote instances explained [here](#), Azure MySQL databases are similar to AWS RDS databases.

Example:

Add remote MySQL Instance

Main details

Hostname ⓘ

pmm-7582-mysql57.mysql.database.azure.com

Service name ⓘ

pmm-7582-mysql

Port ⓘ

3306

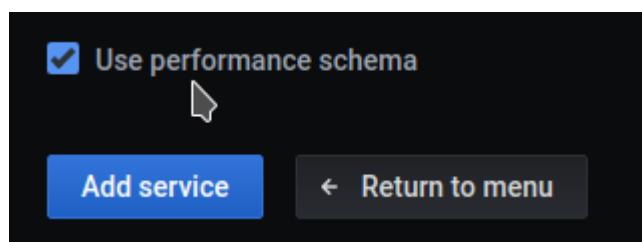
Username ⓘ

msandbox@pmm-7582-mysql57

Password ⓘ

.....

and be sure to set *Performance Schema* as the query collection method for Query Analytics.



## MARIADB

MariaDB up to version 10.2 works out of the box but starting with MariaDB 10.3 instrumentation is disabled by default and cannot be enabled since there is no SUPER role in Azure-MariaDB. So, it is not possible to run the required queries to enable instrumentation. Monitoring will work but Query Analytics won't receive any query data.

## POSTGRESQL

For PostgreSQL follow the same methods used for MySQL and MariaDB and enable `track_io_timing` in the instance configuration to enable Query Analytics.

The screenshot shows the 'Server parameters' section for a PostgreSQL server named 'azure-pg-10'. The 'trac' section is selected. The 'pg\_stat\_statements.track' parameter is set to 'all'. Other parameters shown include 'pg\_qs.query\_capture\_mode', 'pg\_qs.track\_utility', 'pg\_stat\_statements.max', 'pg\_stat\_statements.track', 'pgms\_wait\_sampling.query\_capture\_mode', 'track\_activities', 'track\_activity\_query\_size', 'track\_commit\_timestamp', 'track\_counts', 'track\_functions', and 'track\_io\_timing'. Each parameter has its value, parameter type (Dynamic or Static), and a brief description.

For Query Analytics, set the server parameter:

```
pg_stat_statements.track = all
```

### To discover databases on Azure

You need to get the Client ID, Client Secret, Tenant ID and Subscription ID.

#### GET THE SUBSCRIPTION ID

1. Search **Subscriptions**, click on your susbcription name

The screenshot shows the 'Access control (IAM)' blade for a specific subscription. It lists roles such as 'Add role assignment (disabled)', 'Add co-administrator (disabled)', 'Add custom role (disabled)', and 'View my access'. A 'Grant access to this resource' panel is open, containing fields for 'Add role assignment', 'Use the classic experience', and a 'Learn more' link.

2. Copy the susbcription ID

The screenshot shows the 'Subscription ID' field, which is currently empty. This is where the copied subscription ID would be pasted.

CREATE A NEW APPLICATION TO GET THE TENANT ID, CLIENT ID AND THE CLIENT SECRET.

1. Search for **Azure Active Directory**

Search resources, services, and docs (G+/)

Search history

- subs
- azure
- azure ac
- maria
- mysql

Recent services

- Azure Active Directory
- Subscriptions
- Virtual machines
- Azure Database for..
- Azure Database for..
- Azure Database for..
- SQL databases

Recent resources See more

2. Register a new application

The screenshot shows the Azure Active Directory Overview page. The left sidebar has sections like 'Overview', 'Preview features', 'Diagnose and solve problems', and 'Manage'. Under 'Manage', there are links for 'Users', 'Groups', 'External Identities', 'Roles and administrators', 'Administrative units', 'Enterprise applications', 'Devices', and 'App registrations'. The 'App registrations' link is highlighted with a red box. At the bottom of the sidebar, there's a 'New registration' button. The main content area has tabs for 'Endpoints', 'Troubleshooting', 'Refresh', 'Download', 'Preview features', and 'Got feedback?'.

## Register an application

### \* Name

The user-facing display name for this application (this can be changed later).

pmm-8108-sample-app



### Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (Percona only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[Help me choose...](#)

### Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web

e.g. <https://example.com/auth>

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the Microsoft Platform Policies [↗](#)

[Register](#)

3. At this point you can copy the client and tenant IDs.

| Essentials              |                        |
|-------------------------|------------------------|
| Display name            | : pmm-8108-sample-app  |
| Application (client) ID | : ab[REDACTED]ae       |
| Object ID               | : 4d[REDACTED]00       |
| Directory (tenant) ID   | : 46[REDACTED]d4       |
| Supported account types | : My organization only |

4. Create an application secret.

- |                             |                                               |
|-----------------------------|-----------------------------------------------|
| Client credentials          | : <a href="#">Add a certificate or secret</a> |
| Redirect URIs               | : <a href="#">Add a Redirect URI</a>          |
| Application ID URI          | : <a href="#">Add an Application ID URI</a>   |
| Managed application in I... | : pmm-8108-sample-app                         |

[Add a certificate or secret](#)

**Application registration certificates, secrets and federated credentials can be found in the tabs below.**

Certificates (0) Client secrets (0) Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

| Description       | Expires | Value | Secret ID |
|-------------------|---------|-------|-----------|
| New client secret |         |       |           |

No client secrets have been created for this application.

## Add a client secret

Description: pmm-8108-sample-secret

Expires: Recommended: 6 months

Add Cancel

5. Copy the **value** of the application secret. Once you leave this page you won't be able to see the secret again and you will have to generate a new one.

| New client secret      |           |                   |                                      |
|------------------------|-----------|-------------------|--------------------------------------|
| Description            | Expires   | Value             | Secret ID                            |
| pmm-8108-secret        | 5/16/2022 | vIA*****          | 548cbfc3-66af-4f5d-bc87-d681eeb9419b |
| pmm-8108-sample-secret | 5/16/2022 | TQU7Q  [REDACTED] | 15962343-c30f-4df6-9f5a-598cd661425d |

6. Give API access permissions to your application.

6.1. Search for **Subscriptions** like in step 1.

6.2. Select your application and grant **Monitor Reader** permissions. This might require you have admin permissions in your Azure account.

The screenshot shows the Microsoft Azure portal interface for managing access control (IAM). The left sidebar shows 'Subscriptions > Pay-As-You-Go'. The main area is titled 'Pay-As-You-Go | Access control (IAM)' and shows a list of roles. A red box highlights the 'Access control (IAM)' section in the sidebar. A mouse cursor is hovering over the 'Add role assignment' button in the top right of the main panel. The right side has a sidebar with 'Grant access to' and 'Add role assignment' buttons.

| Name                          | Description                                                                                       | Type        | Category  | Details              |
|-------------------------------|---------------------------------------------------------------------------------------------------|-------------|-----------|----------------------|
| Data Lake Analytics Developer | Lets you submit, monitor, and manage your own jobs but not create or delete Data Lake Analy...    | BuiltInRole | Storage   | <a href="#">View</a> |
| Log Analytics Contributor     | Log Analytics Contributor can read all monitoring data and edit monitoring settings. Editing m... | BuiltInRole | Analytics | <a href="#">View</a> |
| Log Analytics Reader          | Log Analytics Reader can view and search all monitoring data as well as and view monitoring s...  | BuiltInRole | Analytics | <a href="#">View</a> |
| Monitoring Contributor        | Can read all monitoring data and update monitoring settings.                                      | BuiltInRole | Monitor   | <a href="#">View</a> |
| Monitoring Metrics Publisher  | Enables publishing metrics against Azure resources                                                | BuiltInRole | Monitor   | <a href="#">View</a> |
| Monitoring Reader             | Can read all monitoring data.                                                                     | BuiltInRole | Monitor   | <a href="#">View</a> |

Buttons at the bottom include 'Review + assign', 'Previous', and 'Next'.

**Select members**

Select ⓘ  
pmm

| Name       | Object ID  | Type |
|------------|------------|------|
| PM         | [REDACTED] |      |
| PM         | [REDACTED] |      |
| [REDACTED] | [REDACTED] |      |
| [REDACTED] | [REDACTED] |      |

Selected members:  
pmm-8108-sample-app Remove

Review + assign    Previous    Next    Select    Close

**Review + assign**

Role    Members    Review + assign

**Selected role** Monitoring Reader

**Assign access to**  User, group, or service principal  Managed identity

**Members** [+ Select members](#)

| Name                | Object ID       | Type |
|---------------------|-----------------|------|
| pmm-8108-sample-app | 0 [REDACTED] .. | App  |

**Description** Optional

Review + assign    Previous    Next    Close

When you fill in all fields press the *Discover* button and you will see a list of available databases for monitoring.

The screenshot shows the 'Add Instance' screen for Microsoft Azure. It includes fields for Client ID, Client Secret, Tenant ID, and Subscription ID. Below these fields is a 'Discover' button, which is highlighted with a red arrow. A tooltip above the button reads: 'Where do I get the security credentials for my Azure DB instance?'. The main area displays a table of discovered databases:

| Region | Resource group | Name               | Engine     | Address                                | Action                            |
|--------|----------------|--------------------|------------|----------------------------------------|-----------------------------------|
| eastus | [REDACTED]     | [REDACTED]-sql-db3 | MySQL      | [REDACTED].mysql.database.azure.com    | <button>Start monitoring</button> |
| eastus | [REDACTED]     | [REDACTED]-sql-db1 | PostgreSQL | [REDACTED].postgres.database.azure.com | <button>Start monitoring</button> |
| eastus | [REDACTED]     | [REDACTED]         | MySQL      | [REDACTED].mariadb.database.azure.com  | <button>Start monitoring</button> |
| eastus | [REDACTED]     | [REDACTED]         | PostgreSQL | [REDACTED].postgres.database.azure.com | <button>Start monitoring</button> |
| eastus | [REDACTED]     | [REDACTED]         | MySQL      | [REDACTED].mysql.database.azure.com    | <button>Start monitoring</button> |

You can monitor 6 types of databases:

- Microsoft.DBforMySQL/servers
- Microsoft.DBforMySQL/flexibleServers
- Microsoft.DBforMariaDB/servers
- Microsoft.DBforPostgreSQL/servers
- Microsoft.DBforPostgreSQL/flexibleServers
- Microsoft.DBforPostgreSQL/serversv2

You can find more details on how to create DB on Azure at:

- <https://docs.microsoft.com/en-us/azure/postgresql/>
- <https://docs.microsoft.com/en-us/azure/mysql/>

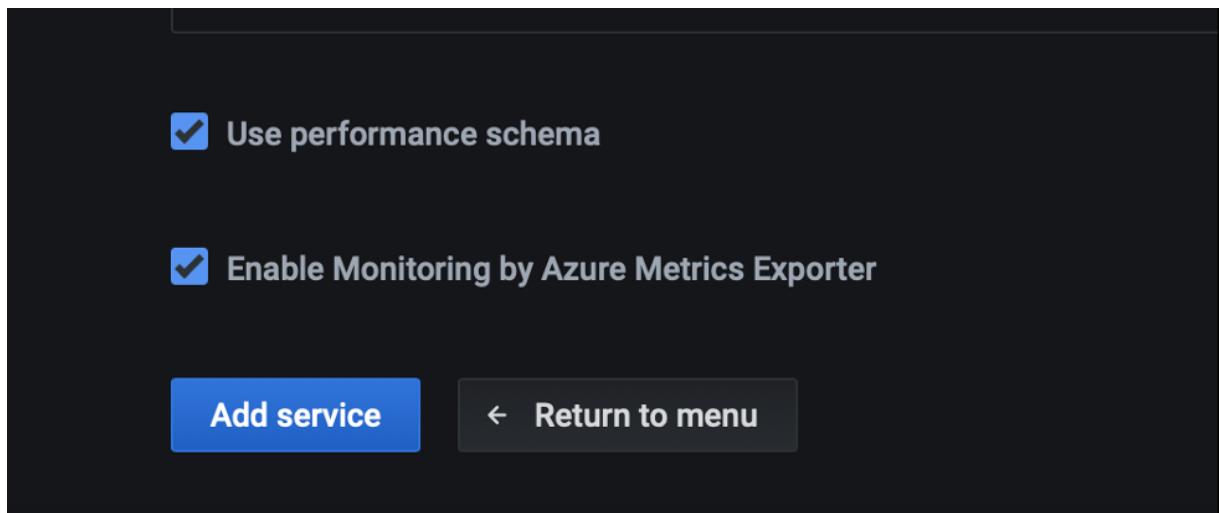
### Tip

You must set `pg_stat_statements.track = all` in your PostgreSQL Server settings to use PMM Query Analytics. ([Read more.](#))



In the list of databases on the Discovery page click *Start Monitoring* to add the selected Azure Database to PMM.

Fill in all required fields and click *Add service*.



PMM can use 3 exporters to collect metrics:

- Azure Metrics Exporter – collect “system” metrics related to DB.
  - `node_cpu_average`
  - `azure_resource_info`
  - `node_filesystem_size_bytes`
  - `azure_memory_percent_average`
  - `azure_storage_percent_average`
  - `azure_storage_used_bytes_average`
  - `node_network_receive_bytes_total`
  - `node_network_transmit_bytes_total`
- `mysql_exporter` or `postgres_exporter` – to collect database related metrics.
- PMM Agent to collect queries related metrics using `pg_stat_statements` for PostgreSQL or Performance Schema for MySQL (MariaDB)

#### ADDING AN AZURE INSTANCE ON PMM-CLIENT SIDE

TLS/SSL is enforced on the server by default. So please download the certificate needed to communicate over SSL with your Azure Database. It can be done on Networking tab for your Azure Database instance.

**azuremysql | Networking**

Azure Database for MySQL flexible server

Search (Ctrl+ /) Save Discard Download SSL Certificate Feedback

Overview Activity log Access control (IAM) Tags

**Settings**

Compute + storage Networking Connection strings Server parameters

**Enforced TLS/SSL connection**

TLS/SSL is enforced on the server by default. You can download the SSL public certificate parameter to OFF. You can also change the TLS version by updating the tls\_version server parameter.

**Network connectivity**

You can connect to your server by specifying a public IP address specified below or

Connectivity method  Public access (allowed IP addresses)  Private access (VNet Integration)

**Note:** Connections from the IP addresses below will have access to this server. B

Also enforced TLS/SSL connection option can be disabled on server side.

Command for adding an azure database service for monitoring without TLS/SSL.

```
pmm-admin add mysql --username=azureuser --password=secure --host=azuremysql.mysql.database.azure.com --service-name=azure1 --query-source=perfschema
```

Downloaded certificate is named `DigiCertGlobalRootCA.crt.pem`.

An example of the command for adding an Azure database service for monitoring with TLS/SSL would be:

```
pmm-admin add mysql --username=azureuser --password=secure --host=azuremysql.mysql.database.azure.com --service-name=azure1 --query-source=perfschema --tls --tls-ca=DigiCertGlobalRootCA.crt.pem --tls-cert=client-cert.pem --tls-key=client-key.pem --tls-skip-verify
```

Last update: 2021-11-24

### 2.3.8 Google Cloud Platform

PMM can monitor MySQL or PostgreSQL instances hosted on the [Google Cloud Platform](#).

The connection can be direct, or indirect using [Cloud SQL Proxy](#).

#### MySQL

1. Set up a MySQL instance on Google Cloud.
2. The database server must be accessible by PMM Client. If PMM Client is not also hosted on GCP, you will need to add a network interface with a public interface.
3. Configure *Performance Schema* on the MySQL server. Using the GCP console's *Cloud Shell* or your own `gcloud` installation, run:

```
gcloud sql instances patch <instance_name> --database-flags performance_schema=on
```

4. Log into the PMM user interface.
5. Select *Configuration* →  *PMM Inventory* →  *Add Instance*.
6. Click *MySQL Add a remote instance*.
7. Fill in the details for the remote MySQL instance.
  - Ensure *Use performance schema* is selected.
8. Click *Add service*.
9. Check for values in the *MySQL Instance Overview* dashboard and in *Query Analytics*.

#### PostgreSQL

1. Set up a PostgreSQL instance on Google Cloud.
2. The database server must be accessible by PMM Client. If PMM Client is not also hosted on GCP, you will need to add a network interface with a public interface.
3. Configure `pg_stat_statements`. Open an interactive SQL session with your GCP PostgreSQL server and run:

```
CREATE EXTENSION pg_stat_statements;
```

4. Log into the PMM user interface.
5. Select *Configuration* →  *PMM Inventory* →  *Add Instance*.
6. Select *PostgreSQL Add a remote instance*.
7. Fill in the details for the remote PostgreSQL instance:
  - In *Stat tracking options*, select *PG Stat Statements*.
8. Click *Add service*.
9. Check for values in the *PostgreSQL Instance Overview* dashboard and *Query Analytics*.

#### Cloud SQL Proxy

##### MySQL

1. Create instance on GCP.
2. Note connection as `<project_id>:<zone>:<db_instance_name>`.

3. [Enable Admin API](#) and download the JSON credential file.
4. Enable *Performance Schema*.
5. Run Cloud SQL Proxy (runs on PMM Client node).

- As a Docker container:

```
docker run -d \
-v ~/path/to/admin-api-file.json:/config \
-p 127.0.0.1:3306:3306 \
gcr.io/cloudsql-docker/gce-proxy:1.19.1 \
/cloud_sql_proxy \
--instances=example-project-NNNN:us-central1:mysql-for-pmm=tcp:0.0.0.0:3306 \
--credential_file=/config
```

- On Linux:

```
wget https://dl.google.com/cloudsql/cloud_sql_proxy.linux.amd64 -O cloud_sql_proxy
chmod +x cloud_sql_proxy
./cloud_sql_proxy --instances=example-project-NNNN:us-central1:mysql-for-pmm=tcp:3306 \
--credential_file=/path/to/credential-file.json
```

6. Add instance.

```
pmm-admin add mysql --host=127.0.0.1 --port=3306 \
--username=root --password=secret \
--service-name=MySQLGCP --query-source=perfschema
```

#### POSTGRESQL

1. Create instance on GCP.
2. Note connection as `<project_id>:<zone>:<db_instance_name>`.
3. [Enable Admin API](#) and download the JSON credential file.
4. Run Cloud SQL Proxy.

```
./cloud_sql_proxy --instances=example-project-NNNN:us-central1:pg-for-pmm=tcp:5432 \
--credential_file=/path/to/credential-file.json
```

5. Log into PostgreSQL.

6. Load extension:

```
CREATE EXTENSION pg_stat_statements;
```

7. Add service:

```
pmm-admin add postgresql --host=127.0.0.1 --port=5432 \
--username="postgres" --password=secret --service-name=PGGCP
```

Last update: 2021-07-07

## 2.3.9 Linux

### Adding general system metrics service

PMM collects Linux metrics automatically starting from the moment when you have configured your node for monitoring with `pmm-admin config`.

---

Last update: 2020-12-10

### 2.3.10 External Services

#### Adding general external services

You can collect metrics from an external (custom) exporter on a node when:

- there is already a PMM Agent instance running and,
- this node has been [configured](#) using the `pmm-admin config` command.

#### USAGE

```
pmm-admin add external --service-name=<service-name> --listen-port=<listen-port> --metrics-path=<metrics-path> --scheme=<scheme>
```

```
pmm-admin add external-serverless --external-name=<external-service-name> --host=<hostname> --listen-port=<listen-port> --metrics-path=<metrics-path> --scheme=<scheme>
```

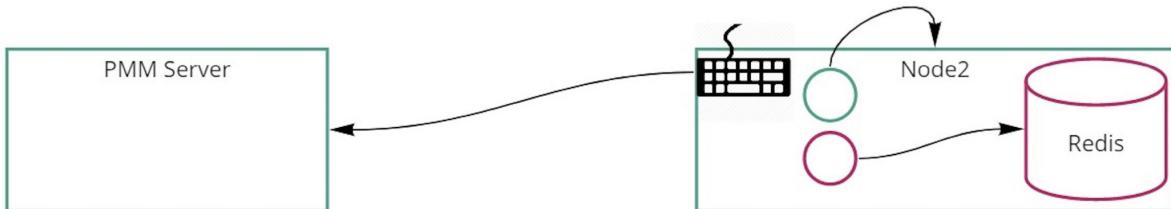
#### Getting data from external exporters

There two ways to get metrics from other exporters:

- `external` will collect metrics from the exporter that is run on the same host as PMM Client's connection to it by a port. (See more details with `pmm-admin add external --help`.)
- `external-serverless` is useful for collecting metrics from cloud services. You need a host and port number to add it to PMM Server. (See more details with `pmm-admin add external-serverless --help`.)

Here are the differences between `external` and `external-serverless` types.

Connection schema of external exporter:



Connection schema of external serverless exporter:

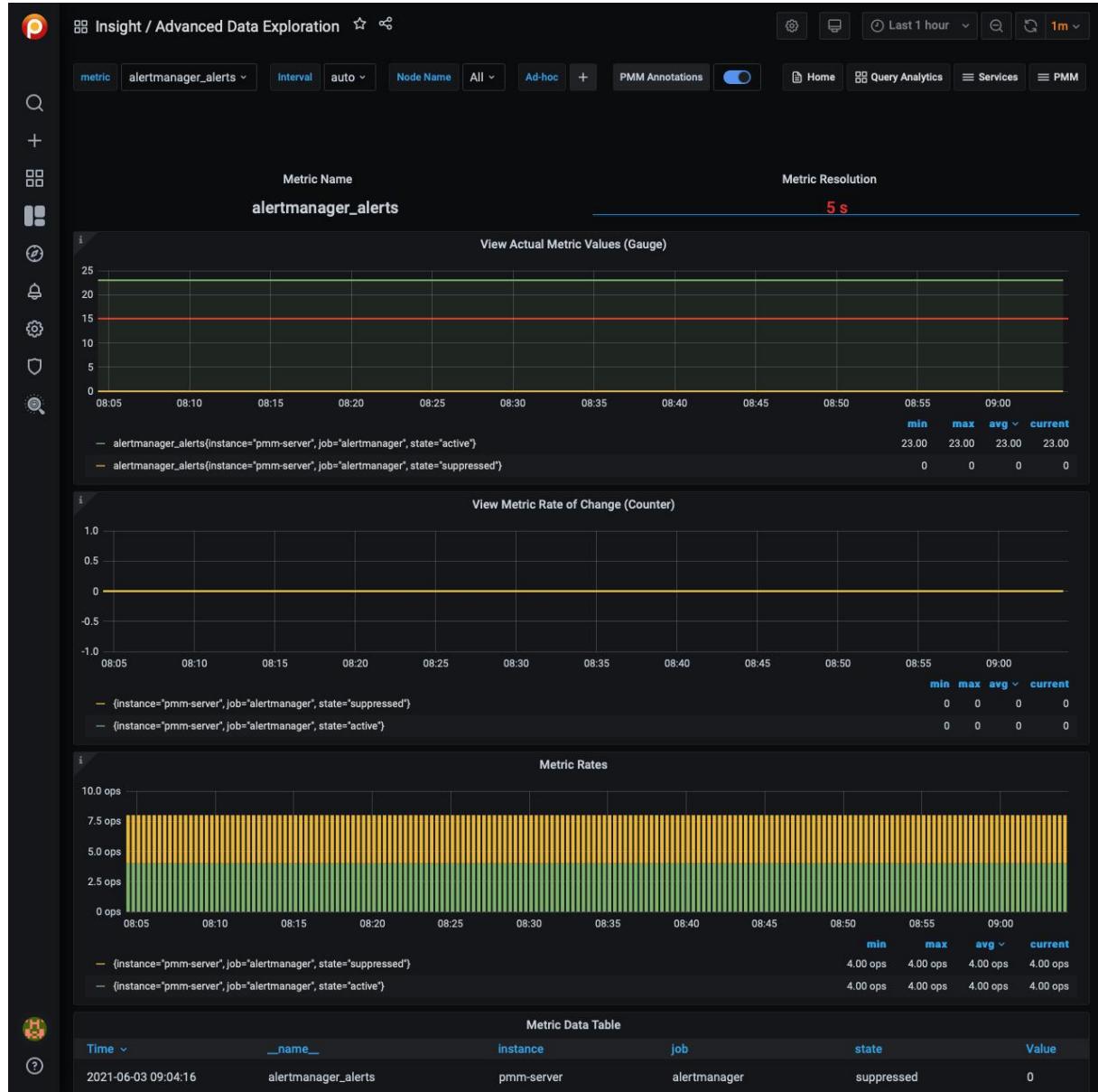


#### How I can add something not supported by PMM

PMM can collect any metrics in [Open metrics](#) or [Prometheus exposition](#) format. You must specify the host and port of these metrics using the `pmm-admin add external` or `pmm-admin add external-serverless` commands.

From this point, PMM will collect and store available metrics.

To browse and visualize collected metrics as a first step, we can look at the Advanced Data Exploration dashboard and select informative services and metrics.



Another way is to create a [new Grafana Dashboard](#) to PMM as needed.

One more way is to search for an already created dashboard at <https://grafana.com/grafana/dashboards> for the added exporter and import it into PMM.

#### THIRD-PARTY EXPORTERS

You can find more exporters on the [official Prometheus page](#).

#### CUSTOM EXPORTER

You can write a custom external exporter or extend your application to expose metrics in Prometheus format.

For more details see [https://prometheus.io/docs/instrumenting/writing\\_exporters/](https://prometheus.io/docs/instrumenting/writing_exporters/).

#### EXAMPLES

```
root@mysql1:~# pmm-admin add external --group=processes --listen-port=9256
External Service added.
Service ID : /service_id/6485f4fd-745b-4dfb-8b72-328e300f8b50
Service name: mysql1-processes
Group       : processes
```

- Add an exporter running on local port 9256 to the group called `processes`.
- Use the group and host names to automatically generate a service name.
- Use the default schema and metrics path.

#### Adding an External service via UI

1. In the PMM web interface, go to [Configuration](#) → [PMM Inventory](#) → [Add Instance](#).
2. Select *External Service - Add a remote instance*.

Add Instance

External service connection details

Service name (default: Hostname)

Group

External service connection parameters

Set manually Parse from URL string

Schema

HTTPS HTTP

External service hostname

Hostname

Metrics path

/path/to/metrics

External service port

443

3. Fill the form and set the external service endpoint.

The endpoint can be set manually:

**External service connection parameters ⓘ**

**Schema ⓘ**

**External service hostname ⓘ**

**Metrics path ⓘ**

**External service port ⓘ**

**Username ⓘ**

**Password ⓘ**

This screenshot shows the configuration interface for an external service connection. It includes sections for setting connection parameters manually or parsing from a URL string, choosing the schema (HTTPS or HTTP), specifying the external service's hostname and metrics path, defining the port number, and entering a username and password.

or by parsing required data from a URL string, in which case you only need to pass a valid URL.

**External service connection parameters ⓘ**

**External service endpoint ⓘ**

This screenshot shows the configuration interface for an external service connection, specifically using the 'Parse from URL string' method. It displays a single input field containing the URL 'http://example.com:3333/path/to/metrics'.

## 2.3.11 HAProxy

### Adding HAProxy services

You can collect metrics from [HAProxy](#) on a node when:

- There is already a configured haproxy instance.
- See [How to configure HAProxy](#).
- After HAProxy is running (default address <http://localhost:8404/metrics>) you can add it to PMM.
- Use the `haproxy` alias to enable HAProxy metrics monitoring.
- There is already a PMM Agent instance running.
- This node has been [configured](#) using the `pmm-admin config` command.

### USAGE

```
pmm-admin add haproxy --listen-port=8404
```

where `listen-port` is the port number where HAProxy running. (This is the only required flag.)

The output of this command should look as follows:

```
HAProxy Service added.
Service ID : /service_id/c481183f-70a2-443f-91e5-cae5cecd06a2
Service name: Ubuntu-haproxy
```

Additionally, one positional argument can be appended to the command line flags: a service name to be used by PMM. If not specified, they are substituted automatically as `<node>-haproxy`.

During adding here is connection check (can be skipped by flag `--skip-connection-check`). If HAProxy doesn't run properly on the given port then you will see an error message:

```
Connection check failed: Get "http://127.0.0.1:8404/metrics": dial tcp 127.0.0.1:8404:
connect: connection refused.
```

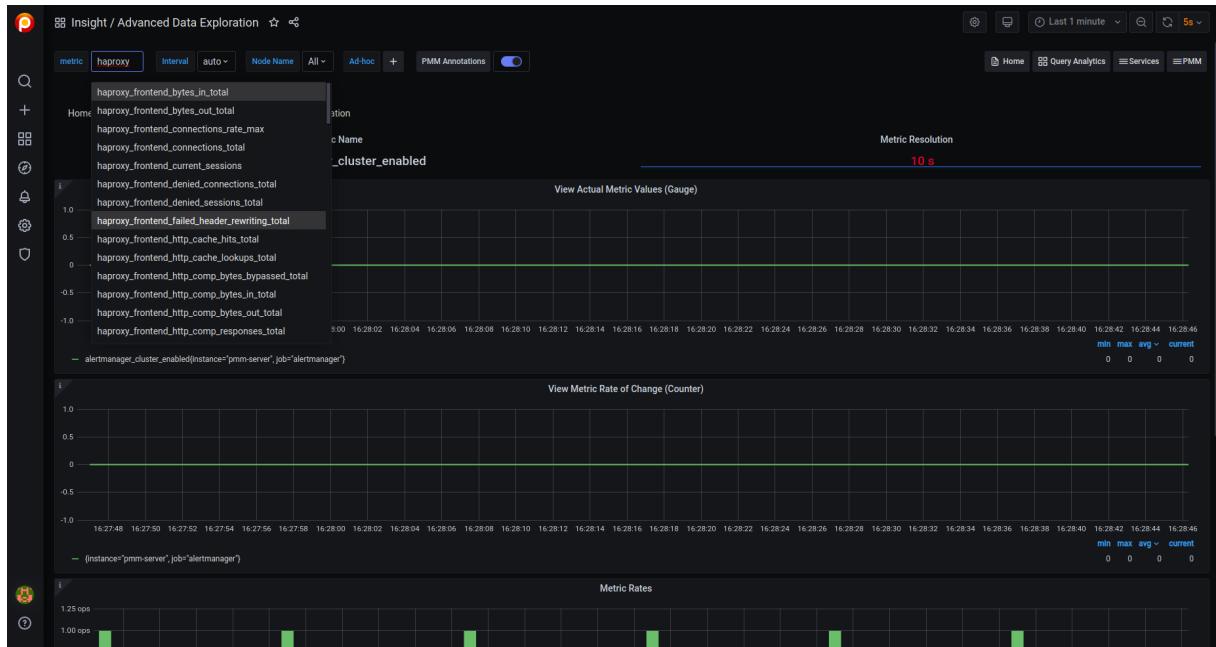
Beside positional argument shown above you can specify service name with the following flags: `--username`, `--password`, `--metrics-path` (path for scraping metrics, default: `/metrics`) and `--scheme` (`http` or `https`). Here are some examples:

```
pmm-admin add haproxy --listen-port=8404 --username=pmm --password=pmm new-haproxy
pmm-admin add haproxy --listen-port=8404 --metrics-path=/prom-metrics --scheme=https
```

Here you can check list of all available flags: `pmm-admin`.

You can also add HAProxy by UI in Grafana: Select `Configuration` →  `PMM Inventory` →  `Add Instance`.

HAProxy data is visible in the *Advanced Data Exploration* dashboard:



Last update: 2021-07-07

## 2.3.12 Remote instances

### Recommended settings

When monitoring remote instances including RDS and Google instances, network latency might affect the scrape process and throw timeout errors. For this reason, it is recommended to [lower the metrics resolution](#).

Starting with PMM 2.18, the scrape timeout has been updated according to the following rules:

- For resolutions <= 2 seconds, scrape timeout is 1 second.
- For resolutions <= 10 seconds, timeout is set to resolution minus 1 second. For example, for 10 second resolution, timeout will be set at 9 seconds.
- For lower resolutions (values > 10 seconds), the scrape timeout is set to 90% of the resolution time. For example, for 60 second resolution, the scrape timeout will be set to 54 seconds.

### How to check for scrape timeouts

Sometimes it is hard to check if you are using the correct values to scrape or if there some other reason why there is no data in a dashboard even when the instance has been added correctly and the agent is running.

One additional step you can do is to check for scrape target statuses. Browse to <http://<your-pmm-server-address>/prometheus/targets> and then click on the Unhealthy button.

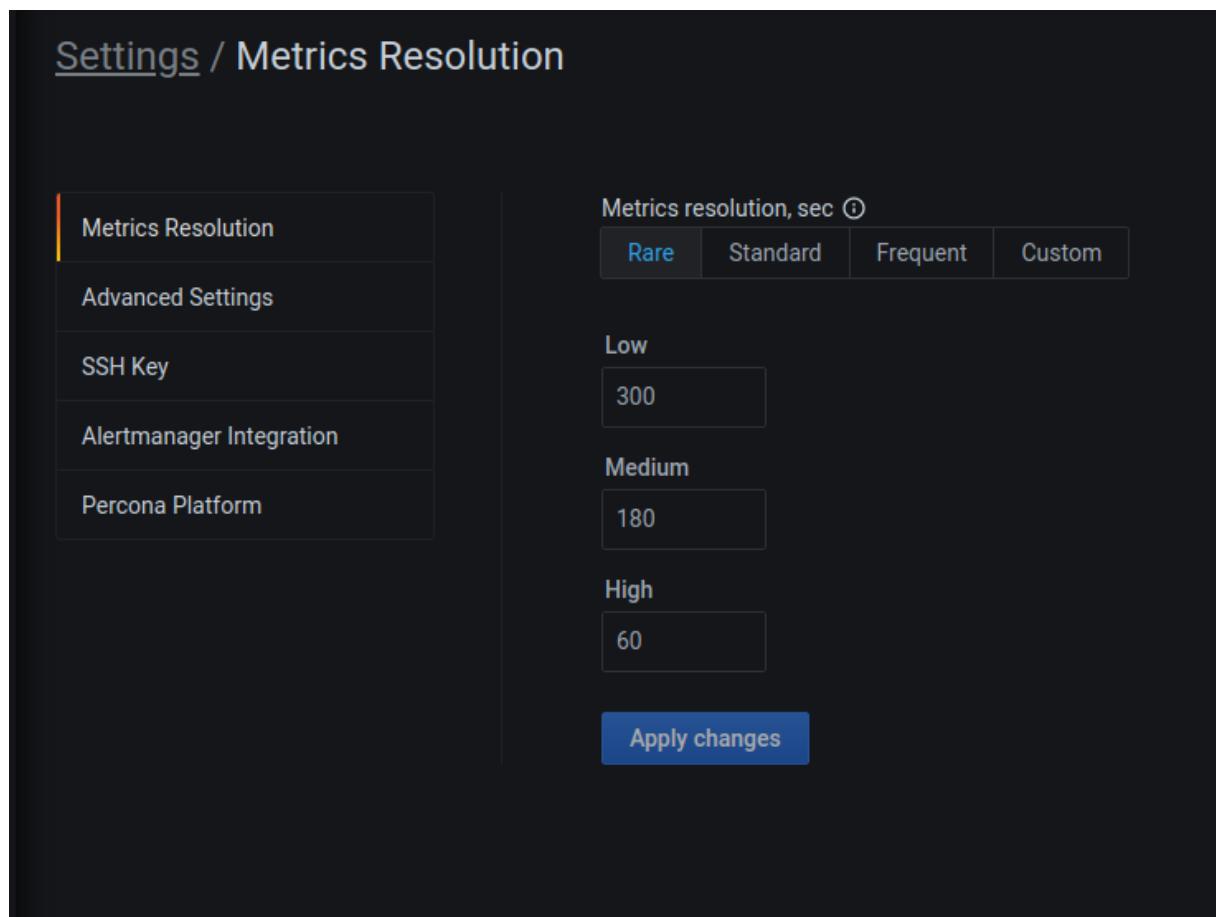
| Endpoint                                                                                                                                                                                                                                                                                                                        | State | Labels                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Last Scrape | Scrape Duration | Error                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="http://127.0.0.1:42002/metrics?collect%5B%5D=custom_query.hr&amp;collect%5B%5D=exporter&amp;collect%5B%5D=standard.go&amp;collect%5B%5D=standard.process">http://127.0.0.1:42002/metrics?collect%5B%5D=custom_query.hr&amp;collect%5B%5D=exporter&amp;collect%5B%5D=standard.go&amp;collect%5B%5D=standard.process</a> | DOWN  | agent_id="agent_id/19d973f9-3928-4ab8-b24f-0afa2cb7faf3"<br>agent_type="postgres_exporter" az="us-east-1" instance="/agent_id/19d973f9-3928-4ab8-b24f-0afa2cb7faf3"<br>job="postgres_exporter_agent_id_19d973f9-3928-4ab8-b24f-0afa2cb7faf3_hr-1m0s"<br>node_id="/node_id/5a29958-f7e4-476c-8170-e8e81eb41377"<br>node_name="pmm-qa-postgres-12" node_type="remote_rds" region="us-east-1" service_id="/service_id/0927b6c4-a4ed-40f6-bd41-31f51f661f05"<br>service_name="pmm-qa-postgres-12" service_type="postgresql" | 11.660s ago | 10.001s         | cannot read data: cannot scrape "http://127.0.0.1:42002/metrics?collect%5B%5D=custom_query.hr&collect%5B%5D=exporter&collect%5B%5D=standard.go&collect%5B%5D=standard.process": Get "http://127.0.0.1:42002/metrics?collect%5B%5D=custom_query.hr&collect%5B%5D=exporter&collect%5B%5D=standard.go&collect%5B%5D=standard.process" context deadline exceeded |

The page will show only agents having issues while scrapping and the scrape result including the error messages.

| Endpoint                                                                                                                                                                                                                                                                                                                        | State | Labels                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Last Scrape | Scrape Duration | Error                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="http://127.0.0.1:42002/metrics?collect%5B%5D=custom_query.hr&amp;collect%5B%5D=exporter&amp;collect%5B%5D=standard.go&amp;collect%5B%5D=standard.process">http://127.0.0.1:42002/metrics?collect%5B%5D=custom_query.hr&amp;collect%5B%5D=exporter&amp;collect%5B%5D=standard.go&amp;collect%5B%5D=standard.process</a> | DOWN  | agent_id="agent_id/19d973f9-3928-4ab8-b24f-0afa2cb7faf3"<br>agent_type="postgres_exporter" az="us-east-1" instance="/agent_id/19d973f9-3928-4ab8-b24f-0afa2cb7faf3"<br>job="postgres_exporter_agent_id_19d973f9-3928-4ab8-b24f-0afa2cb7faf3_hr-1m0s"<br>node_id="/node_id/5a29958-f7e4-476c-8170-e8e81eb41377"<br>node_name="pmm-qa-postgres-12" node_type="remote_rds" region="us-east-1" service_id="/service_id/0927b6c4-a4ed-40f6-bd41-31f51f661f05"<br>service_name="pmm-qa-postgres-12" service_type="postgresql" | 11.660s ago | 10.001s         | cannot read data: cannot scrape "http://127.0.0.1:42002/metrics?collect%5B%5D=custom_query.hr&collect%5B%5D=exporter&collect%5B%5D=standard.go&collect%5B%5D=standard.process": Get "http://127.0.0.1:42002/metrics?collect%5B%5D=custom_query.hr&collect%5B%5D=exporter&collect%5B%5D=standard.go&collect%5B%5D=standard.process" context deadline exceeded |

In the example here, there is a message that says: *context deadline exceeded* and the scrape duration column says the scrape took 10 seconds; this means that the exporter didn't respond in the 10 seconds the scrape process was allowed to run due to the configured metric resolutions and their timeouts.

In this case, we can lower the metric resolutions increasing these values as shown in the image below.



---

Last update: 2021-06-29

## 3. Using

### 3.1 Using

- [User Interface](#)
  - Using the web-based user interface.
  - Finding dashboards.
  - Rendering dashboard images.
  - Viewing graph details.
  - Annotating events.
- [Integrated alerting](#)
- [Backup and Restore](#)
- [Query Analytics](#), a specialized dashboard for detailed query analysis.
- [Security Threat Tool](#): Enabling and seeing the results of database security checks.
- [DBaaS](#): Configuration for DBaaS and UI for creating Database clusters.

These features are currently only available for PMM Admin users:

- Backup;
- DBaaS;
- Integrated Alerting;
- Security Checks.

To use these features you must be logged in as a PMM Admin user and activate the features.

If you are logged in as a user that has a Viewer or Editor role you'll see an 'insufficient access' message when trying to use these features.

---

Last update: 2021-07-07

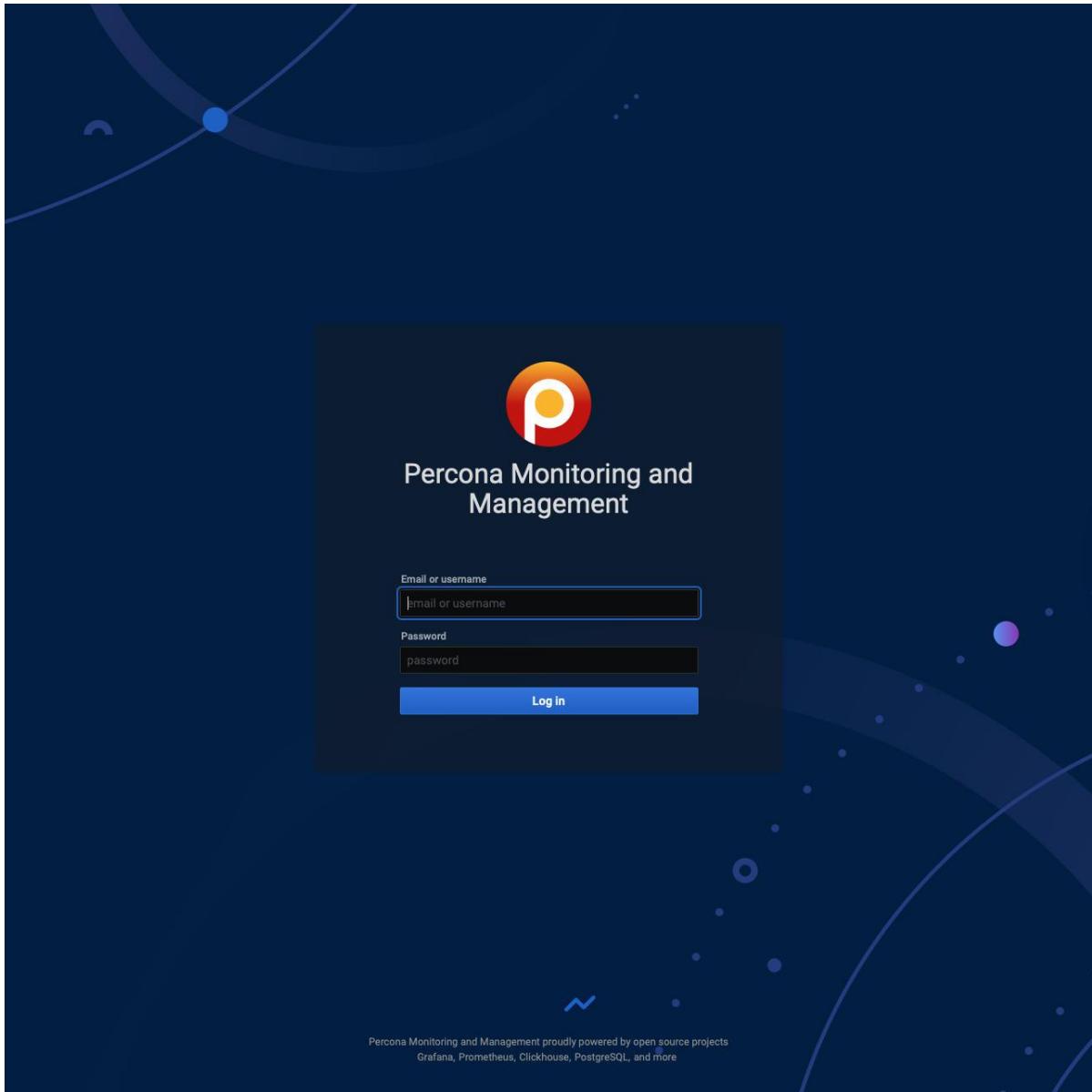
## 3.2 User Interface

How to log in, how the user interface is laid out, and what the controls do.

PMM's user interface is a browser application based on [Grafana](#).

### 3.2.1 Logging in

1. Start a web browser and in the address bar enter the server name or IP address of the PMM server host.
2. The page loads showing the PMM login screen.



3. Enter the username and password given to you by your system administrator. The defaults are:
  - Username: `admin`
  - Password: `admin`
4. Click *Log in*.

5. If this is your first time logging in, you'll be asked to set a new password. (We recommend you do.)

- Either enter a new password in both fields and click *Submit*, or,
- click *Skip* to use the default password.

6. The PMM Home dashboard loads.

### PMM Home dashboard

## 3.2.2 Dashboards

The interface is a collection of web pages called *dashboards*.

Dashboards are grouped into *folders*. You can customize these, by renaming them or creating new ones.

The area inside dashboards is populated by *panels*. Some are in collapsible panel groups. A panel can show a value, a graph, a chart, or a visual representation of a set.

## 3.2.3 Controls

These menus and controls appear on all dashboards:

1. Main menu (also *Grafana menu*, *side menu*).
2. Navigation bar.
3. View controls.
4. View selectors (with dynamic contents).
5. Shortcut menu (with dynamic contents).

(For details see [UI Components](#).)

The screenshot shows the PMM dashboard interface. Key components include:

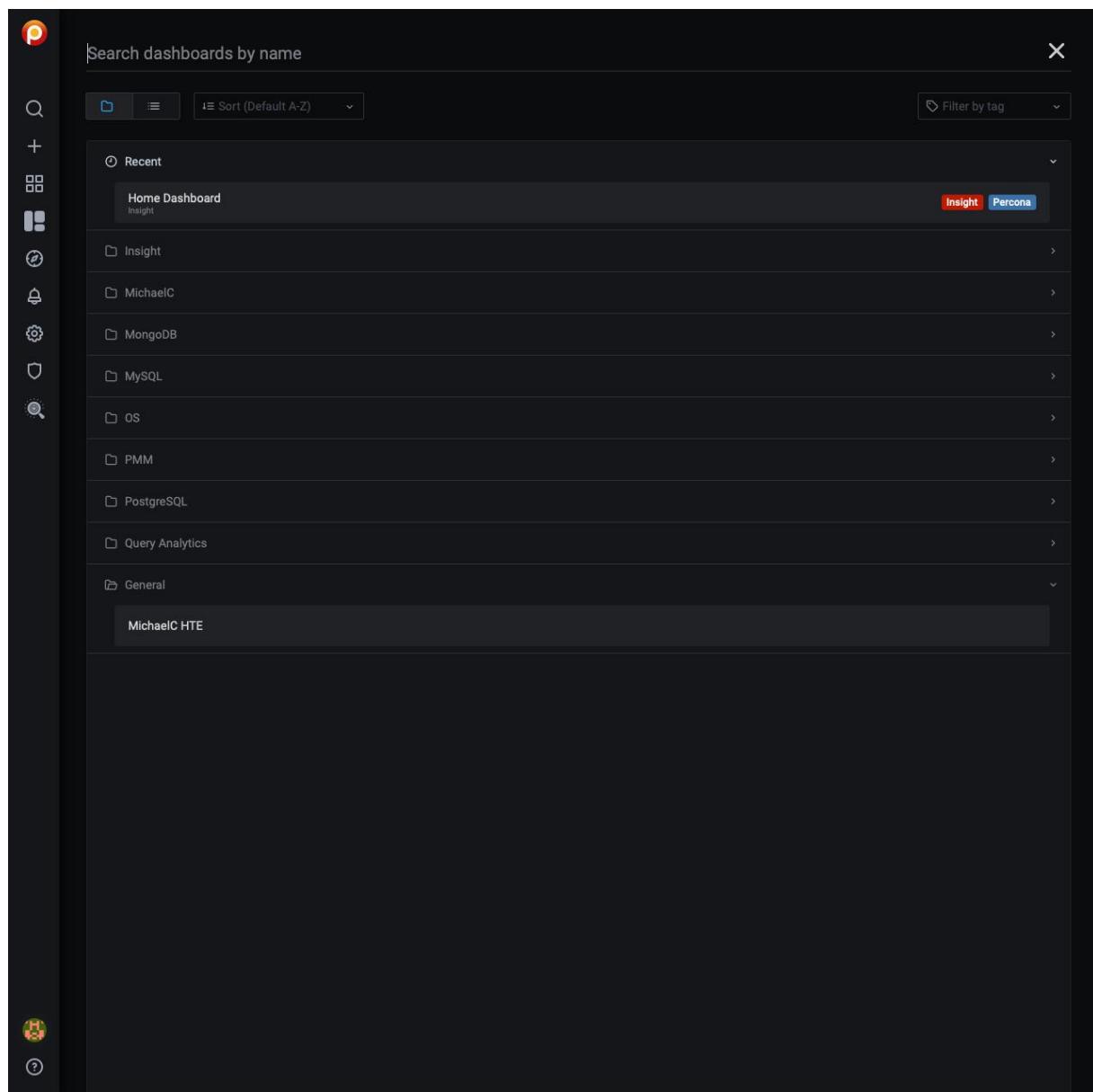
- Left Sidebar:** Contains icons for search (1), plus (+), folder (2), and help (?) (3).
- Top Bar:** Shows the title "Insight / Home Dashboard" (2), a search icon, and a refresh icon.
- Header:** Includes "Interval" dropdown (auto), "Environment" dropdown (All), "Node Name" dropdown (All), and a search bar with a magnifying glass icon (4).
- Header Buttons:** Includes a gear icon, a mail icon, a clock icon labeled "Last 12 hours", a magnifying glass icon, a refresh icon, and a dropdown menu labeled "1m" (5).
- Section Headers:** "General information" (1), "Percona News" (2), "Starred & Recently us..." (3), "Monitored nod..." (4), and "Monitored DB ..." (5).
- Content Areas:**
  - General Information:** Overview of PMM, Documentation, Community and Blogs.
  - Percona News:** Articles like "Using MySQL 8 Dual Passwords" and "Installing Percona Server for MySQL on Rocky Linux 8".
  - Starred & Recently used:** Lists Home Dashboard, PMM Query Analytics, ProxySQL Instance Summary, VictoriaMetrics Agents Overview, and VictoriaMetrics.
  - Monitored nodes:** Shows 30 monitored nodes with a count of 30 and 2 failed security checks (2/9/2).
  - Monitored databases:** Shows 28 monitored databases.
  - PMM Upgrade:** Current version: 2.19.0 (June 30, 2021). Status: You are up to date.
- Environment Overview:** A grid of metrics for different hosts. For example, the first host has CPU usage at 28%, memory usage at 52%, disk read at 2.0 MB/s, and network traffic at 37.2 MB/s.
- Host Details:** Three sections for hosts "AzureDB-mysql", "load-gen-sysbench", and "N/A". Each section shows metrics like CPU usage, memory usage, disk read, network traffic, and other system statistics.

### 3.2.4 Navigation

#### Search for a dashboard by name

There are two ways to open the dashboard search page. (Each takes you to the same search screen.)

- Click the icon in the main menu.
- Click the dashboard name in the navigation bar (top row, to the right of the icon). (To search within the current folder, click the folder name instead of the dashboard name.)



1. Click *Search dashboards by name* and begin typing any part of the dashboard name (in this example, “*Instances*”).

2. Click one of the search results to go to that dashboard. Change the search text to refine the list.

3. To abandon the search, click the icon at the end of the search bar.

#### Open a dashboard with the menu

In the **main menu**, the *PMM Dashboards* icon reveals a submenu containing links to all PMM dashboards grouped by service type. (This menu will replace the **shortcut menu** which has links to commonly-used dashboards.)

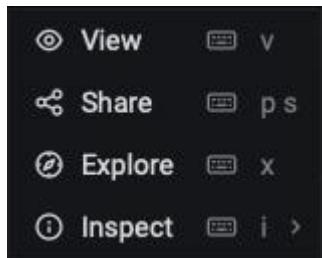
### 3.2.5 Panels

Charts, graphs and set-based panels reveal extra information when the mouse is moved over them.

Some panels have an information icon in the top left corner. Mouse over this to reveal panel information.

## Panel menu

At the top of each panel and to the right of the panel name is the *panel menu*.



### Tip

The menu is hidden until you mouse over it. Look for the symbol in the title bar of a panel.

| Item    | Description                                  |
|---------|----------------------------------------------|
| View    | Open the panel in full window mode.          |
| Share   | Render the panel's image for sharing.        |
| Explore | Run PromQL queries.                          |
| Inspect | See the panel's data or definition.          |
| More    | (Only charts and graphs) Additional options. |

### View

The *View* menu items opens panels in full-window mode. This is useful for graphs with several metrics.

Exit a panel's full window mode by pressing *Escape* or clicking the left arrow next to the dashboard name.

### See also

- [How to render dashboard images](#)
- [How to annotate special events](#)

## 3.2.6 Timezones

By default Grafana uses the timezone from your web browser. However you can change this setting.

### Set user timezone

1. On the left menu, hover your cursor over your avatar and then click *Preferences*.
2. Click to select an option in the *Timezone list*.
3. Click *Save*

Last update: 2021-12-14

## 3.3 Integrated Alerting

 **Caution**

Integrated Alerting is a [technical preview](#) and is subject to change.

*Integrated Alerting* lets you know when certain system events occur.

- Alerts are generated when their criteria (*alert rules*) are met; an *alert* is the result of an *alert rule* expression evaluating to *true*.
- Alert rules are based on *alert rule templates*. We provide a default set of templates. You can also create your own.

 **Summary**

- [Activate Integrated Alerting](#)
- [Set up a communication channel](#)
- [Add a notification channel](#)
- [Add an alert rule](#) (based on a built-in alert rule template)
- (Optional) [Create your own alert rule template](#)

This short video shows how to activate and configure Integrated Alerting.

Your browser does not support playing this video.

### 3.3.1 Before you start

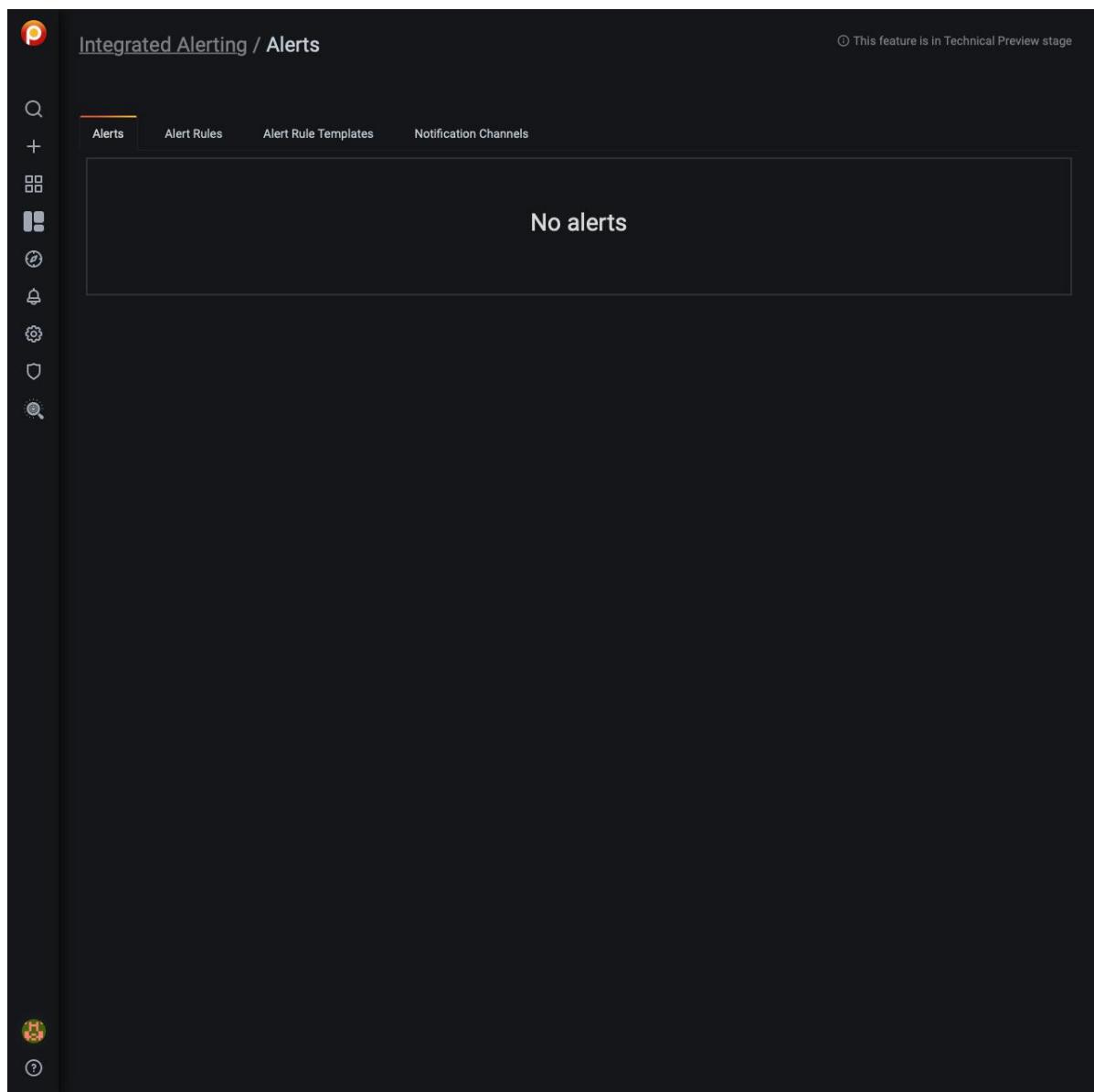
Before you can get alerts, you must [activate Integrated Alerting](#), and [set up a communication channel](#) (define how alerts should arrive, as emails or slack messages).

#### Activate Integrated Alerting

1. Select *Configuration* → *Settings* → *Advanced Settings*.
2. Under *Technical preview features*, turn on *Integrated Alerting*.
3. Click *Apply changes*. This adds a new *Communication* tab to the *Settings* menu.

#### Set up communication channel

1. Open the new *Communication* tab under *Configuration* → *Settings*.
2. Choose whether you want to configure *Email* or *Slack* alerting:
3. For *Email*, fill in the details of your SMTP email server, enforce TLS encryption if your SMTP server requires this, then click *Test* to confirm that you can connect to the SMTP server with the current settings.
4. For *Slack*, specify the Slack webhook URL to use.
5. Click *Apply changes* to save your settings.
6. From the left menu, select *Alerting* → *Integrated Alerting*. The default tab of the *Integrated Alerting* page lists alerts (if set up).



- The *Alerting* menu also lists *Alert Rules* and *Notification Channels*. These are for Grafana's alerting functionality.
- PMM's *Integrated Alerting* is a customized and separate instance of the Prometheus Alertmanager, and distinct from Grafana's alerting functionality.

### 3.3.2 Add a Notification Channel

A *notification channel* is a specific instance of a *communication channel*. For example, for email, the communication channel defines a server, while the notification channel specifies recipients (one or more email addresses) who receive alerts sent via the email server.

1. Select *Alerting* → *Integrated Alerting*.

2. Select the *Notification Channels* tab.

The screenshot shows a dark-themed user interface for managing notification channels. At the top, there's a navigation bar with tabs: 'Alerts', 'Alert Rules', 'Alert Rule Templates', and 'Notification Channels'. The 'Notification Channels' tab is selected. A sub-header 'Integrated Alerting / Notification Channels' is displayed above the main content area. In the top right corner, a note says 'This feature is in Technical Preview stage'. Below the header, there's a table with three columns: 'Name', 'Type', and 'Actions'. Two rows are listed:

| Name           | Type  | Actions |
|----------------|-------|---------|
| Michael Coburn | Email |         |
| roma           | Email |         |

Below the table, there are controls for 'Rows per page' (set to 25) and a page navigation section showing 'Showing 1-2 of 2 items' with buttons for navigating between pages.

3. Click *Add*.

4. Fill in the details:

Integrated Alerting / Notification Channels  ⓘ This feature is in Technical Preview stage

Alerts Alert Rules Alert Rule Templates **Notification Channels**  Add

Name Type Actions

|                |       |                                                                            |
|----------------|-------|----------------------------------------------------------------------------|
| Michael Coburn | Email | <span style="color: #ccc;">Edit</span> <span style="color: #ccc;">X</span> |
| roma           | Email | <span style="color: #ccc;">Edit</span> <span style="color: #ccc;">X</span> |

Rows per page: 25

**Add Notification Channel**

Name:

Type:

Addresses:  
example1@percona.comexample2@percona.comexample3@percona.com

Add Cancel

- Name:
- Type:
  - Email:
    - Addresses:
  - Pager Duty:
    - Routing key:
    - Service key:
  - Slack:
    - Channel:
  - Webhook:
    - URL:
    - Authorization type: Basic
      - Username (optional):
      - Password (optional):
    - Authorization type: Bearer Token
      - Bearer Token (optional):
    - Authorization type: None
  - Use TLS
    - CA Certificate (optional):
    - Certificate (optional):
    - Certificate Key (optional):
    - Server Name (optional):
    - Skip TLS verification:
  - Notify about resolved alerts:
  - Maximum number of alerts to include in message (0 to include all):

5. Click *Add* to add the notification channel, or *Cancel* to abort the operation.

### 3.3.3 Add an Alert Rule

1. Select the *Alert Rules* tab.

Integrated Alerting / Alert Rules

This feature is in Technical Preview stage

Alerts Alert Rules Alert Rule Templates Notification Channels

Add

| Name                                 | Parameters           | Duration  | Severity | Filters | Created                 | Actions                                                                                                                                       |
|--------------------------------------|----------------------|-----------|----------|---------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| MySQL Connections in use more than 1 | Threshold: 1 %       | a minute  | High     |         | 2021-02-04 22:14:04.943 | <input checked="" type="checkbox"/> <input type="button" value="edit"/> <input type="button" value="x"/> <input type="button" value="trash"/> |
| Michael MySQL Down                   |                      | 5 seconds | Critical |         | 2021-04-17 06:51:57.513 | <input checked="" type="checkbox"/> <input type="button" value="edit"/> <input type="button" value="x"/> <input type="button" value="trash"/> |
| roma                                 | Threshold: 1 seconds | 1 seconds | High     |         | 2021-03-05 18:45:41.331 | <input checked="" type="checkbox"/> <input type="button" value="edit"/> <input type="button" value="x"/> <input type="button" value="trash"/> |

Rows per page: 25

Showing 1-3 of 3 items

« < 1 > »

2. Click *Add*.

3. Fill in the details

The screenshot shows the 'Alert Rules' section of the Percona Platform. A modal window titled 'Add Alert Rule' is displayed, containing fields for 'Template' (with a dropdown menu 'Choose'), 'Name' (with a dropdown menu 'Choose'), 'Duration (s)' (with a dropdown menu 'Choose'), 'Severity' (with a dropdown menu 'Choose'), 'Filters' (an empty text area), 'Channels' (with a dropdown menu 'Choose'), and an 'Activate' toggle switch which is turned on. At the bottom of the modal are 'Add' and 'Cancel' buttons.

- Template:
- Name:
- Threshold:
- Duration(s):
- Severity:
- Filters:
- Channels:
- Activate:

4. Click *Add* to add the alert rule, or *Cancel* to abort the operation.

### 3.3.4 Add an Alert Rule Template

If the provided alert rule templates don't do what you want, you can create your own.

- Select the *Alert Rule Templates* tab.

| Name                               | Source   | Created | Actions                                     |
|------------------------------------|----------|---------|---------------------------------------------|
| Memory used by MongoDB connections | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |
| Memory used by MongoDB             | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |
| MongoDB restarted                  | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |
| MySQL down                         | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |
| MySQL restarted                    | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |
| MySQL connections in use           | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |
| Node high CPU load                 | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |
| Node out of memory                 | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |
| Node high swap filling up          | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |
| PostgreSQL down                    | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |
| PostgreSQL restarted               | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |
| PostgreSQL connections in use      | Built-in |         | <a href="#">Edit</a> <a href="#">Delete</a> |

- Click *Add*.

- Enter a template in the *Alert Rule Template* text box.

```
---
templates:
  - name: mysql_too_many_connections
    version: 1
    summary: MySQL connections in use
    tiers: [anonymous, registered]
    expr: |-
      max_over_time(mysql_global_status_threads_connected[5m]) / ignoring (job)
      mysql_global_variables_max_connections
      * 100
      > [[ .threshold ]]
    params:
      - name: threshold
        summary: A percentage from configured maximum
        unit: '%'
        type: float
        range: [0, 100]
```

```

    value: 80
for: 5m
severity: warning
labels:
  foo: bar
annotations:
  description: |-  

    More than [[ .threshold ]]% of MySQL connections are in use on  

{{ $labels.instance }}
  VALUE = {{ $value }}
  LABELS: {{ $labels }}
summary: MySQL too many connections (instance {{ $labels.instance }})

```

Integrated Alerting / Alert Rule Templates  ⓘ This feature is in Technical Preview stage

Alerts Alert Rules Alert Rule Templates Notification Channels Add

| Name                               | Source   | Created | Actions                          |
|------------------------------------|----------|---------|----------------------------------|
| Memory used by MongoDB connections | Built-in |         | <span>edit</span> <span>×</span> |
| Memory used by MongoDB             | Built-in |         | <span>edit</span> <span>×</span> |
| MongoDB restarted                  |          |         | <span>edit</span> <span>×</span> |
| MySQL down                         |          |         | <span>edit</span> <span>×</span> |
| MySQL restarted                    |          |         | <span>edit</span> <span>×</span> |
| MySQL connections in use           |          |         | <span>edit</span> <span>×</span> |
| Node high CPU load                 |          |         | <span>edit</span> <span>×</span> |
| Node out of memory                 |          |         | <span>edit</span> <span>×</span> |
| Node high swap filling up          |          |         | <span>edit</span> <span>×</span> |
| PostgreSQL down                    |          |         | <span>edit</span> <span>×</span> |
| PostgreSQL restarted               |          |         | <span>edit</span> <span>×</span> |
| PostgreSQL connections in use      | Built-in |         | <span>edit</span> <span>×</span> |

Rows per page: 25 Showing 1-12 of 12 items

Upload Add Cancel



### Alert Rule Template parameters

The parameters used in the template follow a format and might include different fields depending on their `type`:

- `name` (required): the name of the parameter. Spaces and special characters not allowed.
- `summary` (required): a short description of what this parameter represents.
- `type` (required): PMM currently supports the `float` type. (More will be available in the future, such as `string` or `bool`.)
- `unit` (optional): PMM currently supports either `s` (seconds) or `%` (percentage).
- `value` (optional): the parameter value itself.
- `range` (optional): only for `float` parameters, defining the boundaries for the value.

#### Restrictions

- Value strings must not include any of these special characters: < > ! @ # \$ % ^ & \* ( ) \_ / \ ' + - = (space)
- Any variables must be predefined.

4. Click *Add* to add the alert rule template, or *Cancel* to abort the operation.

---

Last update: 2022-02-08

## 3.4 Backup and Restore

### ⚠ Caution

- Backup and restore are [technical preview](#) features.
- Currently supported: MySQL database server or MongoDB replica set cluster, backing up to Amazon AWS S3 storage locations.

### ☰ Summary

- [Enable backup features](#).
- [Add a storage location](#).
- [Satisfy preconditions](#):
  - For [MySQL](#):
    - Confirm instance service parameters and storage location.
    - Install required packages.
  - For [MongoDB](#):
    - Install and run [Percona Backup for MongoDB](#) on every node in the replica set.
- [Make a backup](#), or,
- [Make or edit a scheduled backup](#), or,
- [Enable MongoDB Point-In-Time-Recoverable Backups](#), or,
- [Restore a backup](#).
- [Delete a backup](#).

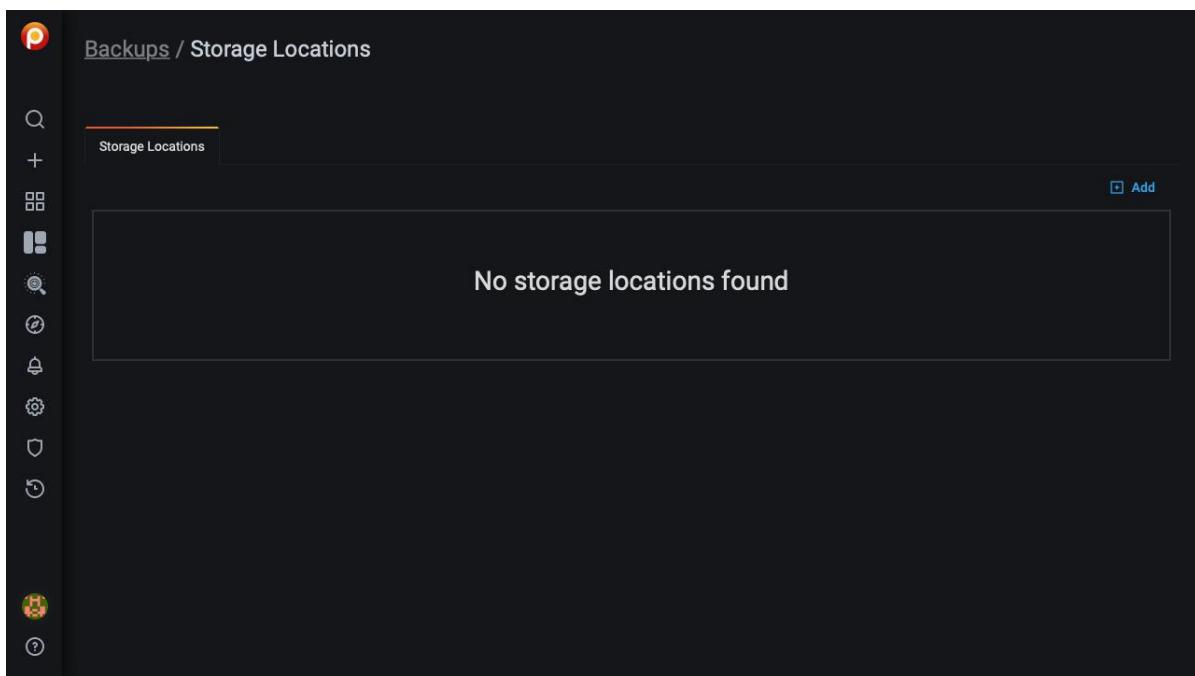
### 3.4.1 Before you start

- You have an AWS S3 storage account and location details for it.
- Backup management has been enabled:
  - a. Select *Configuration* → *Settings* → *Advanced Settings*.
  - b. Activate *Backup Management*.
  - c. Click *Apply changes*.

If PMM Server runs as a Docker container, enable backup features at container creation time by adding `-e ENABLE_BACKUP_MANAGEMENT=1` to your `docker run` command.

### 3.4.2 Add a storage location

1. Select → *Backup*.
2. Select *Storage locations*.



3. Click *Add*.

4. Fill in the form fields.

The screenshot shows the 'Add Storage Location' dialog box. The 'Name' field is empty. The 'Description' field is also empty. Under 'Type', the 'S3' tab is selected, while 'Local Client' and 'Local Server' are unselected. The 'Endpoint' field contains the URL 'http://127.0.0.1:9000'. The 'Bucket Name' field is empty. The 'Access Key' and 'Secret Key' fields are empty. At the bottom of the dialog are three buttons: 'Add' (blue), 'Test' (green), and 'Cancel' (grey).

- *Name*: A short name for this location.
- *Description*: A long description for this location.
- *Type*: Choose the type of storage:
  - S3: Use [Amazon AWS S3](#)
    - *Endpoint*: The S3 backup location endpoint (URL).
    - *Bucket Name*: The bucket name.
    - *Access Key*: The access key string.
    - *Secret Key*: The secret key string. (Click to reveal and to hide again.)
    - *Local Client*: (Not currently implemented)
    - *Local Server*: (Not currently implemented)

5. (Optional) Click *Test* to test the connection.

6. Click *Add* to add the location.

### 3.4.3 MySQL backup preconditions

- [PMM Client](#) is installed and running on the node.
- There is only one MySQL instance running on the node.
- MySQL is running:
  - as a service via `systemd`;
  - with the name `mysql` (to confirm, use `systemctl status mysql`);
  - from a `mysql` system user account.
- There is a `mysql` system group.
- MySQL is using the `/var/lib/mysql` directory for database storage.
- `pmm-agent` has read/write permissions to the `/var/lib/mysql` directory.
- The latest versions of the following packages are installed. They should be included in the `$PATH` environment variable:
  - `xtrabackup`, which includes:
    - `xbcloud`;
    - `xbstream`;
    - `qpress`.

#### Important

The versions of each must be compatible with the installed version of MySQL.

### 3.4.4 MongoDB backup preconditions

- [Percona Backup for MongoDB](#) is installed and `pbm-agent` is running on all MongoDB nodes in the replica set.
- MongoDB is a member of a replica set.

### 3.4.5 Make a backup

Make a single on-demand backup.

1. Add a storage location.
2. Select → *Backup*.
3. Select *Backup Inventory*.
4. Click *Add*.
5. In the *Backup on demand* dialog, enter values for:
  - *Service name*: Choose from the menu the service to back up.
  - *Vendor*: A value is automatically selected based on the service type.
  - *Backup name*: Enter a unique name for this backup.
  - *Description*: (Optional) Enter a long description for this backup.
  - *Location*: Choose from the menu the storage location.
6. Click *Backup*.
7. In the *Backup Inventory* pane, watch the *Status* column. An animated ellipsis indicator shows activity in progress.

### 3.4.6 Make a scheduled backup

Make regular scheduled backups.

1. Add a storage location.
2. Select → *Backup*.
3. Select *Scheduled Backups*.
4. Click *Add*.

5. In the *Schedule backup* dialog, enter values for:

- *Service name*: Choose from the menu the service to back up.
- *Backup name* : Enter a unique name for this scheduled backup.
- *Vendor*: A value is automatically selected based on the service type.
- *Location*: Choose from the menu the storage location.
- *Data model*: Select one of the options:
  - *Physical*: Backup the physical data model.
  - *Logical*: (Not currently implemented)
- *Description*: (Optional) Enter a long description for this scheduled backup.
- *Schedule*: The schedule for the backup.
  - *Every*: The backup interval. Choose from the menu one of:
    - *Year*
    - *Month*
    - *Week*
    - *Day*
    - *Hour*
    - *Minute*
  - Depending on the interval chosen, the remaining options will be active or inactive.
    - *Month*: Select one or more months.
    - *Day*: Select one or more day numbers.
    - *Weekday*: Select one or more week day names.
    - *Start time, h/m*: The hour and minute for the backup.
      - In the first field, select one or more hours (00 to 23, 00 is midnight).
      - In the second field, select one or more minutes (00 to 59).
  - *Retention*: How many backups to keep. For unlimited, use **0** (zero).
  - *Retry mode*: In case of error, either let PMM retry the backup again ("Auto") or do it again yourself ("Manual").
  - *Retry times and interval*: If "Auto" retry mode is selected, the maximum number of retries - up to 10 - and the interval between them - up to 8 hours - can be set here.
  - *Enable*: Deselect to define the scheduled backup without enabling it.

Schedule - UTC time

|                                                                      |                                                                      |
|----------------------------------------------------------------------|----------------------------------------------------------------------|
| Every                                                                | Month                                                                |
| Year                                                                 | Every                                                                |
| Day                                                                  | Weekday                                                              |
| Every                                                                | Every                                                                |
| Start time, h/m                                                      |                                                                      |
| 00 <input type="button" value="x"/> <input type="button" value="x"/> | 00 <input type="button" value="x"/> <input type="button" value="x"/> |
| Retention (Number of backups - 0 for unlimited)                      |                                                                      |
| 7                                                                    |                                                                      |
| Retry mode                                                           |                                                                      |
| Auto                                                                 | Manual                                                               |
| Retry, times                                                         | Retry interval, seconds                                              |
| 2                                                                    | 30                                                                   |
| <input checked="" type="checkbox"/> Enabled                          |                                                                      |

For this release (2.26.0), times are UTC.

6. Click *Schedule*.
7. A new entry will appear in the list.

### 3.4.7 Edit a scheduled backup

1. Select → *Backup*.
2. Select *Scheduled Backups*.

3. In the *Actions* column:

- Click the switch to enable or disable the backup.
- Click to edit the backup schedule.
- Click to delete the backup schedule.
- Click to create a (by default, disabled) copy of the backup schedule.

| Name                     | Vendor | Frequency                          | Type | Location | Last backup | Actions                             |
|--------------------------|--------|------------------------------------|------|----------|-------------|-------------------------------------|
| Copy of backup-for-mysql | MySQL  | At 12:00 AM, on day 5 of the month | Full | s3       |             | <input checked="" type="checkbox"/> |
| backup-for-mysql         | MySQL  | At 12:00 AM, on day 5 of the month | Full | s3       |             | <input type="checkbox"/>            |

### 3.4.8 MongoDB Point-In-Time-Recoverable Backups

**Caution**

- MongoDB Point-In-Time-Recoverable Backups is part of Backup Management which is a [technical preview](#) feature.

#### What is it?

Better described by our team mates that develop Percona Backup for MongoDB:

Point-in-Time Recovery is restoring a database up to a specific moment. Point-in-Time Recovery includes restoring the data from a backup snapshot and replaying all events that occurred to this data up to a specified moment from [oplog slices](#). Point-in-Time Recovery helps you prevent data loss during a disaster such as crashed database, accidental data deletion or drop of tables, unwanted update of multiple fields instead of a single one.

Point-In-Time-Recovery (PITR) Backups for MongoDB is new functionality available with PMM 2.23.0 as part of the larger Backup Management feature. This implementation in PMM uses Percona Backup for MongoDB `pbm >= 1.6.0` behind the scenes.

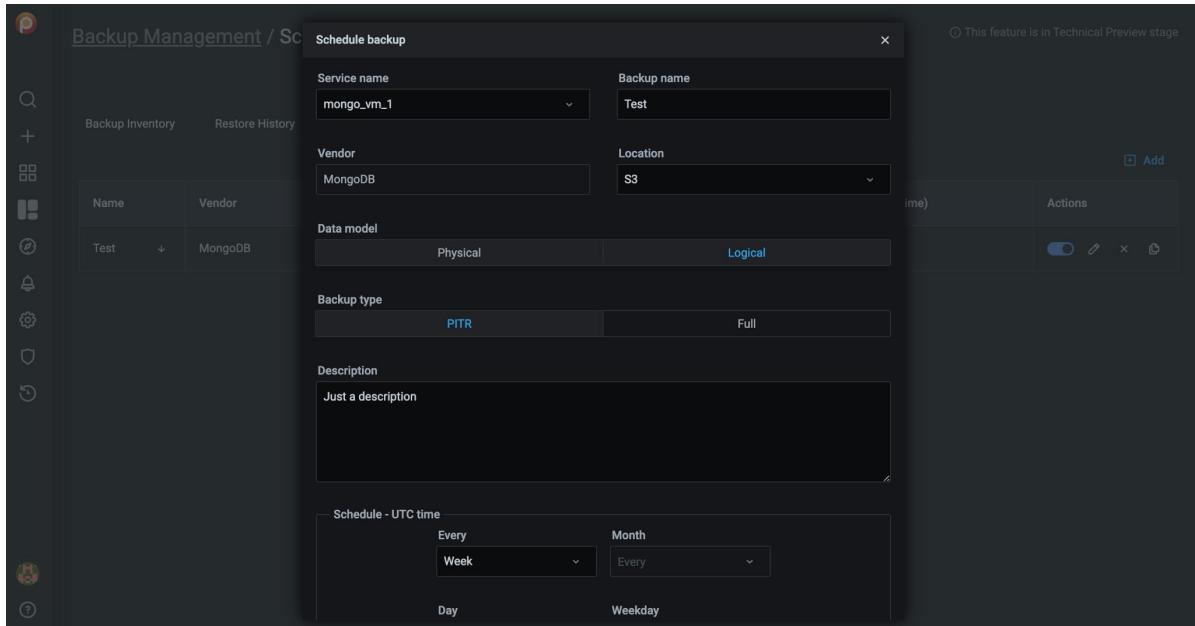
- Percona Backup for MongoDB is a distributed, low-impact solution for achieving consistent backups of MongoDB sharded clusters and replica sets. Percona Backup for MongoDB supports [Percona Server for MongoDB](#) and MongoDB Community v3.6 or higher with [MongoDB Replication](#) enabled. Learn more about [Percona Backup for MongoDB](#).

## How does it work?

### ENABLING PITR

The very first thing you want to do is to enable PITR. Here's how:

1. Go to *Backup Management*.
2. Select *Scheduled Backups*.
3. Click on *Add* to create a new scheduled backup.
4. Click on the PITR button to enable Point-In-Time-Recovery.



Once you've enabled PITR, head to the list of Scheduled Backups to confirm PITR is enabled.

| Backup Management / Scheduled Backups |         |                   |           |                   |          |                          |                                                                                                          |
|---------------------------------------|---------|-------------------|-----------|-------------------|----------|--------------------------|----------------------------------------------------------------------------------------------------------|
| Backup Inventory                      |         | Scheduled Backups |           | Storage Locations |          |                          |                                                                                                          |
| Name                                  | Vendor  | Frequency         | Retention | Type              | Location | Last backup (local time) | Actions                                                                                                  |
| PITR_test                             | MongoDB | Every minute      | 7 backups | PITR              | S3       | 2021-10-14 15:59:00      | <input checked="" type="checkbox"/> <input type="button"/> <input type="button"/> <input type="button"/> |
| Test                                  | MongoDB | At 12:00 AM       | 7 backups | Full              | S3       |                          | <input type="checkbox"/> <input type="button"/> <input type="button"/> <input type="button"/>            |

To disable PITR use the corresponding switch available on the list.

## PITR ARTIFACTS

The PITR artifacts will be available once your PITR job has run for the first time. Go to Backup Inventory to see the corresponding PITR artifact.

| Backup name | Vendor  | Created             | Type | Location | Status  | Actions              |
|-------------|---------|---------------------|------|----------|---------|----------------------|
| PITR_test   | MongoDB | 2021-10-14 15:59:00 | PITR | S3       | Success | <a href="#">Logs</a> |

## PITR AND OTHER SCHEDULED BACKUPS

It is important to notice that enabling PITR requires any other scheduled backup jobs to be disabled. If you try to enable PITR while other scheduled backup jobs are active, you will be shown an error message as seen in the image below.

Go ahead to manually disable the existing scheduled jobs, then you'll be able to enable PITR.

The above constraint applies at the service level. That said, you can still have PITR enabled for one service while having regular scheduled backup jobs for other services.

### 3.4.9 Restore a backup

For now, you can only restore a backup to the same service it was created from, or to a compatible one.

1. Select → *Backup* → *Backup Inventory*.
2. Find the row with the backup you want to restore.
3. In the *Actions* column for that row, click *Restore from backup*.
4. In the *Restore from backup* dialog:
  - Select *Same service* to restore to a service with identical properties.
    - Select the service in the *Service name* menu.
  - Select *Compatible services* to restore to a compatible service.
    - Select the compatible service in the *Service name* menu.
5. Check the values and click *Restore*.
6. Navigate to the *Restore History* tab to check the status of the restored backup.

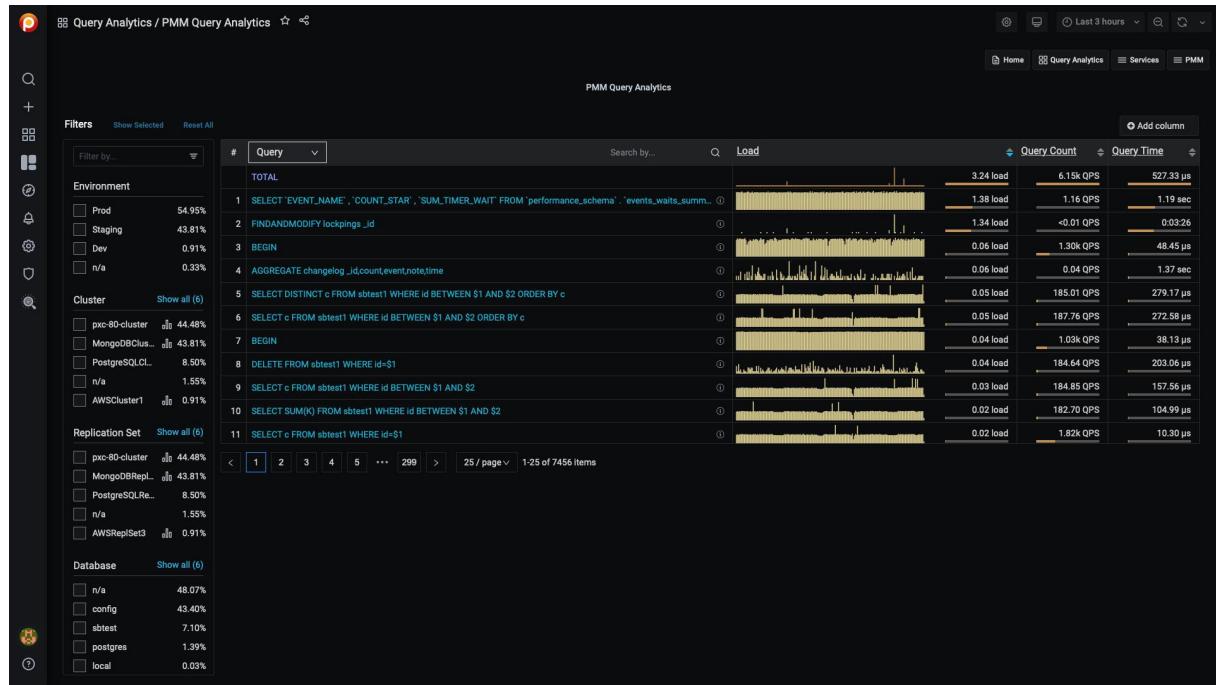
### 3.4.10 Delete a backup

1. Select → *Backup* → *Backup Inventory*.
  2. Find the row with the backup you want to delete.
  3. In the *Actions* column for that row, click *Delete backup*.
  4. (Optional) Check *Delete from storage* to also delete the actual backup content besides just the backup register.
  5. Click *Delete* to proceed.
- 

Last update: 2021-10-21

## 3.5 Query Analytics

The *Query Analytics* dashboard shows how queries are executed and where they spend their time. It helps you analyze database queries over time, optimize database performance, and find and remedy the source of problems.



Query Analytics supports MySQL, MongoDB and PostgreSQL. The minimum requirements for MySQL are:

- MySQL 5.1 or later (if using the slow query log).
- MySQL 5.6.9 or later (if using Performance Schema).

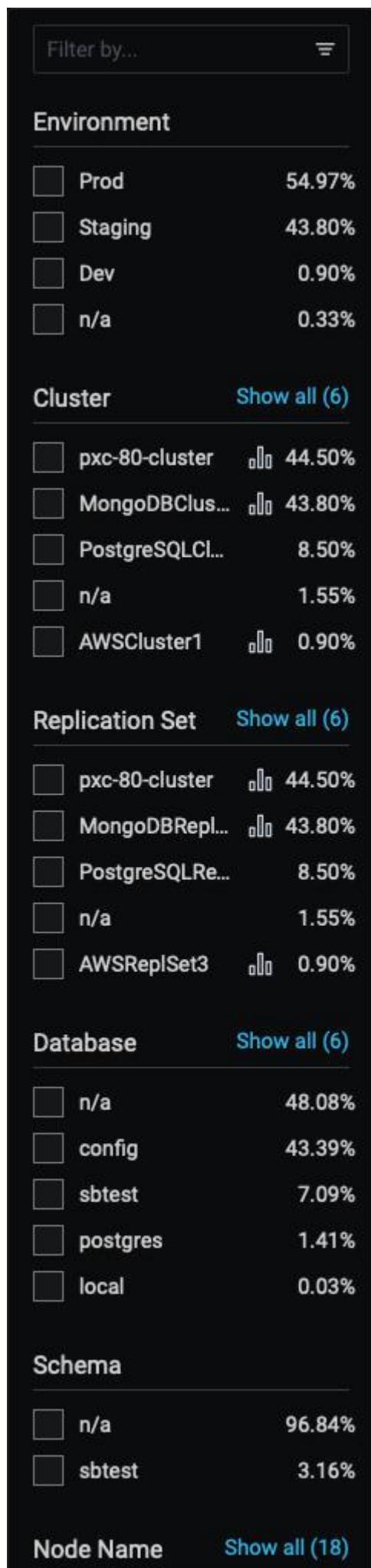
Query Analytics displays metrics in both visual and numeric form. Performance-related characteristics appear as plotted graphics with summaries.

The dashboard contains three panels:

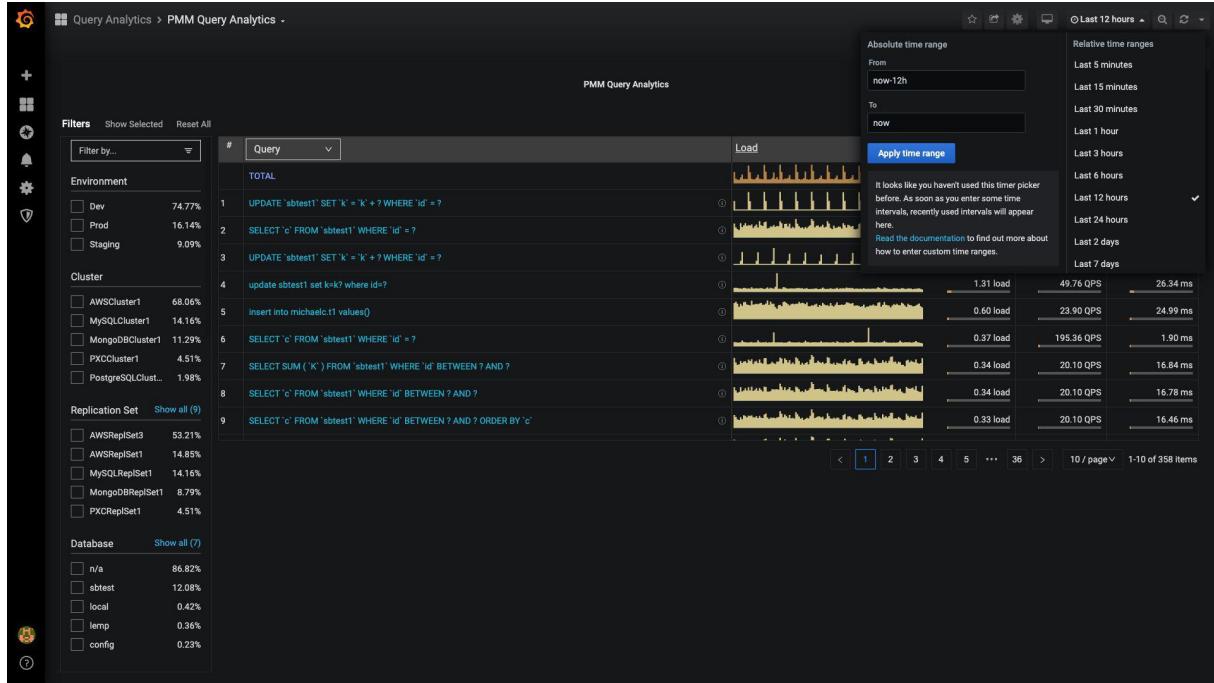
- the [Filters Panel](#);
- the [Overview Panel](#);
- the [Details Panel](#).

Query Analytics data retrieval is not instantaneous and can be delayed due to network conditions. In such situations *no data* is reported and a gap appears in the sparkline.

### 3.5.1 Filters Panel

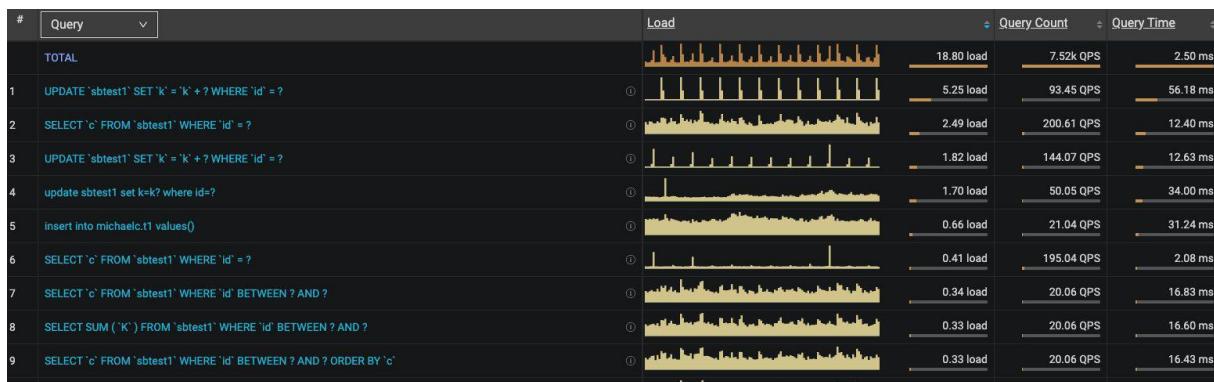


- The Filter panel occupies the left side of the dashboard. It lists filters, grouped by category. Selecting one reduces the Overview list to those items matching the filter.
- The first five of each category are shown. If there are more, the list is expanded by clicking *Show all* beside the category name, and collapsed again with *Show top 5*.
- Applying a filter may make other filters inapplicable. These become grayed out and inactive.
- Click the chart symbol to navigate directly to an item's associated dashboard.
- Separately, the global *Time range* setting filters results by time, either your choice of *Absolute time range*, or one of the predefined *Relative time ranges*.



## 3.5.2 Overview Panel

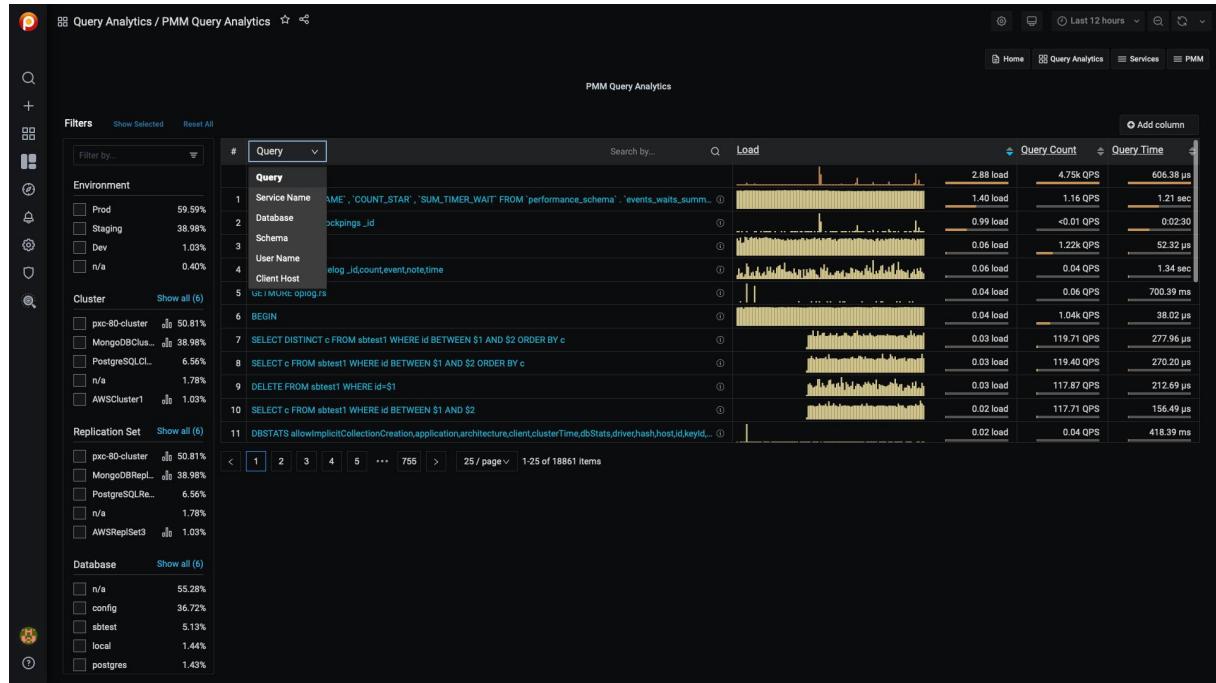
To the right of the Filters panel and occupying the upper part of the dashboard is the Overview panel.



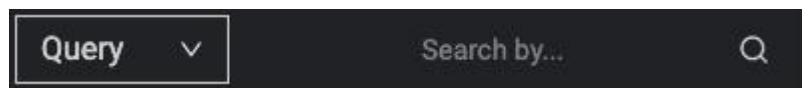
Each row of the table represents the metrics for a chosen object type, one of:

- Query;
- Service Name;
- Database;
- Schema;
- User Name;
- Client Host.

At the top of the second column is the *dimension* menu. Use this to choose the object type.



On the right side of the dimension column is the *Dimension Search* bar.



Enter a string and press Enter to limit the view to queries containing only the specified keywords.

Delete the search text and press Enter to see the full list again.

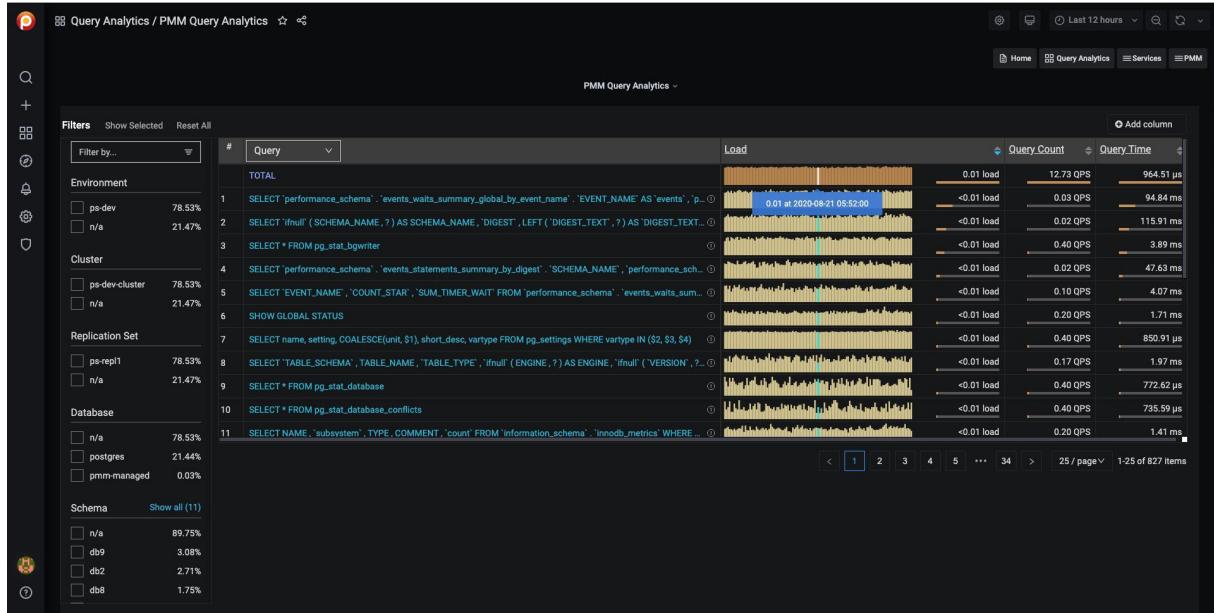
## Columns

- The first column is the object's identifier. For *Query*, it is the query's [Fingerprint](#).
- The second column is the *Main metric*, containing a reduced graphical representation of the metric over time, called a *sparkline*, and a horizontal meter, filled to reflect a percentage of the total value.
- Additional values are revealed as mouse-over tool-tips.

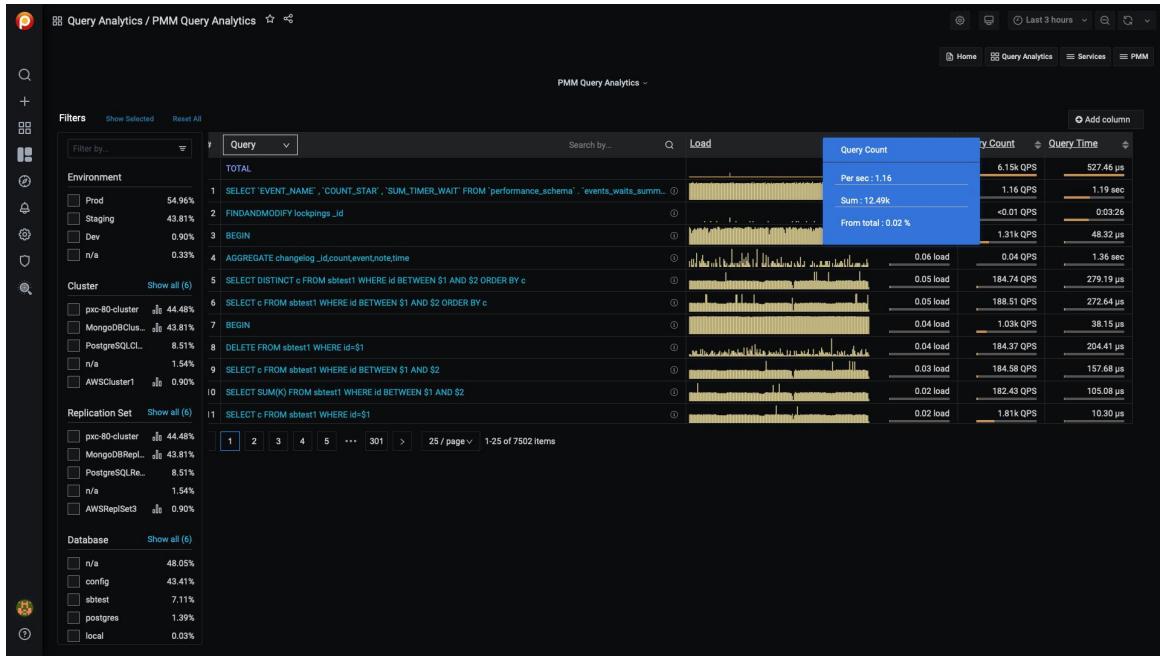
## Tool-tips

- For the *Query* dimension, hovering over the information icon ⓘ reveals the query ID and its example.
- Hovering on a column header reveals an informative tool-tip for that column.

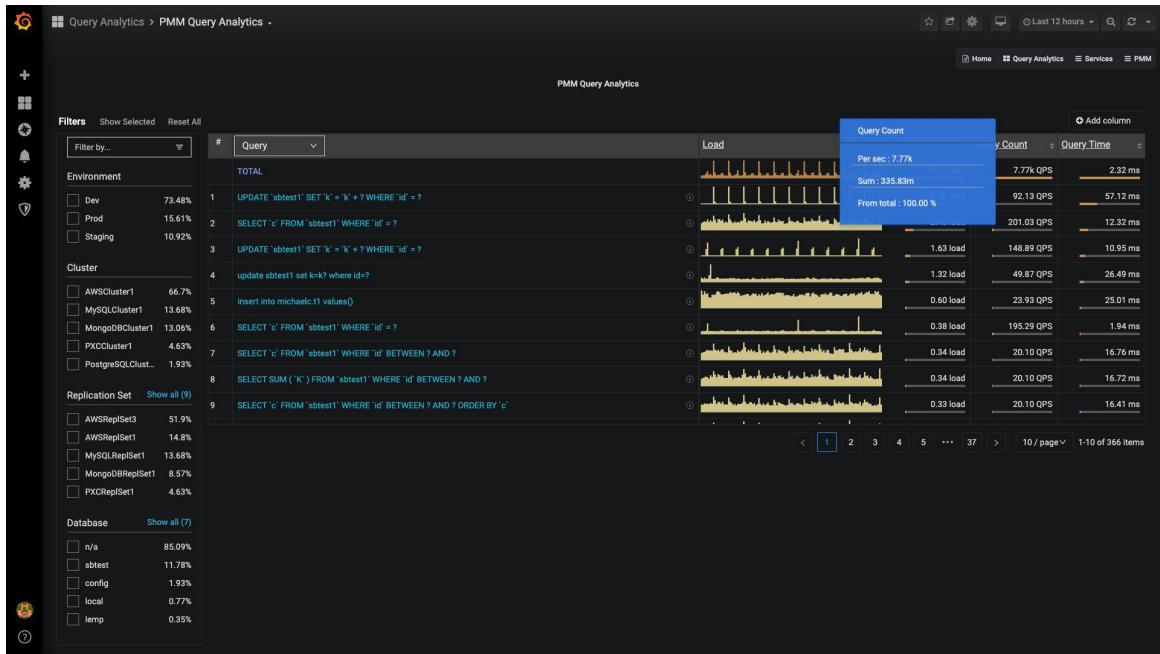
- Hovering on the main metric sparkline highlights the data point and a tooltip shows the data value under the cursor.



- Hovering on the main metric meter reveals the percentage of the total, and other details specific to the main metric.



- Hovering on column values reveals more details on the value. The contents depends on the type of value.



## Adding and removing columns

- Metrics columns are added with the *Add column* button.



- When clicked, a text field and list of available metrics are revealed. Select a metric or enter a search string to reduce the list. Selecting a metric adds it to the panel.
- A metric column is removed by clicking on the column heading and selecting *Remove column*.

- The value plotted in the *main metric* column can be changed by clicking a metric column heading and selecting *Swap with main metric*.

## Sorting

- The entire list is sorted by one of the columns.
- Click either the up or down caret to sort the list by that column's ascending or descending values.

## Pagination

- The pagination device lets you move forwards or backwards through pages, jump to a specific page, and choose how many items are listed per page.



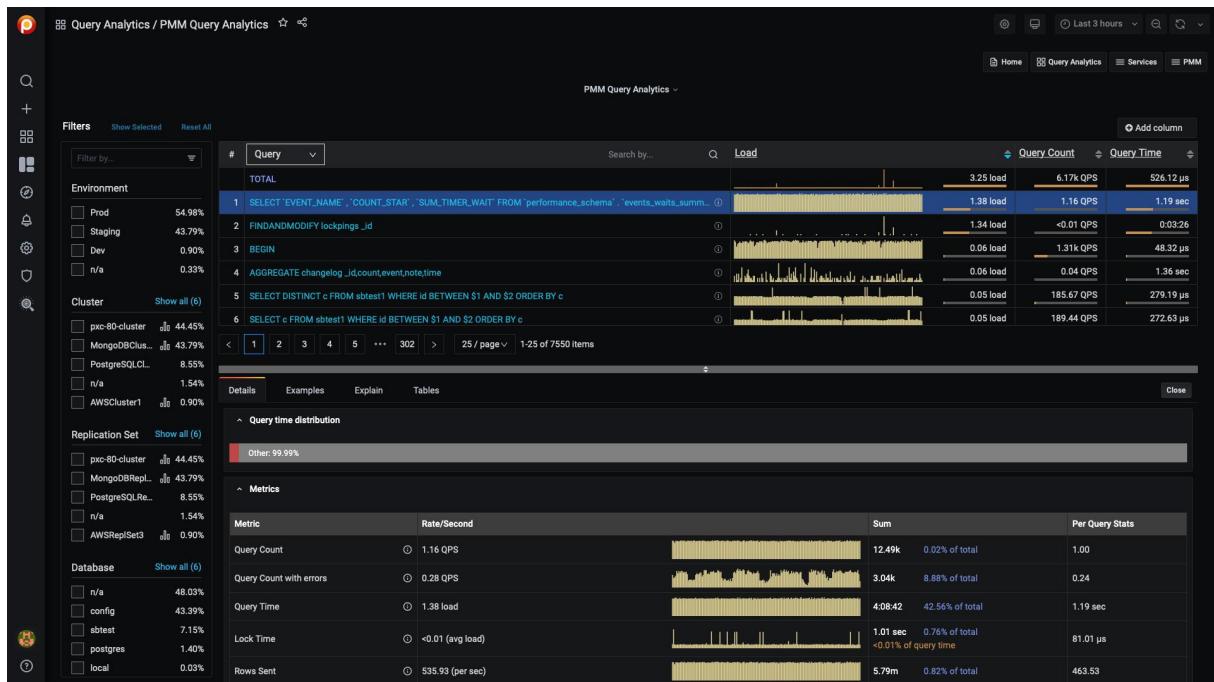
- Queries are grouped into pages of 25, 50 or 100 items.

## 3.5.3 Details Panel

- Selecting an item in the Overview panel opens the *Details panel* with a **Details Tab**.
- If the dimension is *Query*, the panel also contains the **Examples Tab**, **Explain Tab**, and **Tables Tab**.

### Details Tab

The *Details tab* contains a *Query time distribution* bar (only for MySQL databases) and a set of *Metrics* in collapsible subpanels.



- The *Query time distribution* bar shows a query's total time made up of colored segments, each segment representing the proportion of time spent on a named activity.
  - `query_time`: Statement execution time.
  - `lock_time`: Time to acquire locks.
  - `blk_read_time`: Total time the statement spent reading blocks (if `track_io_timing` is enabled, otherwise zero).
  - `blk_write_time`: Total time the statement spent writing blocks (if `track_io_timing` is enabled, otherwise zero).
  - `innodb_io_r_wait`: Time for InnoDB to read the data from storage.
  - `innodb_queue_wait`: Time the query spent either waiting to enter the InnoDB queue, or in it pending execution.
  - `innodb_rec_lock_wait`: Time the query waited for row locks.
  - `other`: Remaining uncategorized query time.
- *Metrics* is a table with headings:
  - *Metric*: The Metric name, with a question-mark tool-tip that reveals a description of the metric on mouse-over;
  - *Rate/Second*: A sparkline chart of real-time values per unit time;
  - *Sum*: A summation of the metric for the selected query, and the percentage of the total;
  - *Per Query Stats*: The value of the metric per query.
- Each row in the table is a metric. The contents depends on the chosen dimension.

For PostgreSQL queries (when using `pg_stat_monitor`) the top query will also be shown in the details section if the query was called by an outer query.

The screenshot shows the PMM Query Analytics interface. On the left, there's a sidebar with various monitoring and configuration icons. The main area has a title bar "Query Analytics / PMM Query Analytics" with a star icon and a refresh button. The top right has a "Last 6 hours" time range selector and other navigation buttons.

**Database**

| Database           | Percentage |
|--------------------|------------|
| contrib_regression | 91.85%     |
| postgres           | 4.34%      |
| pmm-agent          | 3.63%      |
| pmm-managed        | 0.19%      |

**Node Name**

| Node Name            | Percentage |
|----------------------|------------|
| host.docker.internal | 96.99%     |
| pmm-server           | 3.01%      |

**Service Name**

| Service Name         | Percentage |
|----------------------|------------|
| host.docker.internal | 96.99%     |
| pmm-server-p...      | 3.01%      |

**Client Host**

| Client Host | Percentage |
|-------------|------------|
| 172.18.0.3  | 91.85%     |
| 172.18.0.1  | 5.14%      |
| n/a         | 3.01%      |

**User Name**

| User Name   | Percentage |
|-------------|------------|
| "pmm-agent" | 96.99%     |
| pmm-managed | 3.01%      |

**TOTAL**

1 with stat as (select queryid, bucket, unnest(range0)) as ra...

< 0.01 load > 0.01 load

1-1 of 1 items

**Details** Examples Tables Plan Close

**Query time distribution**

Other: 99.98%

**Top Query**

```
SELECT * FROM histogram(0, 'F44CD1B4B33A47AF') AS a(range TEXT, freq INT, bar TEXT);
```

**Metrics**

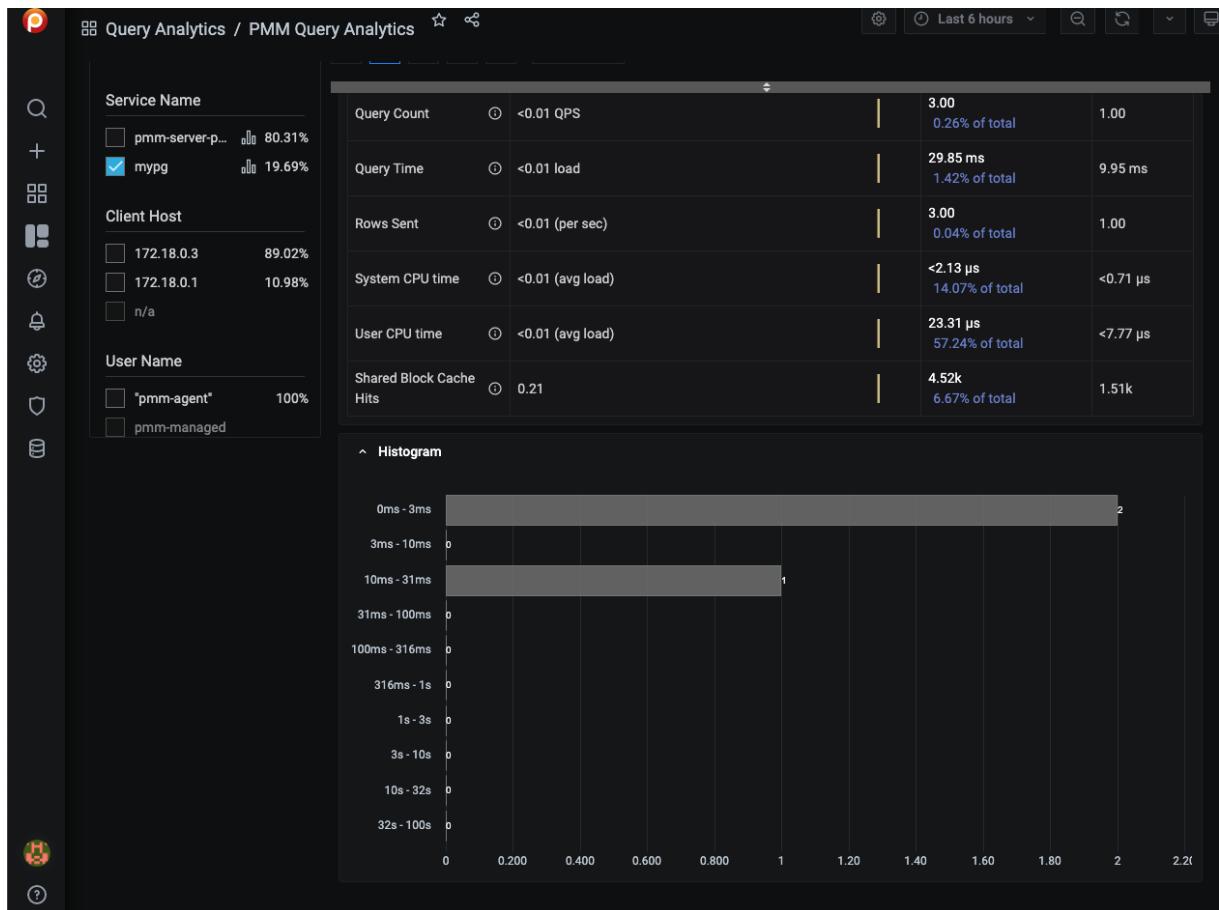
| Metric          | Rate/Second      | Sum                        | Per Query Stats |
|-----------------|------------------|----------------------------|-----------------|
| Query Count     | <0.01 QPS        | 3.00<br>0.06% of total     | 1.00            |
| Query Time      | <0.01 load       | 41.33 ms<br>0.09% of total | 13.78 ms        |
| System CPU time | <0.01 (avg load) | <4.96 µs<br>0.12% of total | <1.65 µs        |
| User CPU time   | <0.01 (avg load) | <1.99 µs                   | <0.66 µs        |

Other useful metrics (when using `pg_stat_monitor`) to monitor PostgreSQL Server performance are [Histograms](#). *Histograms* provide more explicit information about number of queries for fingerprint (queryid). Ranges are from 0 seconds up to 100 seconds.

Ranges (numbers are in milliseconds):

- 0 - 3
- 3 - 10
- 10 - 31
- 31 - 100
- 100 - 316
- 316 - 1000
- 1000 - 3162
- 3162 - 10000
- 10000 - 31622
- 31622 - 100000

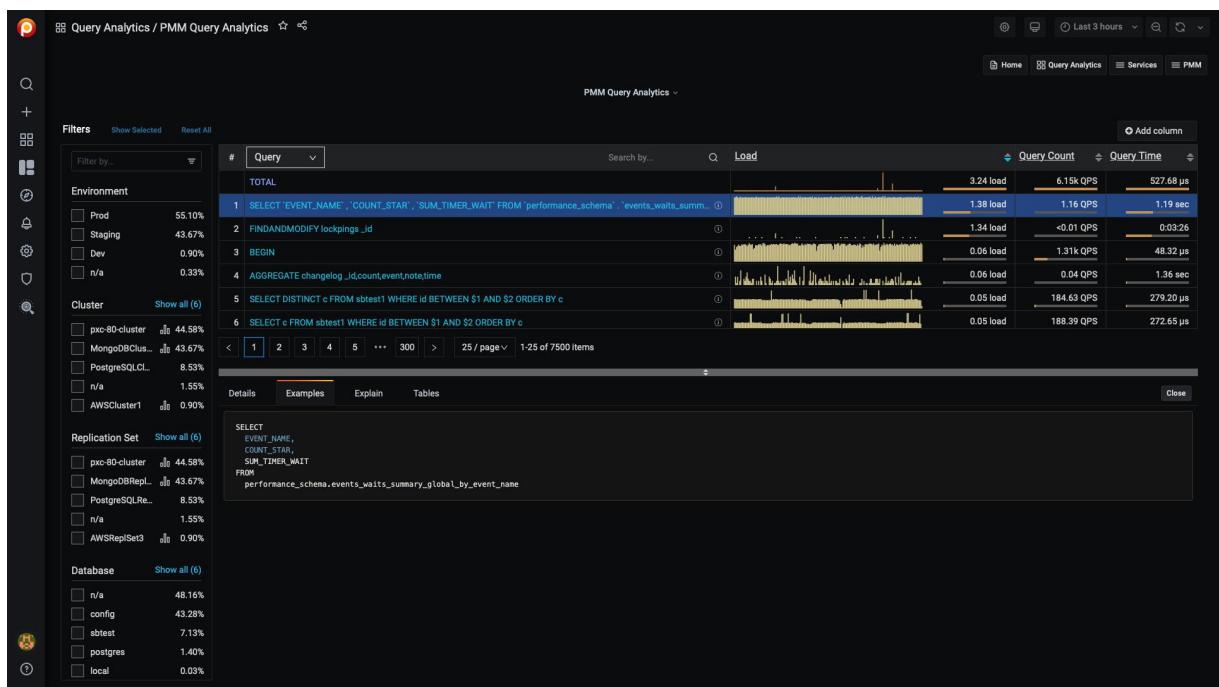
Here is picture of *histogram* in graph:



## Examples Tab

(For Query dimension.)

The *Examples* tab shows an example of the selected query's fingerprint or table element.



Query example and fingerprint can be truncated to 1024 long to reduce space usage. In this case, the query explains section will not work.

## Explain Tab

(For Query dimension.)

The *Explain* tab shows the `explain` output for the selected query, in Classic or JSON formats.

- MySQL: Classic and JSON.
- MongoDB: JSON only.
- PostgreSQL: Not supported.

| # | Query                                                                                                                   | Load      | Query Count | Query Time |
|---|-------------------------------------------------------------------------------------------------------------------------|-----------|-------------|------------|
| 1 | SELECT 'EVENT_NAME', 'COUNT_STAR', 'SUM_TIMER_WAIT' FROM 'performance_schema'.events_waits_summary_global_by_event_name | 3.24 load | 6.15k QPS   | 527.77 μs  |
| 2 | FINDANDMODIFY lockings_id                                                                                               | 1.38 load | 1.16 QPS    | 1.19 sec   |
| 3 | BEGIN                                                                                                                   | 1.34 load | <0.01 QPS   | 0.0326     |
| 4 | AGGREGATE changelog_id,count,event,note,ime                                                                             | 0.06 load | 1.31k QPS   | 48.32 μs   |
| 5 | SELECT DISTINCT c FROM sbtest1 WHERE id BETWEEN \$1 AND \$2 ORDER BY c                                                  | 0.06 load | 0.04 QPS    | 1.36 sec   |
| 6 | SELECT c FROM sbtest1 WHERE id BETWEEN \$1 AND \$2 ORDER BY c                                                           | 0.05 load | 184.63 QPS  | 279.20 μs  |

To run Explain you need the same permissions as for executing the original query. For example, to run explain on `updates` you need update permissions.

Example: Grant the `explainRole` with update permissions.

```
db.grantPrivilegesToRole( "explainRole", [ { resource: { db: "", collection: "" }, actions: [ "update" ] } ] )
```

| # | select_type | table                                     | partitions | type | possible_keys | key  | key_len | ref  | rows | filtered | Extra |
|---|-------------|-------------------------------------------|------------|------|---------------|------|---------|------|------|----------|-------|
| 1 | SIMPLE      | events_waits_summary_global_by_event_name | NULL       | ALL  | NULL          | NULL | NULL    | NULL | 514  | 100.00   | NULL  |

**JSON**

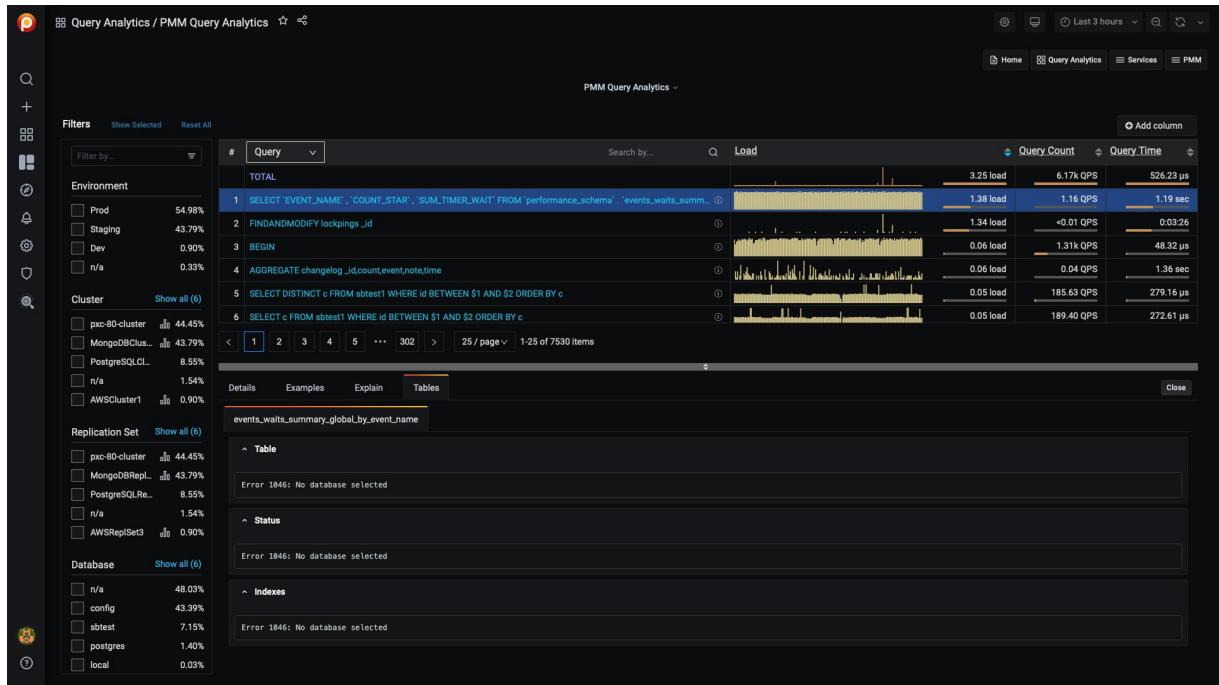
```

"root": {
  "query_blocks": [
    {
      "cost_info": {
        "query_cost": "51.65"
      }
    }
  ],
  "select_id": 1,
  "table": {
    "access_type": "ALL",
    "cost_info": {
      "data_read_per_join": "281K"
    }
  }
}
  
```

## Tables Tab

(For Query dimension.)

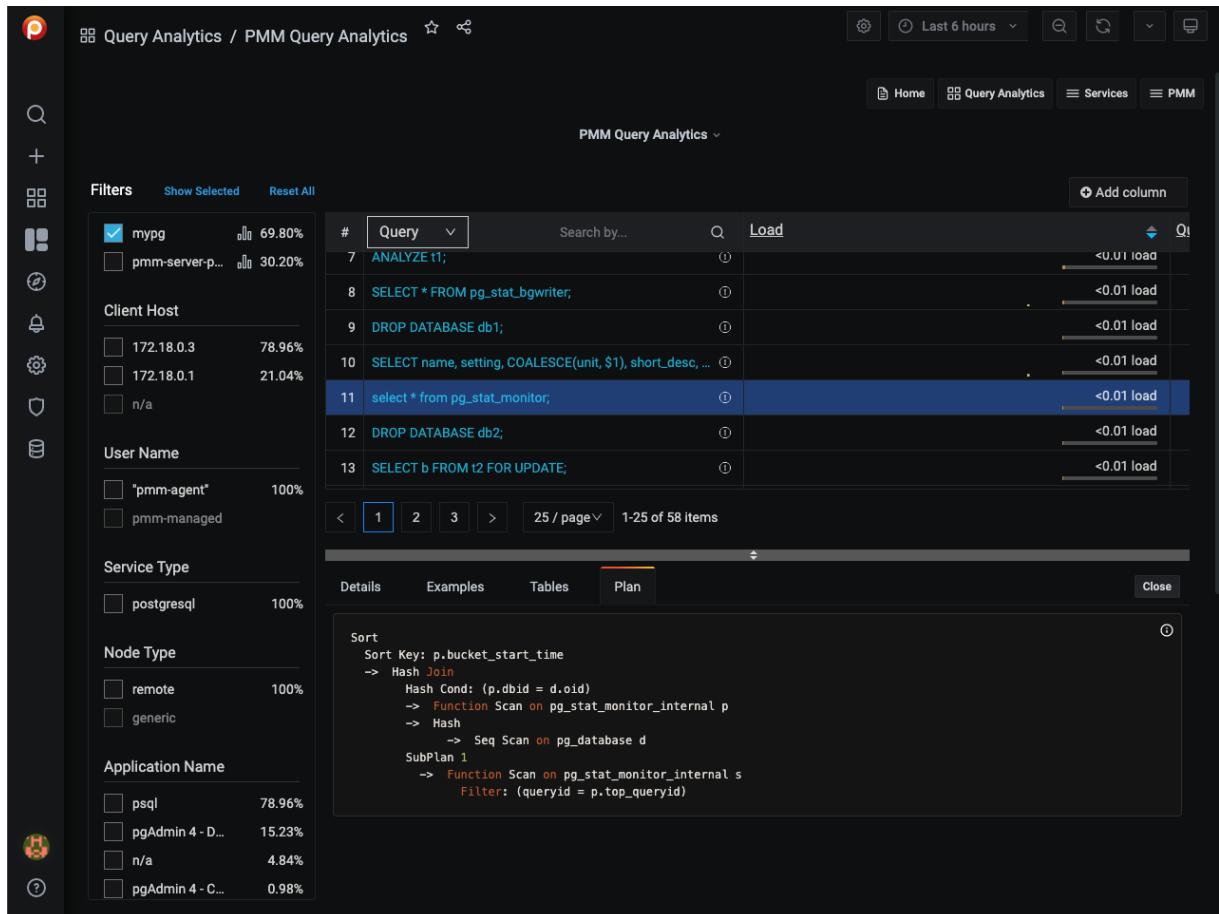
The *Tables* tab shows information on the tables and indexes involved in the selected query.



## Plan Tab

(For Query dimension.)

The *Plan* tab shows the plan for PostgreSQL queries (only available when using *pg\_stat\_monitor*).



### 3.5.4 Query Analytics for MongoDB

MongoDB is conceptually different from relational database management systems, such as MySQL and MariaDB.

Relational database management systems store data in tables that represent single entities. Complex objects are represented by linking tables.

In contrast, MongoDB uses the concept of a document where all essential information for a complex object is stored in one place.

Query Analytics can monitor MongoDB queries. Although MongoDB is not a relational database management system, you analyze its databases and collections in the same interface using the same tools.

---

Last update: 2021-11-18

## 3.6 DBaaS

### ⚠ Caution

DBaaS functionality is currently in [technical preview](#) and is subject to change.

The DBaaS dashboard is where you add, remove, and operate on Kubernetes and database clusters.

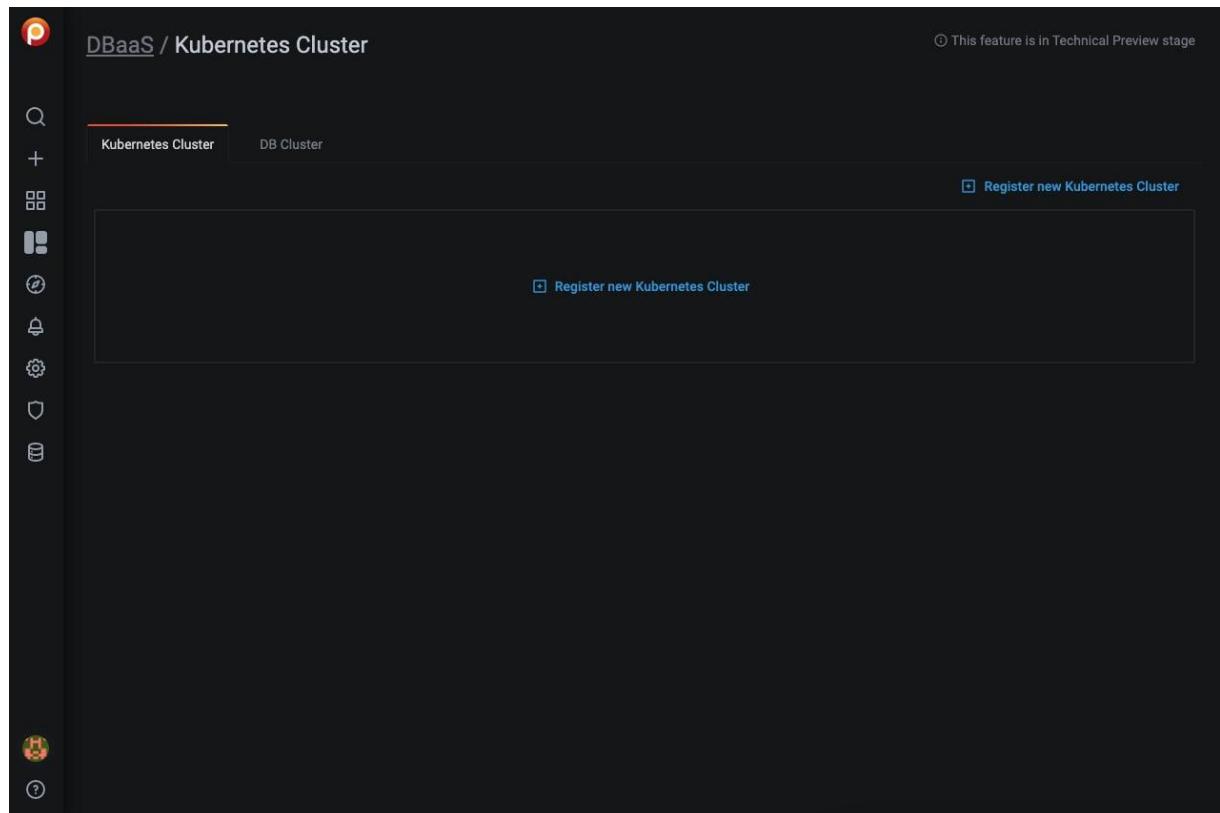
### 3.6.1 Activate DBaaS

The DBaaS feature is turned off by default. To turn it on:

1. Go to *Configuration* → *Settings* → *Advanced Settings*.
2. Click the  toggle in the *Technical preview features* section of the page.

### 3.6.2 Open the DBaaS dashboard

From the left menu, select  *DBaaS*.



This screenshot shows the DBaaS / Kubernetes Cluster dashboard. The interface is in a technical preview stage, as indicated by a note in the top right corner. The main area features two tabs: 'Kubernetes Cluster' (which is selected) and 'DB Cluster'. Below the tabs, there is a large, empty rectangular area with a placeholder message: 'Register new Kubernetes Cluster'. On the left side, there is a vertical sidebar with several icons: a magnifying glass for search, a plus sign for adding, a refresh symbol, a gear for settings, a shield for security, and a question mark for help. At the bottom of the sidebar, there are icons for Percona XtraDB Cluster and Percona Server.

### 3.6.3 Kubernetes clusters

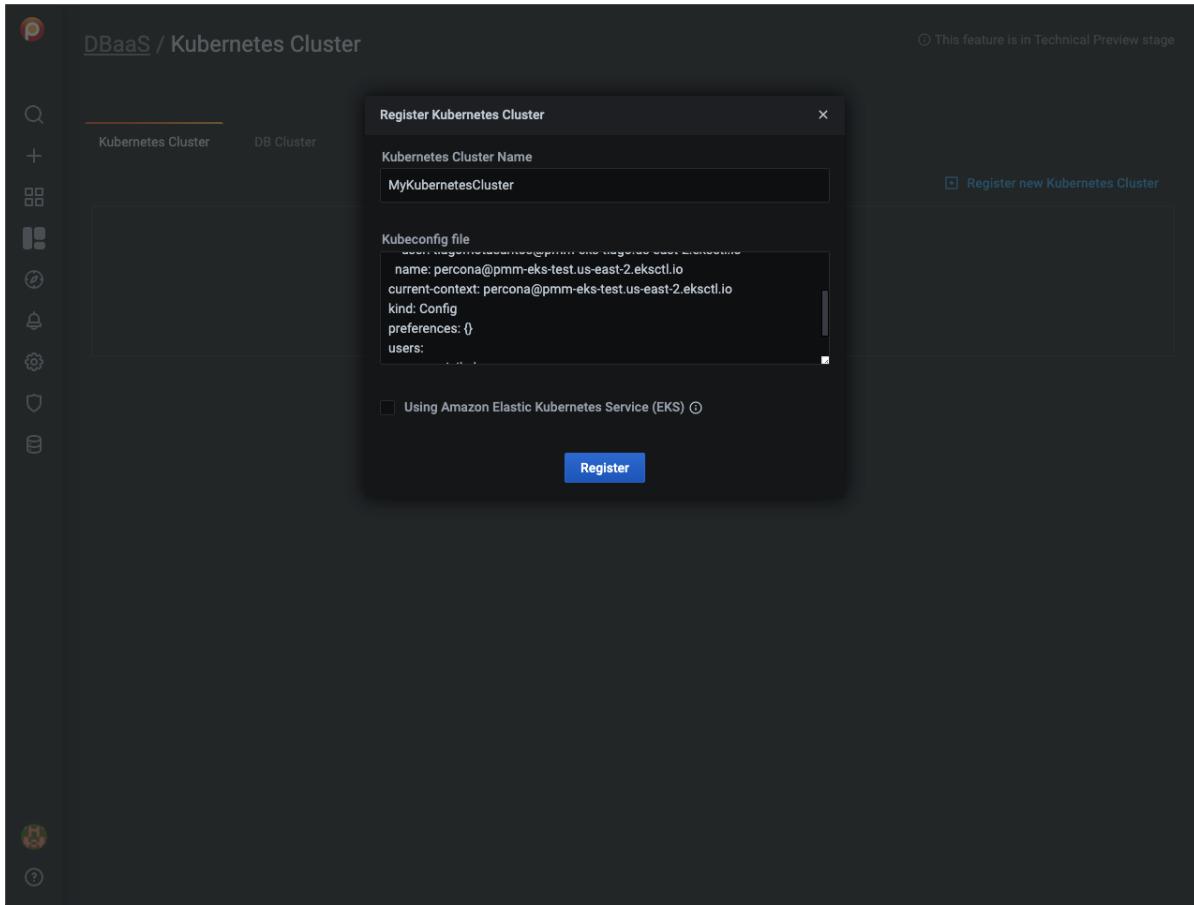
#### Add a Kubernetes cluster

PXC and PSMDB operators are installed as part of the Kubernetes cluster registration process. It enables you to deploy database clusters into the Kubernetes cluster.

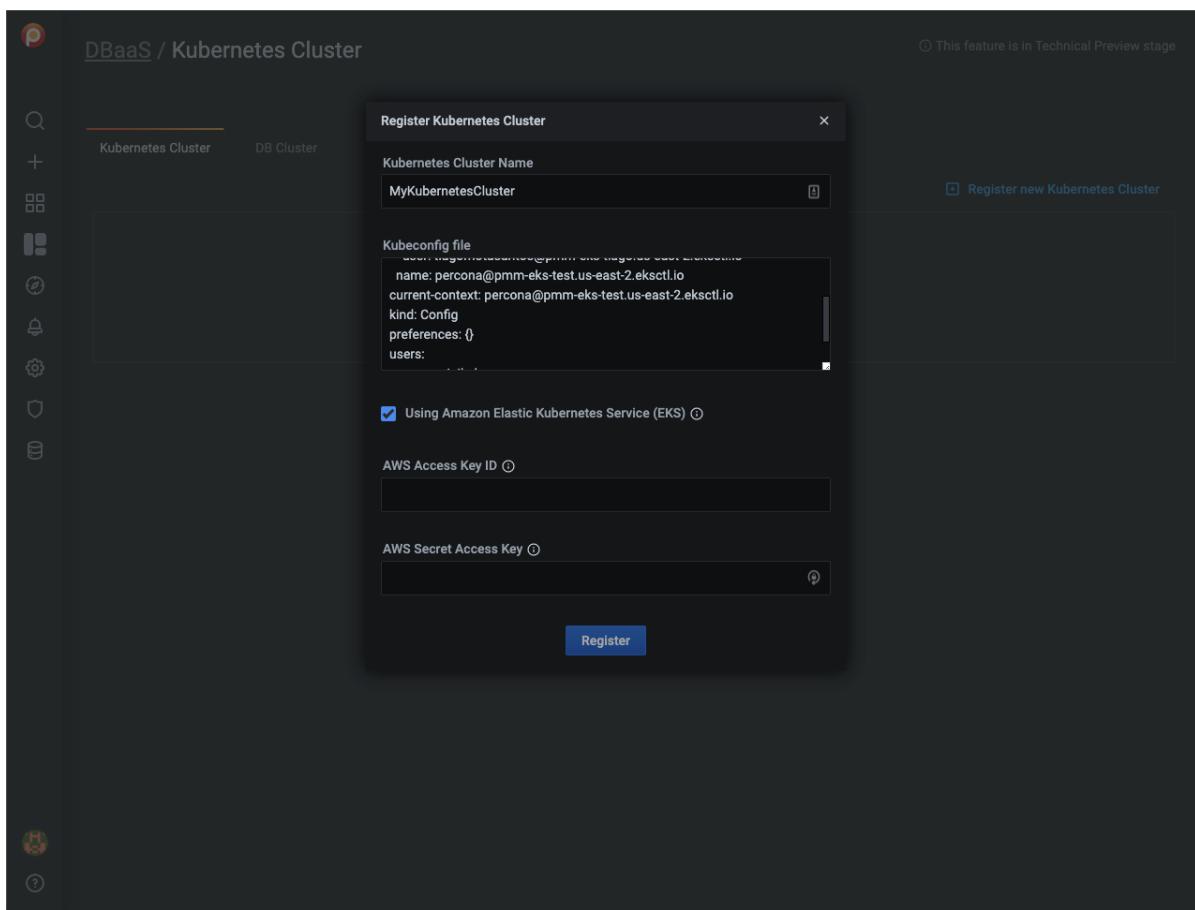
If a public address is set VM Operator is also installed as part of the Kubernetes cluster registration process. It lets you monitor a kubernetes cluster via PMM.

1. Click *Register new Kubernetes Cluster*.

2. Enter values for the *Kubernetes Cluster Name* and *Kubeconfig file* in the corresponding fields.



For a Kubernetes cluster, when using Amazon Elastic Kubernetes Service (EKS) and the *kubeconfig file* does not contain the AWS access key ID and AWS secret access key. Select the *Using Amazon Elastic Kubernetes Service (EKS)* checkbox and enter the access key ID and secret access key in the respective fields. For information on obtaining these, see the [AWS documentation](#).



3. Click *Register*.

4. A message will momentarily display telling you whether the registration was successful or not.

The screenshot shows the DBaaS / Kubernetes Cluster interface. On the left is a sidebar with various icons: a magnifying glass, a plus sign, a square, a database, a gear, a shield, and a list. The main area has a dark header with the title 'DBaaS / Kubernetes Cluster' and a note 'This feature is in Technical Preview stage'. Below the header is a green success message 'Cluster was successfully registered'. There are two tabs: 'Kubernetes Cluster' (selected) and 'DB Cluster'. A button 'Register new Kubernetes Cluster' is visible. The main content is a table with the following data:

| Kubernetes Cluster Name | Kubernetes Cluster Status | Operators                                      | Actions |
|-------------------------|---------------------------|------------------------------------------------|---------|
| minikube                | Active                    | PXC 1.8.0: Installed<br>PSMDB 1.8.0: Installed | ⋮       |

### Unregister a Kubernetes cluster

#### ⚠ Important

You can't unregister a Kubernetes cluster if there DB clusters associated with it.

1. Click *Unregister*.
2. Confirm the action by clicking *Proceed*, or abandon by clicking *Cancel*.

### View a Kubernetes cluster's configuration

1. Find the row with the Kubernetes cluster you want to see.
2. In the *Actions* column, open the ⓘ menu and click *Show configuration*.

### Manage allowed component versions

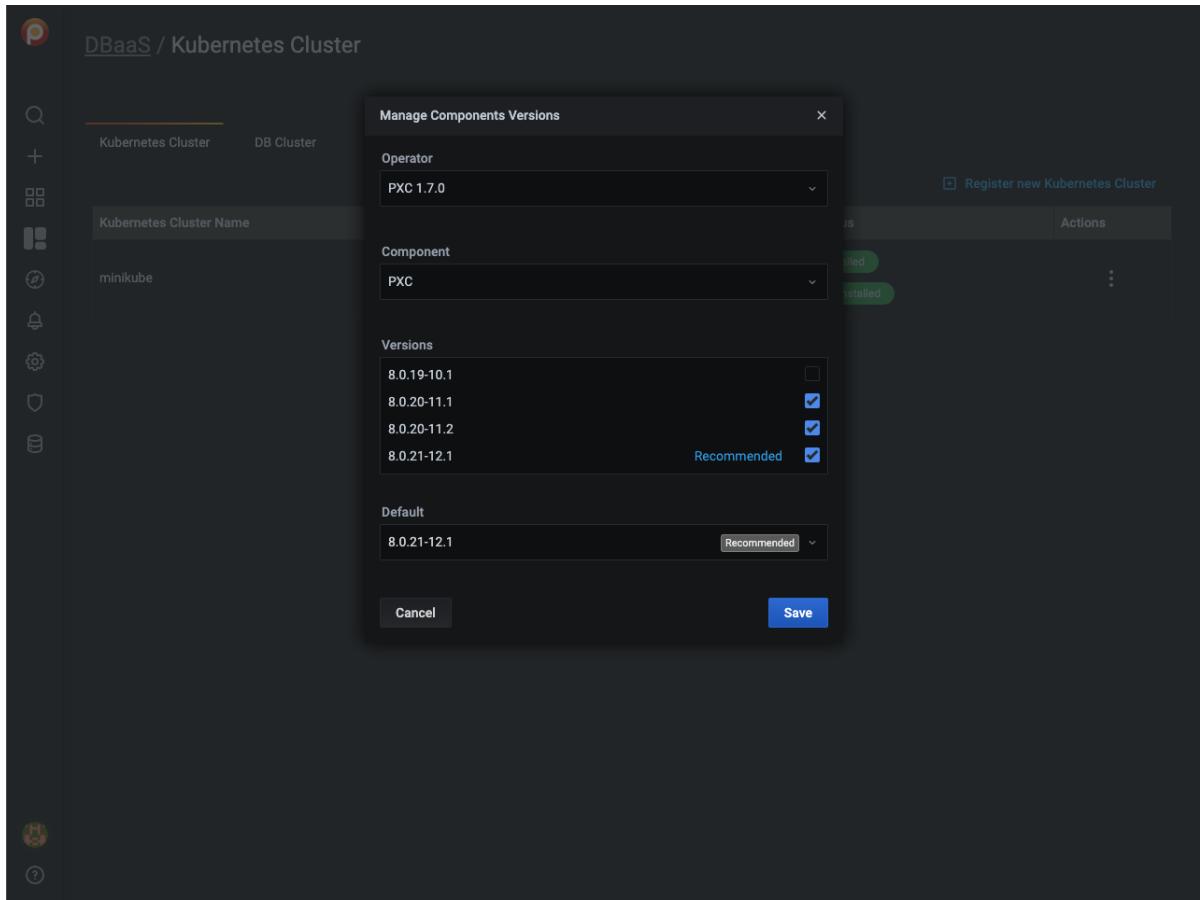
Administrators can select allowed and default versions of components versions for each cluster.

1. Find the row with the Kubernetes cluster you want to manage.
2. In the *Actions* column, open the ⚙ menu and click *Manage versions*.

The screenshot shows a dark-themed user interface for managing Kubernetes clusters. On the left is a sidebar with various icons for search, add, refresh, and configuration. The main area has a header "DBaaS / Kubernetes Cluster" with tabs for "Kubernetes Cluster" (selected) and "DB Cluster". A button "Register new Kubernetes Cluster" is in the top right. The main content is a table:

| Kubernetes Cluster Name | Kubernetes Cluster Status | Operators Status                       | Actions                                                               |
|-------------------------|---------------------------|----------------------------------------|-----------------------------------------------------------------------|
| minikube                | Active                    | MySQL: Installed<br>MongoDB: Installed | <span>⋮</span><br>Unregister<br>Show configuration<br>Manage versions |

3. Select an *Operator* and *Component* from the drop-down menus.



4. Activate or deactivate allowed versions, and select a default in the *Default* menu.

5. Click *Save*.

#### Kubernetes operator status

The Kubernetes Cluster tab shows the status of operators.

| Operators |                                           |
|-----------|-------------------------------------------|
| PXC:      | Installed                                 |
| PSMDB:    | How to install ↗                          |
| PXC:      | Installed (version 1.8.0 available) ↗     |
| PSMDB:    | Not supported (version 1.8.0 available) ↗ |

#### Kubernetes operator update

When a new version of the operator is available the *Operators* column shows a message with this information. Click the message to go to the operator release notes to find out more about the update.

DBaaS / Kubernetes Cluster

This feature is in Technical Preview stage

Kubernetes Cluster DB Cluster

Register new Kubernetes Cluster

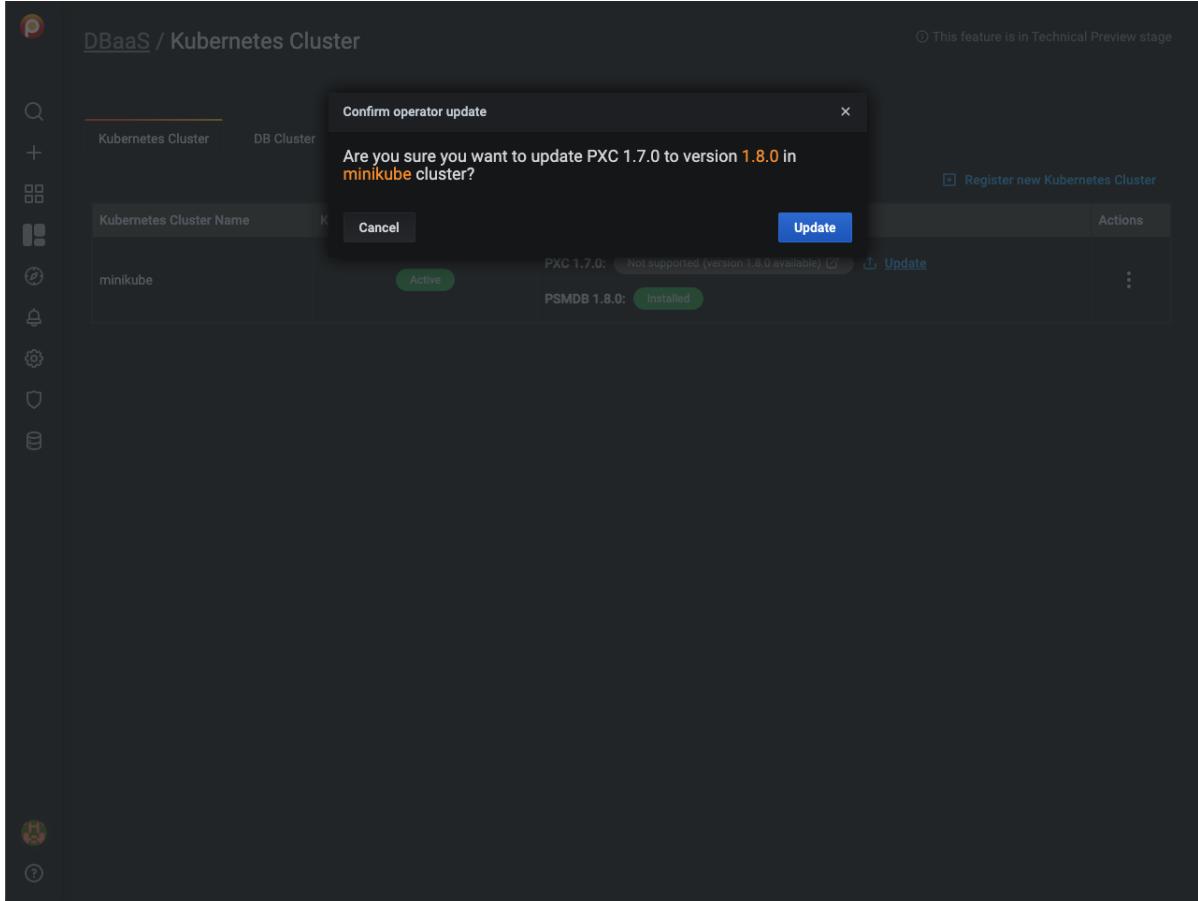
| Kubernetes Cluster Name | Kubernetes Cluster Status | Operators                                                                                           | Actions |
|-------------------------|---------------------------|-----------------------------------------------------------------------------------------------------|---------|
| minikube                | Active                    | PXC 1.7.0: Not supported (version 1.8.0 available) <a href="#">Update</a><br>PSMDB 1.8.0: Installed | ⋮       |

⋮

?

To update the cluster:

1. Find the row with the operator you want to update.
2. Click the *Update* button in front of the operator.
3. Confirm the action by clicking *Update*, or abandon by clicking *Cancel*.



### 3.6.4 DB clusters

#### Add a DB Cluster

You must create at least one Kubernetes cluster to create a DB cluster.

To monitor a DB cluster, set up a [public address](#) for PMM Server first.

1. Select the *DB Cluster* tab.

2. Click *Create DB Cluster*.

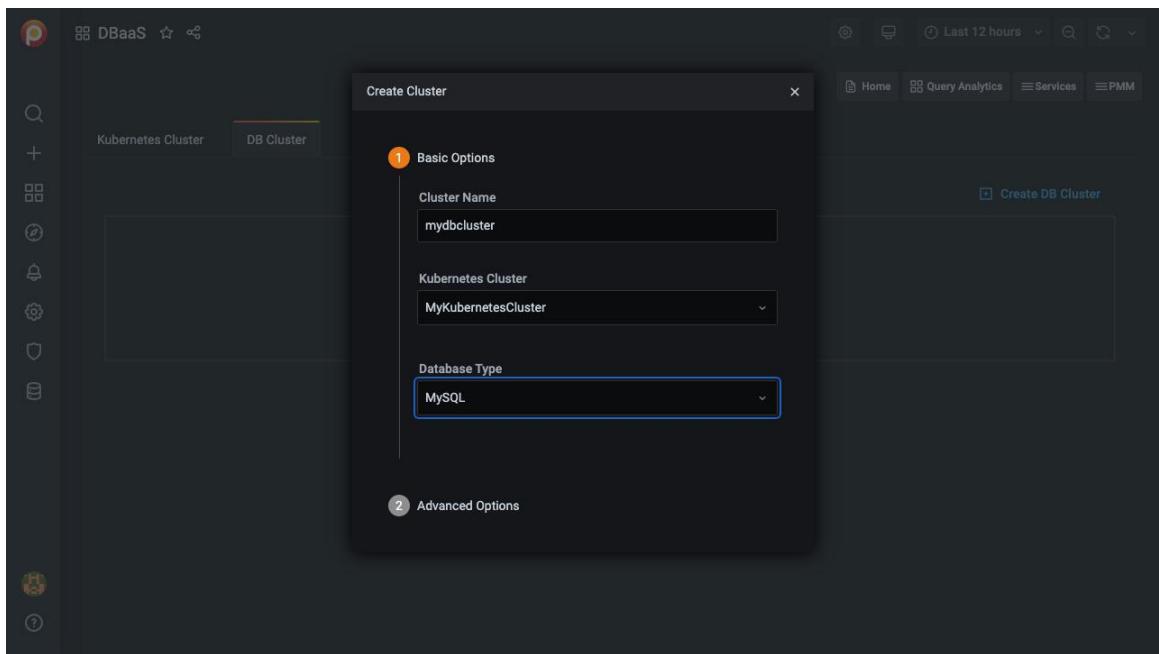
3. In section 1, *Basic Options*:

a. Enter a value for *Cluster name*. A cluster name:

- must begin with a lowercase letter;
- can comprise lowercase letters, numbers and dashes;
- must end with an alphanumeric character.

b. Select a cluster from the *Kubernetes Cluster* menu.

c. Select a database type from the *Database Type* menu.



4. Expand section 2, *Advanced Options*.

- Select *Topology*, either *Cluster* or *Single Node*.
- Select the number of nodes. (The lower limit is 3.)
- Select *External Access* if you want to make your DB cluster available outside of Kubernetes cluster.

By default, only internal access is provided.

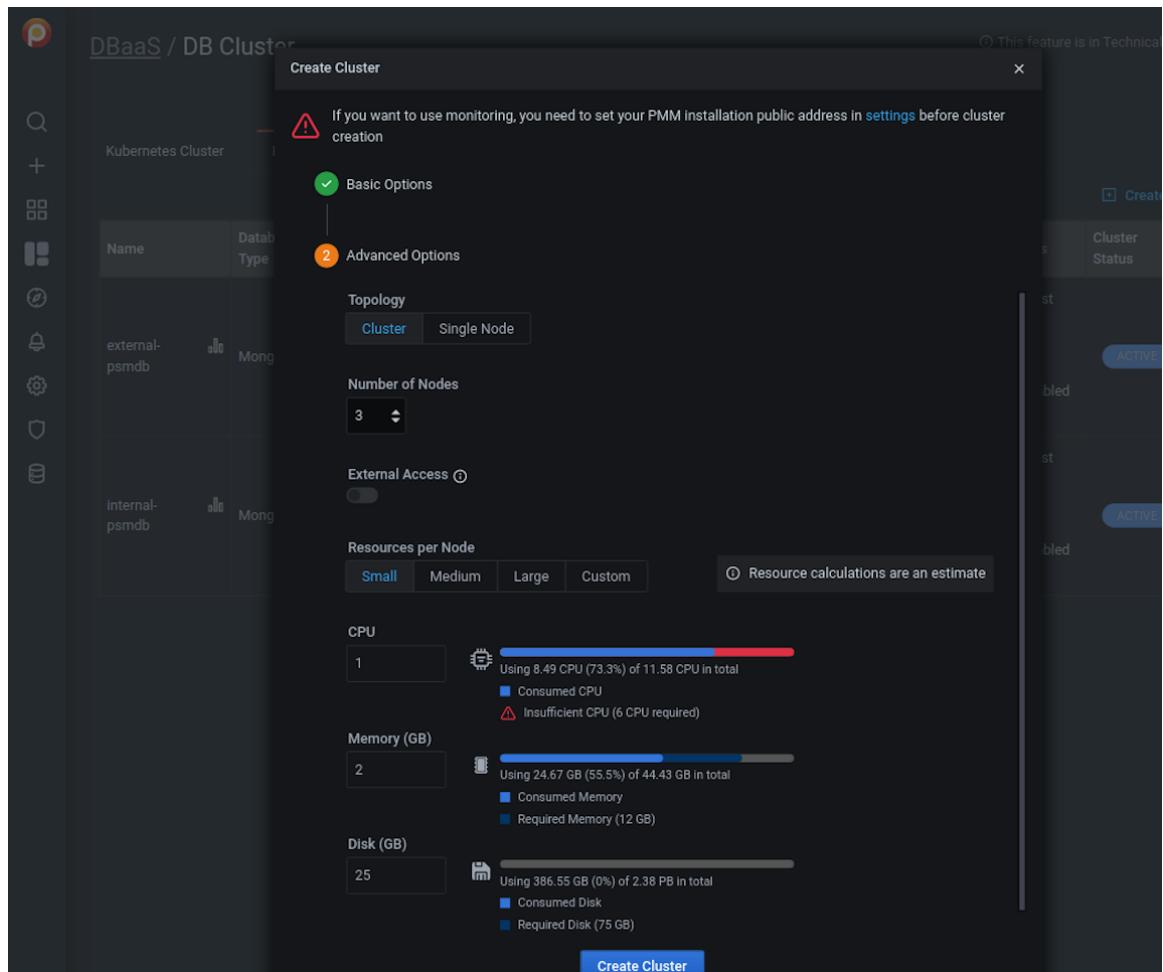
*External Access* can't be granted for local Kubernetes clusters (e.g. minikube).

- Select a preset for *Resources per Node*.

*Small*, *Medium* and *Large* are fixed preset values for *Memory*, *CPU*, and *Disk*.

Values for the *Custom* preset can be edited.

Beside each resource type is an estimate of the required and available resources represented numerically in absolute and percentage values, and graphically as a colored, segmented bar showing the projected ratio of used to available resources. A red warning triangle is shown if the requested resources exceed those available.



5. When both *Basic Options* and *Advanced Options* section icons are green, the *Create Cluster* button becomes active. (If inactive, check the values for fields in sections whose icon is red.)

Click *Create Cluster* to create your cluster.

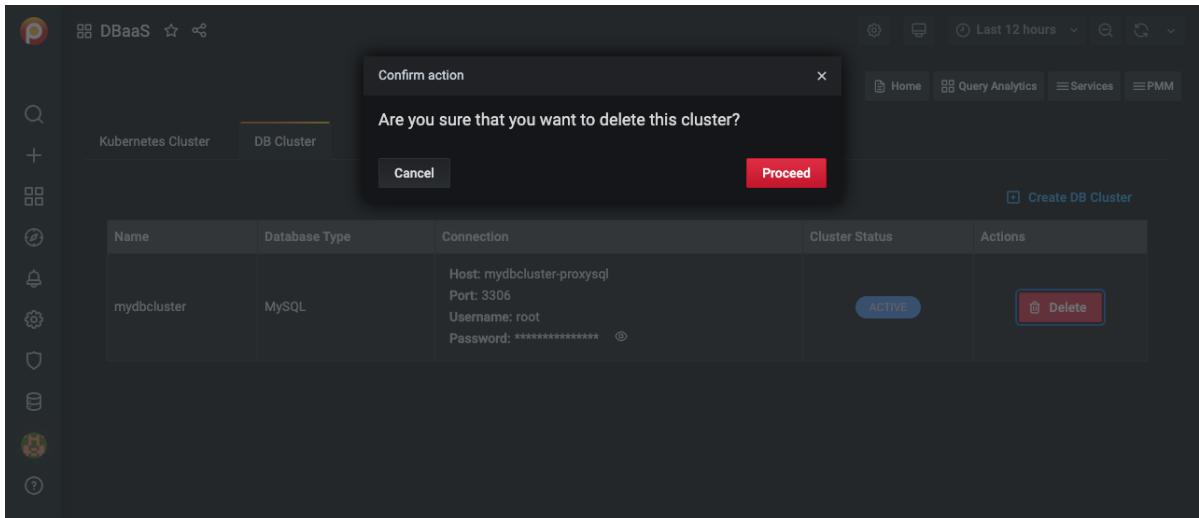
6. A row appears with information on your cluster:

| Name          | Database     | Connection                                                                             | DB Cluster Parameters                                                                           | Cluster Status | Actions                                                                                                                                                     |
|---------------|--------------|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mysqlcluster2 | MySQL 8.0.22 | Host: mysqlcluster2-haproxy.default<br>Port: 3306<br>Username: root<br>Password: ***** | K8s cluster name: minikube<br>CPU: 1<br>Memory: 1 GB<br>Disk: 1 GB<br>External Access: Disabled | ACTIVE         | <a href="#">Update</a><br><a href="#">Delete</a><br><a href="#">Edit</a><br><a href="#">Restart</a><br><a href="#">Suspend</a><br><a href="#">View logs</a> |

- **Name:** The cluster name.
- **Database:** The cluster database type and version.
- **Connection:**
  - **Host:** The hostname.
  - **Port:** The port number.
  - **Username:** The connection username.
  - **Password:** The connection password (click the eye icon to reveal).
- **DB Cluster Parameters:**
  - **K8s cluster name:** The Kubernetes cluster name.
  - **CPU:** The number of CPUs allocated to the cluster.
  - **Memory:** The amount of memory allocated to the cluster.
  - **Disk:** The amount of disk space allocated to the cluster.
- **Cluster Status:**
  - **PENDING:** The cluster is being created.
  - **ACTIVE:** The cluster is active.
  - **FAILED:** The cluster could not be created.
  - **DELETING:** The cluster is being deleted.
  - **UPDATING:** The cluster is being updated.

### Delete a DB Cluster

1. Find the row with the database cluster you want to delete.
2. In the *Actions* column, open the ⓘ menu and click *Delete*.
3. Confirm the action by clicking *Proceed*, or abandon by clicking *Cancel*.



### ⚡ Danger

Deleting a cluster in this way also deletes any attached volumes.

#### Edit a DB Cluster

1. Select the *DB Cluster* tab.
  2. Find the row with the database cluster you want to change.
  3. In the *Actions* column, open the *⋮* menu and click *Edit*.
- A paused cluster can't be edited.

#### Restart a DB Cluster

1. Select the *DB Cluster* tab.
2. Identify the database cluster to be changed.
3. In the *Actions* column, open the *⋮* menu and click *Restart*.

#### Suspend or resume a DB Cluster

1. Select the *DB Cluster* tab.
2. Identify the DB cluster to suspend or resume.

3. In the *Actions* column, open the ⓘ menu and click the required action:

- For active clusters, click *Suspend*.

| Name    | Database Type | Connection                                                                                             | DB Cluster Parameters                                             | Cluster Status | Actions        |
|---------|---------------|--------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|----------------|----------------|
| mongodb | MongoDB       | Host: mongodb-rs0.default.svc.cluster.local<br>Port: 27017<br>Username: userAdmin<br>Password: ***** ⓘ | K8s cluster name: mongodb<br>CPU: 1<br>Memory: 2 GB<br>Disk: 0 GB | ACTIVE         | <span>⋮</span> |

- For paused clusters, click *Resume*.

| Name    | Database Type | Connection | DB Cluster Parameters | Cluster Status | Actions        |
|---------|---------------|------------|-----------------------|----------------|----------------|
| mongodb | MongoDB       | -          | -                     | PAUSED         | <span>⋮</span> |

### Update a DB Cluster

1. Select the *DB Cluster* tab.
2. Identify the DB cluster to update.
3. In the *Actions* column, open the ⓘ menu and click *Update*:

Kubernetes Cluster    DB Cluster

Create DB Cluster

| Name    | Database     | Connection                                                                       | DB Cluster Parameters                                                                           | Cluster Status | Actions                                                                                                                                                                                                                                               |
|---------|--------------|----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cluster | MySQL 8.0.21 | Host: cluster-haproxy.default<br>Port: 3306<br>Username: root<br>Password: ***** | K8s cluster name: minikube<br>CPU: 1<br>Memory: 1 GB<br>Disk: 1 GB<br>External Access: Disabled | ACTIVE         | <input type="button" value="Update"/><br><input type="button" value="Delete"/><br><input type="button" value="Edit"/><br><input type="button" value="Restart"/><br><input type="button" value="Suspend"/><br><input type="button" value="View logs"/> |

4. Confirm the update by clicking on *Update*, or abadon by clicking *Cancel*.

DBaaS / DB Cluster

This feature is in Technical Preview stage

Create DB Cluster

| Name    | Database     | Conn                                                                             | Actions                                                                                         |
|---------|--------------|----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| cluster | MySQL 8.0.21 | Host: cluster-haproxy.default<br>Port: 3306<br>Username: root<br>Password: ***** | K8s cluster name: minikube<br>CPU: 1<br>Memory: 1 GB<br>Disk: 1 GB<br>External Access: Disabled |

Confirm database update

Are you sure you want to update MySQL 8.0.21-12.1 to version 8.0.22-13.1 in cluster cluster?

### See also

[Setting up a development environment for DBaaS](#)

Last update: 2022-01-19

## 3.7 Security Threat Tool

The Security Threat Tool runs regular checks against connected databases, alerting you if any servers pose a potential security threat.

### Anonymous and registered checks

All checks are hosted on Percona Platform. PMM Server automatically downloads them from here when the Security Threat Tool is enabled in PMM.

By default, PMM has access to a set of anonymous checks, which can be downloaded even if PMM is not connected to Percona Platform. As soon as you connect your PMM instance to Percona Platform, you get additional access to registered checks, which offer more advanced database health information.

To see the complete list of available checks, see [Security Checks for PMM](#) topic in the Percona Platform documentation.

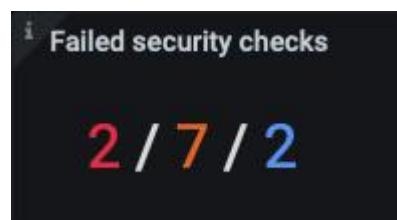
### 3.7.1 How to enable

By default, the Security Threat Tool (STT) is disabled. To enable it, select *Configuration* → *Settings* → *Advanced Settings*. ([Read more](#)).

Enabling STT in the settings causes the PMM server to download STT checks from Percona Platform and run them once. This operation runs in the background, so even though the settings update finishes instantly, it might take some time for the checks to complete download and execution and the results (if any) to be visible in the *PMM Database Checks* dashboard.

### 3.7.2 Checks results

The results are sent to PMM Server where you can review any failed checks on the **Home Dashboard > Failed security checks** panel. The summary count of failed checks is classified as **Critical**, **Major** and **Trivial**:



To see more details about the available checks and any checks that failed, click the *Security Checks* on the main menu. This icon is only available if you have enabled the Security Threat Tool.

**Check results data always remains on the PMM Server.** It is not related to anonymous data sent for Telemetry purposes.

### 3.7.3 Change a check's interval

The checks can be executed manually or automatically. By default, PMM runs automatic checks every 24 hours. To configure this interval:

1. Click *Security Checks*.
  2. Select the *All Checks* tab.
  3. In the *Actions* column for a chosen check, click the *Interval* icon.
  4. Choose an interval: *Standard, Rare, Frequent*.
  5. Click *Save*.
- 

Last update: 2022-02-08

## 4. How to

### 4.1 How to

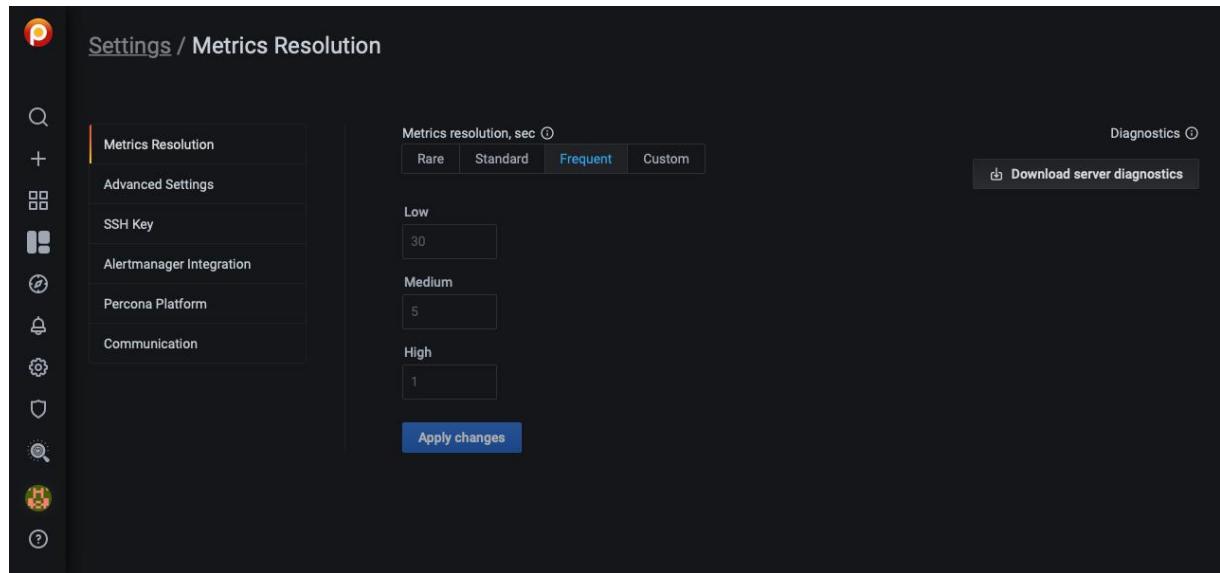
- [Configure](#) via the PMM Settings page.
  - [Manage users](#) via the PMM Users page.
  - [Upgrade PMM](#) Server via the user interface.
  - [Secure](#) your PMM installation.
  - [Optimize](#) the performance of your PMM installation.
  - [Annotate](#) charts to mark significant events.
  - [Render dashboard images](#) to save or share.
  - [Extend Metrics](#) with textfile collector.
  - [Troubleshoot](#)
- 

Last update: 2022-01-18

## 4.2 Configure

The *Settings* page is where you configure PMM.

Open the *Settings* page from the main menu with *Configuration* → *Settings*. The page opens with the *Metrics Resolution* settings tab selected.



On the left are the selector tabs:

- Configure
  - Metrics resolution
  - Advanced Settings
    - Data Retention
    - Telemetry
    - Check for updates
    - Security Threat Tool
  - Public address
    - DBaaS
    - Integrated Alerting
    - Microsoft Azure Monitoring
    - Public Address
  - SSH Key
  - Alertmanager integration
  - Percona Platform
    - Login
    - Sign up
    - Password Reset
      - Password Forgotten
      - Change Password after Login
  - Communication
    - Email
    - Slack



### Tip

Click *Apply changes* to save any changes made here.

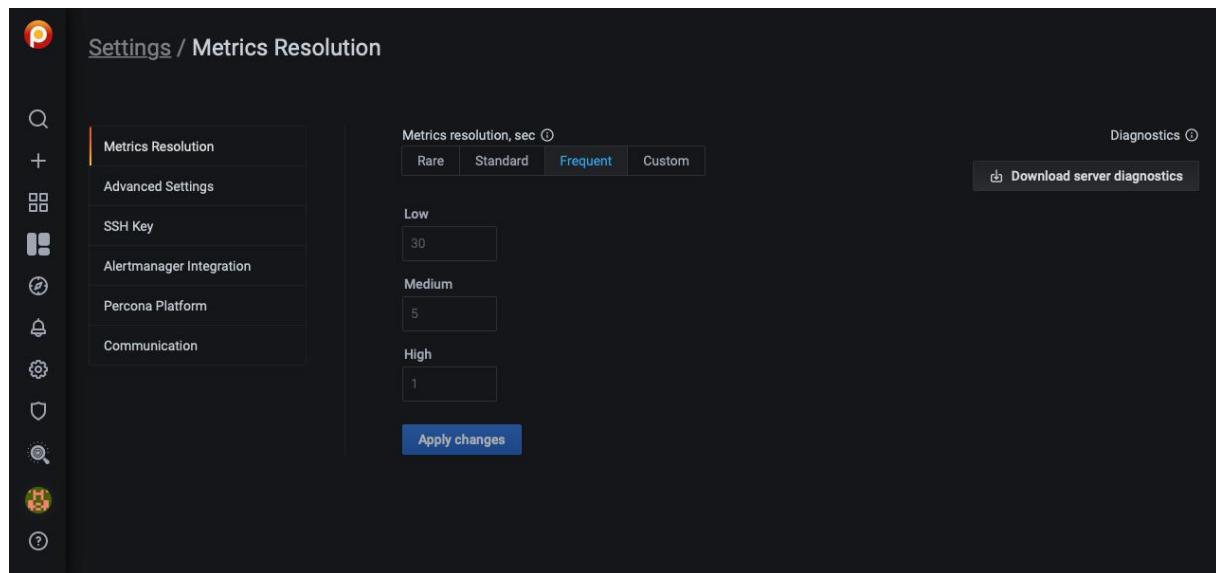
## 4.2.1 Diagnostics

On all tabs is a *Diagnostics* section (top-right). Click *Download server diagnostics* to retrieve PMM diagnostics data which can be examined and/or shared with our support team should you need help.

## 4.2.2 Metrics resolution

Metrics are collected at three intervals representing low, medium and high resolutions.

The *Metrics Resolution* settings tab contains a radio button with three fixed presets (*Rare*, *Standard* and *Frequent*) and one editable custom preset (*Custom*).



Each preset is a group of low, medium and high resolutions. The values are in seconds.

### Time intervals and resolutions

Short time intervals are *high* resolution metrics. Longer time intervals are *low* resolution. So:

- A low resolution interval *increases* the time between collection, resulting in low-resolution metrics and lower disk usage.
- A high resolution interval *decreases* the time between collection, resulting in high-resolution metrics and higher disk usage.

The default values (in seconds) for the fixed presets and their resolution names are:

| Editable? | Preset            | Low | Medium | High |
|-----------|-------------------|-----|--------|------|
| No        | Rare              | 300 | 180    | 60   |
| No        | Standard          | 60  | 10     | 5    |
| No        | Frequent          | 30  | 5      | 1    |
| Yes       | Custom (defaults) | 60  | 10     | 5    |

Values for the *Custom* preset can be entered as values, or changed with the arrows.

If there is poor network connectivity between PMM Server and PMM Client, or between PMM Client and the database server being monitored, scraping every second may not be possible when the network latency is greater than 1 second.

## 4.2.3 Advanced Settings

## Data Retention

*Data retention* specifies how long data is stored by PMM Server. By default, time-series data is stored for 30 days. You can adjust the data retention time to balance your system's available disk space with your metrics history requirements.

## Telemetry

The *Telemetry* switch enables gathering and sending basic **anonymous** data to Percona, which helps us to determine where to focus the development and what is the uptake for each release of PMM. Specifically, gathering this information helps determine if we need to release patches to legacy versions beyond support, determining when supporting a particular version is no longer necessary, and even understanding how the frequency of release encourages or deters adoption.

The following information is gathered:

- PMM Version;
- Installation Method (Docker, AMI, OVF);
- the Server Uptime;
- Security Threat Tool Status (enabled or disabled);
- Integrated Alerting Status (enabled or disabled).

We do not gather anything that identify a system, but the following two points should be mentioned:

1. The Country Code is evaluated from the submitting IP address before being discarded.
2. We do create an “instance ID” - a random string generated using UUID v4. This instance ID is generated to distinguish new instances from existing ones, for figuring out instance upgrades.

The first telemetry reporting of a new PMM Server instance is delayed by 24 hours to allow enough time to disable the service for those that do not wish to share any information.

The landing page for this service, [check.percona.com](https://check.percona.com), explains what this service is.

Grafana's **anonymous usage statistics** is not managed by PMM. To activate it, you must change the PMM Server container configuration after each update.

As well as via the *PMM Settings* page, you can also disable telemetry with the `-e DISABLE_TELEMETRY=1` option in your docker run statement for the PMM Server.

Telemetry is sent straight away; the 24 hour grace period is not honored.

### Check for updates

When active, PMM will automatically check for updates and put a notification in the home page *Updates* dashboard if any are available.

### Security Threat Tool

The **Security Threat Tool** performs a range of security-related checks on a registered instance and reports the findings. It is off by default.



#### Tip

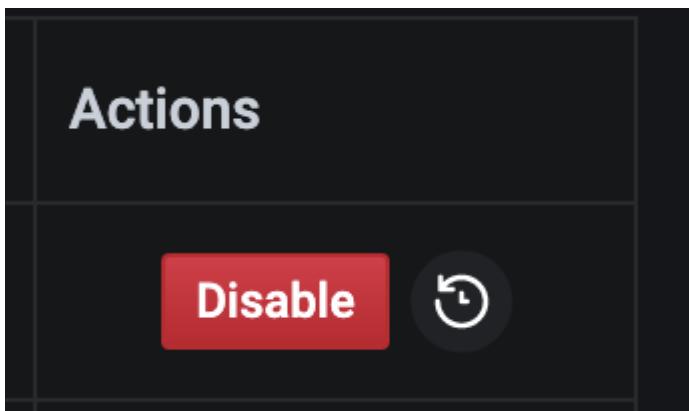
To see the results of checks, select *Security Checks* to open the *Security Checks/Failed Checks* dashboard, and select the *Failed Checks* tab.

Checks are re-fetched and re-run at intervals. There are three customizable check intervals:

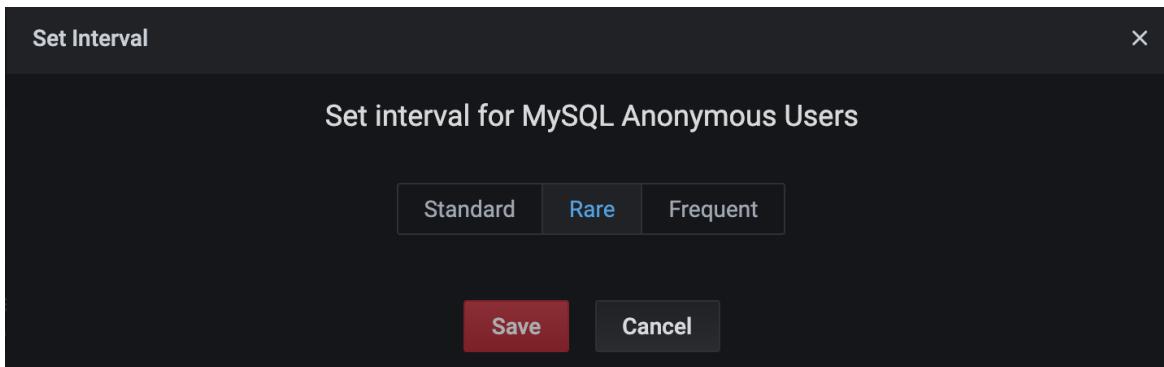
| Interval name                      | Value (hours) |
|------------------------------------|---------------|
| <i>Rare interval</i>               | 78            |
| <i>Standard interval</i> (default) | 24            |
| <i>Frequent interval</i>           | 4             |

Checks use the *Standard* interval by default. To change a check's interval:

- Go to  [PMM Database Checks](#).
- Select [All Checks](#).
- In the *Actions* column, select the  icon.



- Select an interval and click *Save*.



#### 4.2.4 Public address

The address or hostname PMM Server will be accessible at. Click *Get from browser* to have your browser detect and populate this field automatically.

##### DBaaS

 **Caution**

DBaaS functionality is a technical preview that must be turned on with a server feature flag. See [DBaaS](#).

Enables/disables [DBaaS features](#) on this server.

 **Important**

Deactivating DBaaS **does not** suspend or remove running DB clusters.

##### Integrated Alerting

Enables [Integrated Alerting](#) and reveals the [Communication](#) tab.

## Microsoft Azure Monitoring

### ⚠ Caution

This is a technical preview feature.

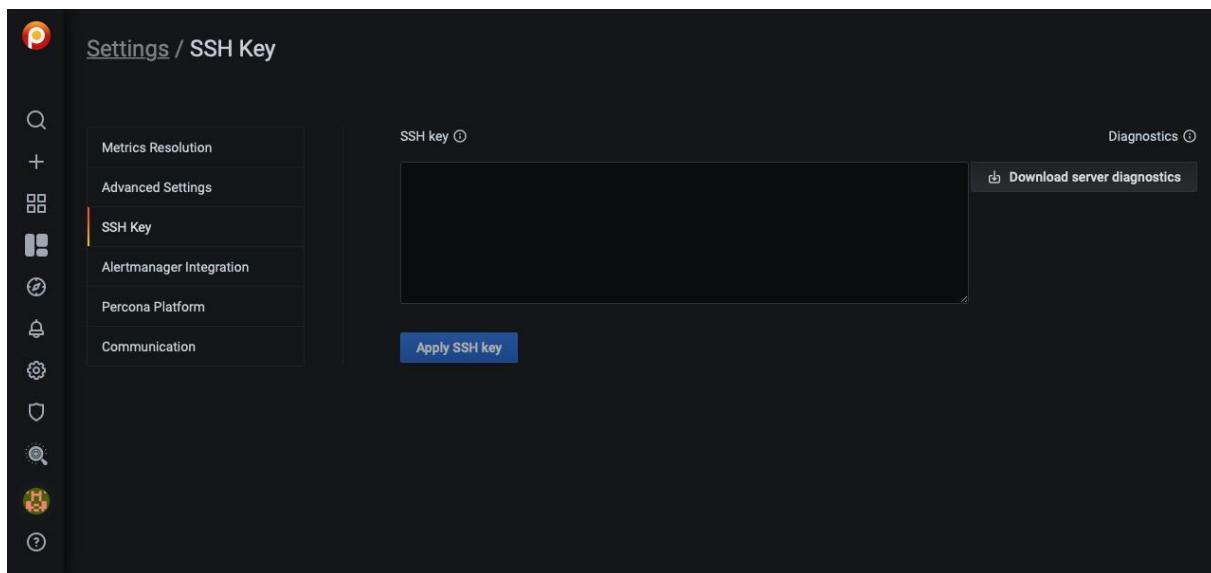
Activates Microsoft Azure monitoring.

### Public Address

Public address for accessing DBaaS features on this server.

## 4.2.5 SSH Key

This section lets you upload your public SSH key to access the PMM Server via SSH (for example, when accessing PMM Server as a [virtual appliance](#)).



Enter your [public key](#) in the *SSH Key* field and click *Apply SSH Key*.

## 4.2.6 Alertmanager integration

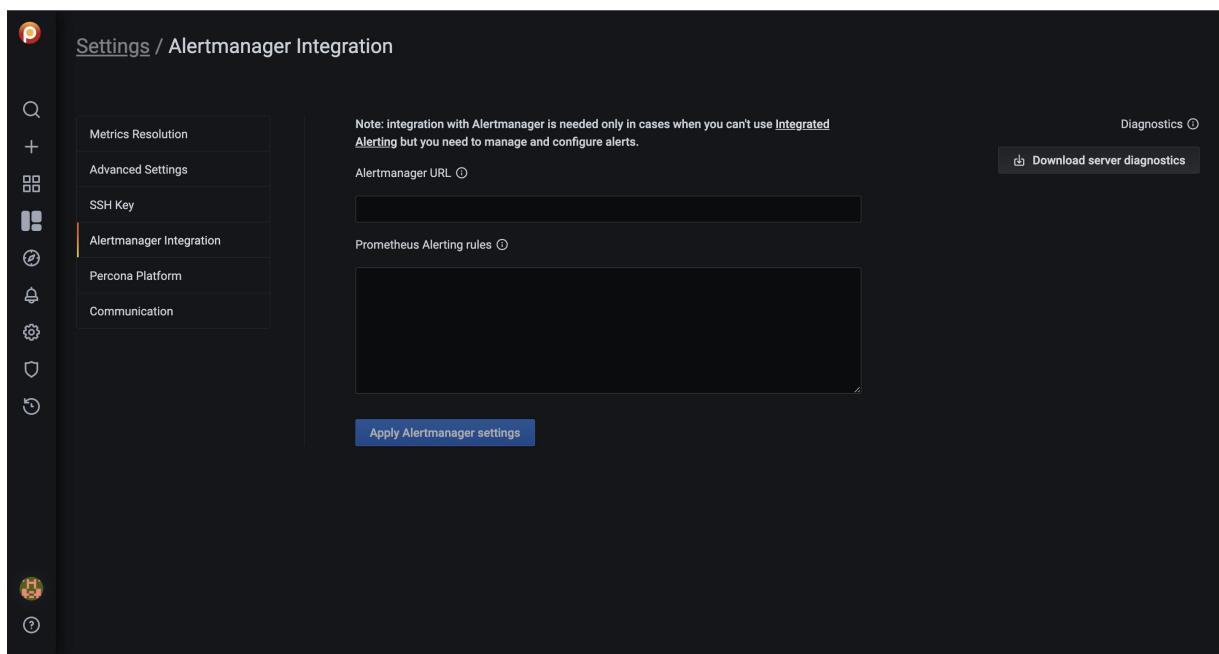
Alertmanager manages alerts, de-duplicating, grouping, and routing them to the appropriate receiver or display component.

This section lets you configure integration of VictoriaMetrics with an external Alertmanager.

### 💡 Tip

If possible, use [Integrated Alerting](#) instead of Alertmanager.

- The *Alertmanager URL* field should contain the URL of the Alertmanager which would serve your PMM alerts.
- The *Prometheus Alerting rules* field is used to specify alerting rules in the YAML configuration format.



Fill both fields and click the *Apply Alertmanager settings* button to proceed.

#### 4.2.7 Percona Platform

This panel is where you connect your PMM server to your Percona Platform Account.

Your Percona Platform Account is separate from your PMM User account.

##### Connect PMM to Percona Platform

**Disclaimer:** Percona Platform is a Preview release and several functionalities are still under development or subject to change. As such, we recommend connecting PMM to Percona Platform only for testing purposes.

Connect your PMM servers to Percona Platform to leverage Platform services that boost the monitoring capabilities of your PMM installations.

To connect to Percona Platform, you need a Percona Account. With a Percona Account you also get access to various Percona services, including Percona Platform, Percona Customer Portal, and Community Forum.

If you don't have a Percona Account, you can create one on the [Percona Account Sign Up](#) page.

##### PRE-REQUISITES

To ensure that PMM can establish a connection to Percona Platform:

##### Check that you are a member of an existing Platform organization

1. Log in to [Percona Platform](#) using your Percona Account.
2. On the *Getting Started* page, check that the *Create organization* step shows an option to view your organization.

If you see an option to create a new organization instead, your Percona Account is not linked to any organization yet. If this is the case, contact your account administrator, or create a new organization for your Percona Account.

### Set the public address of your PMM server

1. In PMM, go to *Settings > Advanced Settings*.
2. Enter your address/hostname or click *Get from browser* to enable your browser to automatically detect and populate this field.
3. Save the changes.

### 4.2.8 Connect PMM to Percona Platform

To connect your PMM server to Percona Platform:

1. In PMM go to *Settings > Percona Platform* tab:

The screenshot shows the PMM Settings interface. On the left is a sidebar with various icons: a red 'P' for Metrics resolution, a plus sign for Advanced settings, a square for SSH key, a gear for AlertManager integration, a shield for Percona Platform (which is highlighted with an orange border), and a bell for Communication. The main area has a header 'PMM > Settings'. To the right, under 'Connect PMM to Percona Portal', there are three input fields: 'PMM Server Name \*' (empty), 'Percona Account (email) \*' (empty), and 'Password \*' (empty). Below these is a blue 'Connect' button and a 'Read more' link at the bottom.

2. Fill in the *Connect PMM to Percona Portal* form with the name of your PMM instance and the credentials of your Percona Account:
3. Click *Connect*. To confirm that you have successfully connected the server and check the list of all servers currently connected to an organization, go to [Percona Platform > Dashboard](#) tab and click **View Instances** next to the **Connect your PMM** step.

### Disconnect a PMM instance

Disconnect a PMM instance when you want to unlink it from your Percona Platform organization or stop monitoring it there.

To disconnect a PMM server, go to > [Configuration > Settings > Percona Platform](#) and click **Disconnect**.

To confirm that the server disconnected successfully, go to Percona Platform > **Dashboard** tab > **View instances**. This displays the **PMM instances** page where you can check the list of servers currently connected to your Platform organization.

## 4.2.9 Sign into PMM with your Percona Account

Once you've successfully connected your PMM instance to Percona Platform, you can also sign into PMM using your Percona Account:

1. Log out of your existing PMM session.
2. On the PMM login screen, click *Sign in with Percona Account*. If you have an active Percona Account session on the same browser, PMM will log you in automatically. Otherwise, enter your Percona Account credentials to start a new session.

### Password Reset

#### PASSWORD FORGOTTEN

In case you forgot your password, click on the *Forgot password* link on the login page.

You will be redirected to a password reset page. Enter the email you are registered with in the field and click on *Reset via Email*.

The screenshot shows a password reset form. At the top is the Percona logo. Below it is a heading 'Reset Password'. Underneath is a label 'Email or Username' followed by a text input field. At the bottom of the form is a large blue button labeled 'Reset via Email'. Below the button is a link 'Back to Sign In'.

An email with a link to reset your password will be sent to you.

#### CHANGE PASSWORD AFTER LOGIN

If you did not forget your password but you still want to change it, go to <https://okta.percona.com/enduser/settings> (make sure you are logged in).

The screenshot shows the Percona account settings interface. At the top right, there are links for 'Home', 'John', and a notification bell. The main area has three tabs: 'Personal Information' (selected), 'Change Password', and 'Display Language'. The 'Personal Information' tab shows fields for First name (John), Last name (Smith), Okta username (john.smith@example.net), Primary email (john.smith@example.net), Secondary email, and Mobile phone. The 'Change Password' tab displays password requirements: At least 10 characters, A lowercase letter, An uppercase letter, A number, No parts of your username, Does not include your first name, Does not include your last name, and Your password cannot be any of your last 4 passwords. It also contains input fields for Current password, New password, and Confirm new password, with a 'Change Password' button. The 'Display Language' tab shows the Language set to English, with a note about it being automatically set by the browser.

Insert you current password and the new password in the form to the bottom right of the page. If you cannot see the form, you will need to click on the *Edit Profile* green button (you will be prompted for you password).

Click on *Change Password*. If everything goes well, you will see a confirmation message.

#### 4.2.10 Communication

Global communications settings for [Integrated Alerting](#).

##### Tip

If there is no *Communication* tab, go to the *Advanced Settings* tab and activate *Integrated Alerting*.

The screenshot shows the 'Communication' settings page in the PMM interface. The left sidebar has a 'Communication' section selected. The main area shows fields for Email (Server Address, Hello, From), Slack (disabled), and Auth Type (None selected). A 'Diagnostics' button is also present.

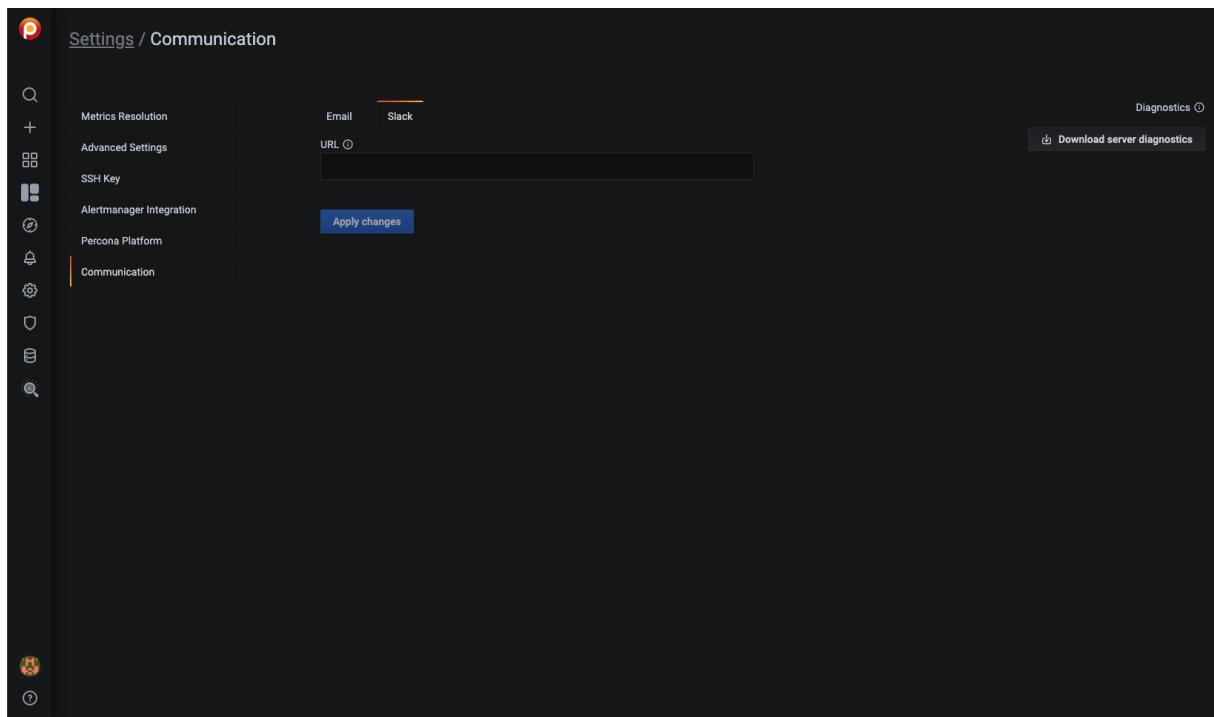
(Integrated Alerting uses a separate instance of Alertmanager run by `pmm-managed`.)

## Email

Settings for the SMTP email server:

- **Server Address:** The default SMTP smarthost used for sending emails, including port number.
- **Hello:** The default hostname to identify to the SMTP server.
- **From:** The sender's email address.
- **Auth type:** Authentication type. Choose from:
  - *None*
  - *Plain*
  - *Login*
  - *CRAM-MD5*
- **Username:** SMTP Auth using CRAM-MD5, LOGIN and PLAIN.
- **Password:** SMTP Auth using CRAM-MD5, LOGIN and PLAIN. After setting up the SMTP server, click *Test* to confirm that you can connect to the server with the current settings.

## Slack



Settings for Slack notifications:

- *URL*: The Slack webhook URL to use for Slack notifications.

#### See also

[Prometheus Alertmanager configuration](#)

---

Last update: 2022-02-08

## 4.3 Manage users

This topic explains user management in PMM.

You can manage users from the [main menu](#) by navigating to *Server Admin* → *Users* page.

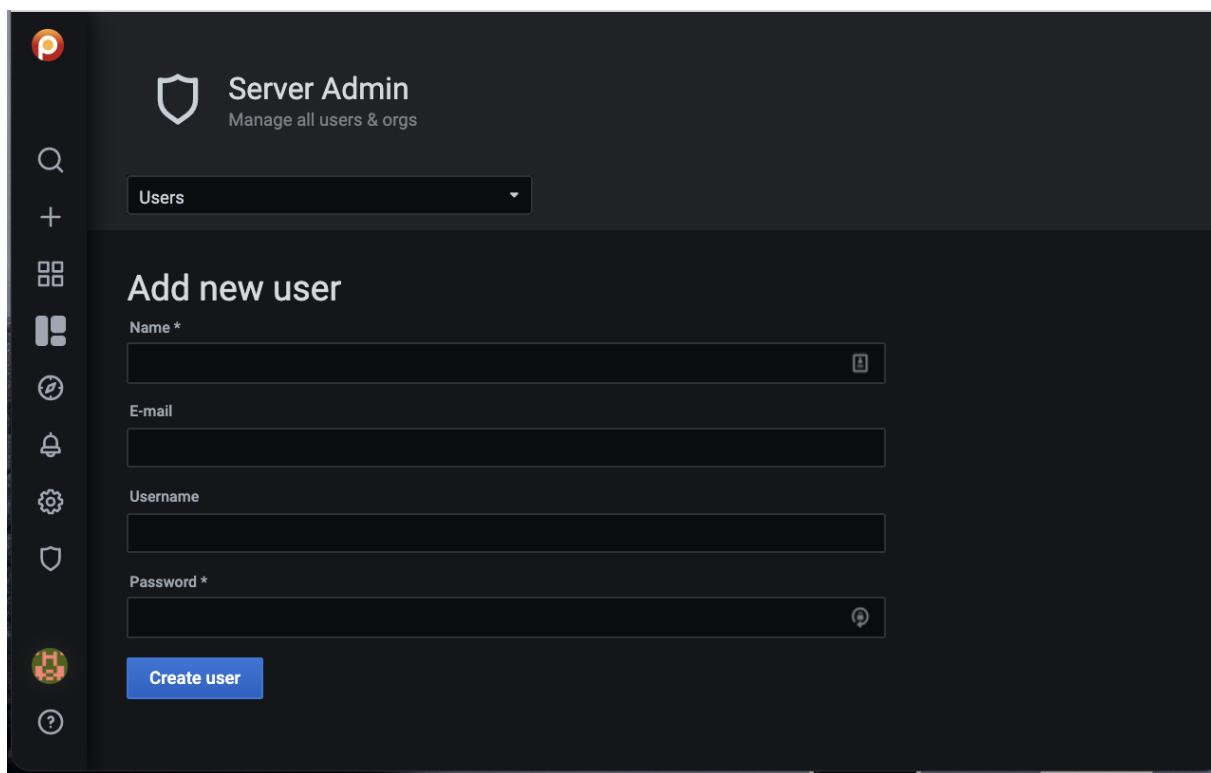
The screenshot shows the PMM Server Admin interface with the following details:

- Header:** "Server Admin" and "Manage all users & orgs".
- Left Sidebar:** Icons for Home, Search, Add, Grid, Cluster, Log, Alert, Settings, and Help.
- Search Bar:** "Users" dropdown and a search input field "Search user by login, email or name".
- Buttons:** "New user" button.
- User Table:** A table listing one user:

| Login | Email           | Name | Seen | Action      |
|-------|-----------------|------|------|-------------|
| admin | admin@localhost |      | < 1m | shield icon |
- Footer:** Links to "Documentation", "Support", and "Community".

### 4.3.1 Add users

You can add a user in PMM from *User* → *New user* tab.



To add a new user in PMM:

1. On the *Users* tab, click *New user*.
2. On the *Add new user* dialog box, enter the following:
  - Name
  - email address or username (if this is an existing grafana user)
  - Username
  - Password
3. Click *create user*.

### 4.3.2 Edit users

You can edit users by changing the information or settings for an individual user account.

#### **⚠️ Important**

After changing the default admin password for the PMM server, register the pmm-agent using the same credentials and add the services again. Otherwise, PMM will cease to monitor the service/nodes.

## Grant or Revoke admin privileges

You can grant or revoke admin access to a user as follows:

1. On the *Users* tab, click the user account you want to edit.
2. To grant or revoke the privileges, click the user. User information dialog box opens.
3. In the *Permissions* section, click *Change* and then select *Yes/No*, depending on whether you want to provide admin access or not.
4. Click *Change*.

### **⚠ Important**

After connecting your PMM instance to the Percona Platform, when you log in using your Percona account, you will be granted the *Viewer* access. For *Admin* access, log in to PMM as an admin, and change the permissions for this user.

## Change organization role

You can change the organization role assigned to your user account.

The screenshot shows the 'Server Admin' interface for managing users and organizations. The main area displays user information for 'admin1' with fields for Name, Email, Username, and Password. Below this is a 'Permissions' section where the 'Grafana Admin' role is set to 'No' and can be changed. The 'Organisations' section allows changing the main organization, with a dropdown menu currently showing 'Admin' (which is highlighted with a blue border). Other options in the dropdown are 'Editor' and 'Viewer'. At the bottom, there are buttons for 'Save' and 'Cancel', and a link to 'Remove from organisation'. The footer of the interface includes links to 'Documentation', 'Support', and 'Community'.

To change the role:

1. On the *Users* tab, click the user for whom you want to change the role.
2. In the *Organisations* section, click *Change role*.
3. Select the role from the dropdown and click *save*.

The following are the privileges for the various roles:

- Admin - Managing data sources, teams, and users within an organization.
- Editor - Creating and editing dashboards.
- Viewer - Viewing dashboards.

For detailed information on the privileges for these roles and the different tasks that they can perform, refer to: [Grafana organization roles](#).

#### 4.3.3 Delete Users

You can delete a user in PMM as follows:

1. On the *User* tab, click the user you want to delete.
  2. Click *Delete user*.
- 

Last update: 2022-01-18

## 4.4 Upgrade

### **⚠ Important**

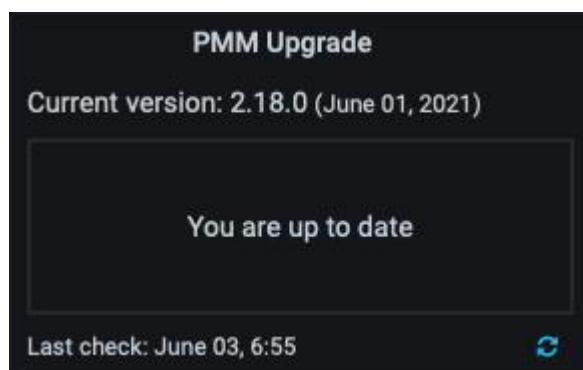
Upgrade PMM Server before upgrading PMM Clients.

### 4.4.1 Updating a Server

Client and server components are installed and updated separately.

PMM Server can run natively, as a Docker image, a virtual appliance, or an AWS cloud instance. Each has its own installation and update steps.

The preferred and simplest way to update PMM Server is with the *PMM Upgrade* panel on the Home page.



The panel shows:

- the current server version and release date;
- whether the server is up to date;
- the last time a check was made for updates.

Click the refresh button to manually check for updates.

If one is available, click the update button to update to the version indicated.

### **See also**

[PMM Server Docker upgrade](#)

### 4.4.2 Updating a PMM-Agent

PMM-Agent can be updated from tarball:

1. Download tar.gz with pmm2-client.
2. Extract it.
3. Run `./install_tarball` script with the “-u” flag.

**Hint!** The configuration file will be overwritten if you do not provide the “-u” flag while the pmm-agent is updated.

#### 4.4.3 Upgrade from PMM 1

Because of the significant architectural changes between PMM1 and PMM2, there is no direct upgrade path. The approach to making the switch from PMM version 1 to 2 is a gradual transition, outlined [in this blog post](#).

In short, it involves first standing up a new PMM2 server on a new host and connecting clients to it. As new data is reported to the PMM2 server, old metrics will age with the retention period (30 days, by default), at which point you'll be able to shut down your existing PMM1 server.

Any alerts configured through the Grafana UI will have to be recreated due to the target dashboard id's not matching between PMM1 and PMM2. In this instance we recommend moving to Alertmanager recipes in PMM2 for alerting which, for the time being, requires a separate Alertmanager instance. We are working on integrating this natively into PMM2 Server and expect to support your existing Alertmanager rules.

---

Last update: 2022-02-08

## 4.5 Secure

You can improve the security of your PMM installation with:

- [SSL encryption](#) to secure traffic between client and server;
- [Grafana HTTPS secure cookies](#)

To see which security features are enabled:

```
pmm-admin status
```



### Tip

You can gain an extra level of security by keeping PMM Server isolated from the internet, if possible.

### 4.5.1 SSL encryption

You need valid SSL certificates to encrypt traffic between client and server.

With our Docker, OVF and AMI images, self-signed certificates are in `/srv/nginx`.

To use your own, you can either:

- mount the local certificate directory to the same location, or,
- copy your certificates to a running PMM Server container.

#### Mounting certificates

For example, if your own certificates are in `/etc/pmm-certs`:

```
docker run -d -p 443:443 --volumes-from pmm-data \
--name pmm-server -v /etc/pmm-certs:/srv/nginx \
--restart always percona/pmm-server:2
```

- The certificates must be owned by root. You can do this with: `chown 0:0 /etc/pmm-certs/*`
- The mounted certificate directory (`/etc/pmm-certs` in this example) must contain the files `certificate.crt`, `certificate.key`, `ca-certs.pem` and `dhpam.pem`.
- For SSL encryption, the container must publish on port 443 instead of 80.

#### Copying certificates

If PMM Server is running as a Docker image, use `docker cp` to copy certificates. This example copies certificate files from the current working directory to a running PMM Server docker container.

```
docker cp certificate.crt pmm-server:/srv/nginx/certificate.crt
docker cp certificate.key pmm-server:/srv/nginx/certificate.key
docker cp ca-certs.pem pmm-server:/srv/nginx/ca-certs.pem
docker cp dhparam.pem pmm-server:/srv/nginx/dhparam.pem
```

#### Enabling SSL when connecting PMM Client to PMM Server

```
pmm-admin config --server-url=https://<user>:<password>@<server IP>
```

#### 4.5.2 Grafana HTTPS secure cookies

To enable:

1. Start a shell within the Docker container.

```
docker exec -it pmm-server bash
```

2. Edit `/etc/grafana/grafana.ini`.
3. Enable `cookie_secure` and set the value to `true`.
4. Restart Grafana.

```
supervisorctl restart grafana
```

---

Last update: 2021-06-29

## 4.6 Optimize

### 4.6.1 Improving PMM Performance with Table Statistics Options

If a MySQL instance has a lot of schemas or tables, there are two options to help improve the performance of PMM when adding instances with `pmm-admin add`:

- `--disable-tablestats`, or,
- `--disable-tablestats-limit`.

#### ⚠ Important

- These settings are only for adding an instance. To change them, you must remove and re-add the instances.
- Only one of these options can be used when adding an instance.

### 4.6.2 Disable per-table statistics for an instance

When adding an instance with `pmm-admin add`, the `--disable-tablestats` option disables table statistics collection when there are more than the default number (1000) of tables in the instance.

#### USAGE

```
pmm-admin add mysql --disable-tablestats
```

### 4.6.3 Change the number of tables beyond which per-table statistics is disabled

When adding an instance with `pmm-admin add`, the `--disable-tablestats-limit` option changes the number of tables (from the default of 1000) beyond which per-table statistics collection is disabled.

#### USAGE

```
pmm-admin add mysql --disable-tablestats-limit=<LIMIT>
```

#### EXAMPLE

Add a MySQL instance, disabling per-table statistics collection when the number of tables in the instance reaches 2000.

```
pmm-admin add mysql --disable-tablestats-limit=2000
```

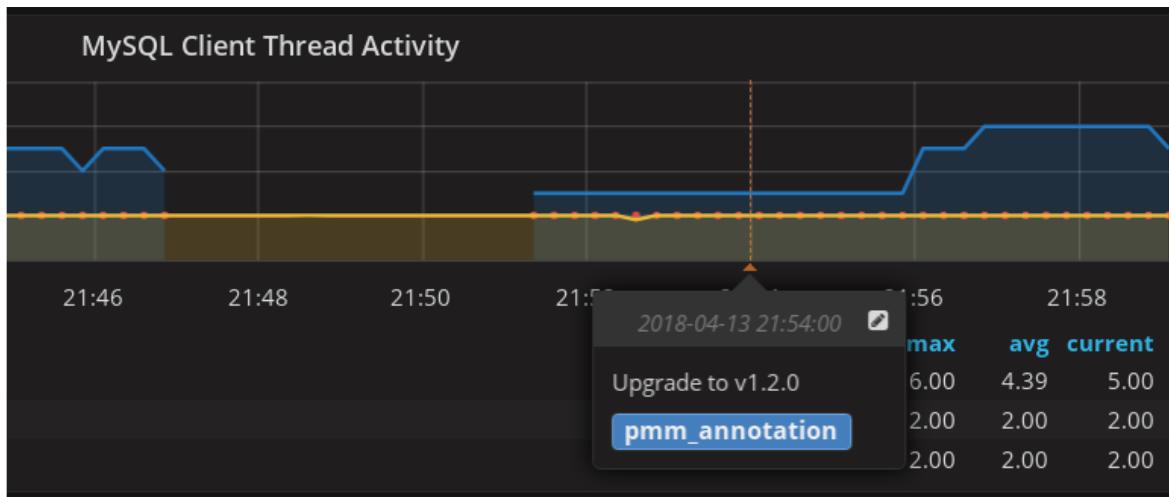
Last update: 2021-07-29

## 4.7 Annotate

Annotations mark a moment in time. They are useful for marking system changes or other significant application events. They can be set globally or for specific nodes or services.

You create them on the command line with the `pmm-admin annotate` command.

Annotations show as a vertical dashed line on a dashboard graph. Reveal the annotation text by mousing over the caret indicator below the line.



You turn annotations on or off with the *PMM Annotations* switch in the second row menu bar.



---

Last update: 2021-05-11

## 4.8 Render dashboard images

At the moment, PMM Server can't render dashboard images exported by Grafana without these steps.

### 4.8.1 Part 1: Install dependencies

1. Connect to your PMM Server Docker container.

```
docker exec -it pmm-server bash
```

2. Install Grafana plugins.

```
grafana-cli plugins install grafana-image-renderer
```

3. Restart Grafana.

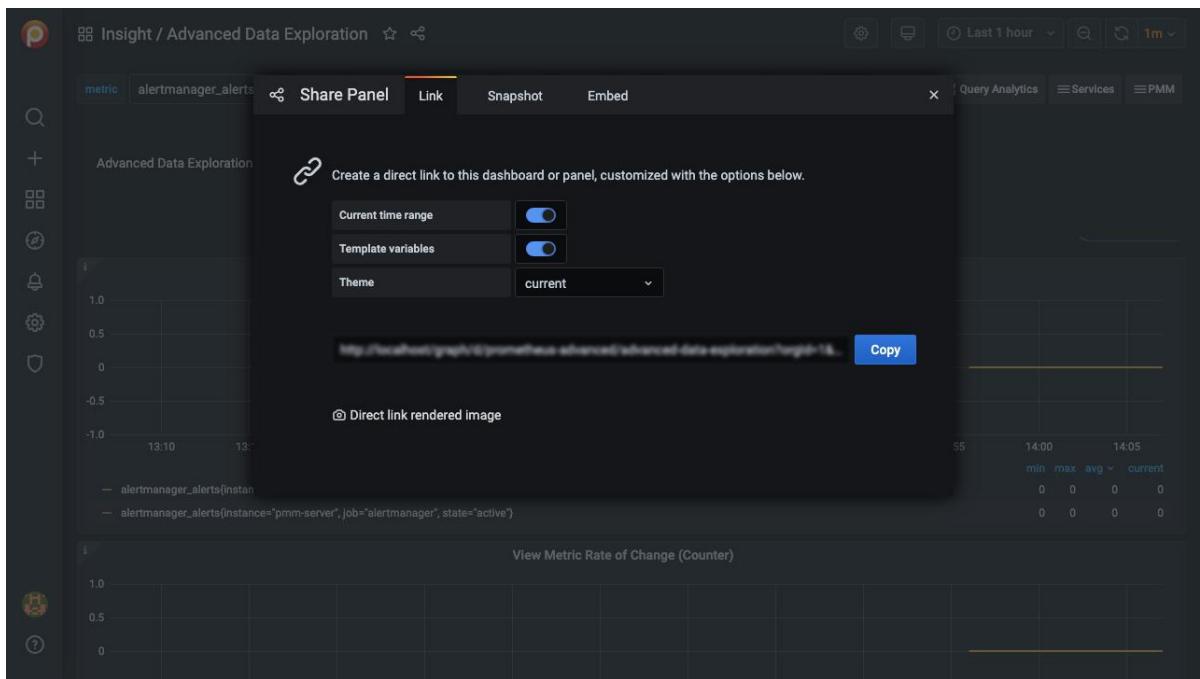
```
supervisorctl restart grafana
```

4. Install libraries.

```
yum install -y libXcomposite libXdamage libXtst cups libXScrnSaver pango \
atk adwaita-cursor-theme adwaita-icon-theme at at-spi2-atk at-spi2-core \
cairo-gobject colord-libs dconf desktop-file-utils ed emacs-filesystem \
gdk-pixbuf2 glib-networking gnutls gsettings-desktop-schemas \
gtk-update-icon-cache gtk3 hicolor-icon-theme jasper-libs json-glib \
libappindicator-gtk3 libdbusmenu libdbusmenu-gtk3 libepoxy \
liberation-fonts liberation-narrow-fonts liberation-sans-fonts \
liberation-serif-fonts libgusb libindicator-gtk3 libmodman libproxy \
libsoup libwayland-cursor libwayland-egl libxkbcommon m4 mailx nettle \
patch psmisc redhat-lsb-core redhat-lsb-submod-security rest spax time \
trousers xdg-utils xkeyboard-config alsa-lib
```

### 4.8.2 Part 2: Share the image

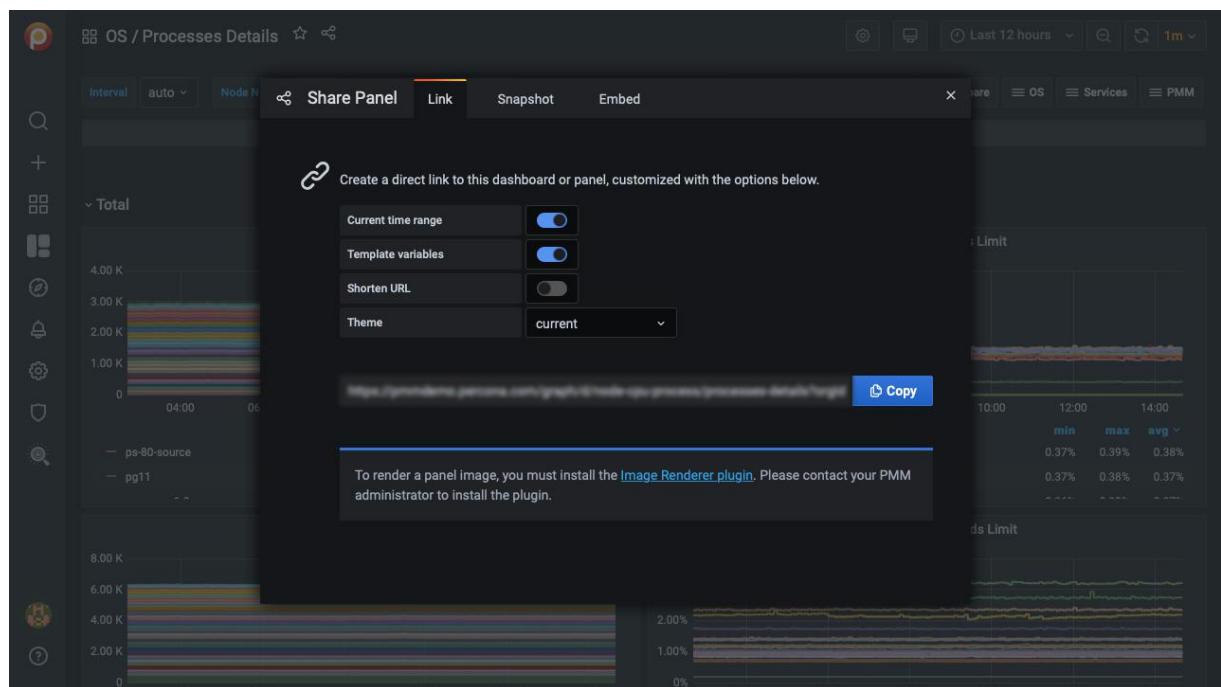
1. Navigate to the dashboard you want to share.
2. Open the panel menu.
3. Select *Share* to reveal the *Share Panel*.



4. Click *Direct link rendered image*.

5. A new browser tab opens. Wait for the image to be rendered then use your browser's image save function to download the image.

If the necessary plugins are not installed, a message in the Share Panel will say so.



Last update: 2021-06-29

## 4.9 Extend Metrics

When you need a metric that's not present in the default list of `node_exporter` metrics you may be able to use the `textfile` collector. The textfile collector allows exporting of statistics from batch jobs. It can also be used to export static metrics, such as what role a machine has.

### 4.9.1 Enable the textfile collector

The collector is enabled by default. The following folders are used for different resolutions:

| Resolution | Folder                                                                  |
|------------|-------------------------------------------------------------------------|
| High       | /usr/local/percona/pmm2/collectors/textfile-collector/high-resolution   |
| Medium     | /usr/local/percona/pmm2/collectors/textfile-collector/medium-resolution |
| Low        | /usr/local/percona/pmm2/collectors/textfile-collector/low-resolution    |

```
root@15918:~# /usr/local/percona/pmm2/exporters/node_exporter --collector.bonding --collector.buddy.info --collector.cpu --collector.diskstats --collector.entropy --collector.filesystem --collector.human --collector.loadavg --collector.meminfo --collector.meminfo_duma --collector.netdev --collector.netstat --collector.netstat.fields=~.*.(InErrors|InFrs|InCsumErrors) | Tcp.(ActIveOpens|PassiveOpens|RetransSegs|CurrEstab|AttemptFails|OutSegs) | Establishesets|OutRsts|OutSegs | Tcp.RtoAlgorithm[Min|Max] | Udp.(RcvBufFrrors|SndBufFrrors) | Udp(67)|Lite67?, | (InDatagrams|OutDatagrams|RecvBufFrrors|SndBufFrrors) | NoPorts | Icmp?(.|OutEchoReps|OutEchos|InEchoes|InEchoReps) | InAddrMaskReps | InAddrMasks | OutAddrMaskReps | OutAddrMasks | InTimestampReps | InTimestamps | OutTimestampReps | OutTimestamps | InDestUnreachs | InTimeExcds | InRedirects | OutRedirects | InMsgs | OutMsgs | IcmpMsg_(InType3|OutType3) | Ip(6|Ext) | InOctets | OutOctets | Ip_Forwarding | TcpExt_.Listen.|Syncookies.*TCPTimeouts) | --collector.proceses --collector.standard.go --collector.standar.d.go --ndand.proces --collector.rstt --collector.textfile.directory.hr=/usr/local/percona/pmm/collectors/textfile-collector/high-resolution --collector.textfile.directory.lw=/usr/local/percona/pmm/collectors/textfile-collector/low-resolution --collector.textfile.directory.mw=/usr/local/percona/pmm/collectors/textfile-collector/medium-resolution --collector.textfile.directory.rw=/usr/local/percona/pmm/collectors/textfile-collector/real-time --collector.vfs --collector.vmstat --collector.vmsstat --collector.vmsstat.fields=~(pglsta|l|kswapd|direct)|refill|alloc|movable|normal|dma32?27)|nr_(dirty|*|slab|*|vmscan|*|isolated|*|free|*|shmem|*|l2|active|*|anon_transparent|*|writeback|*|unstable|unevictable|lock|mapped|bounce|page_table_pages|kernel_stack)|drop_slabs|slabs_scanned|ppd7e?activate|ppg|in|out|pswp|in|out|ppm7a?|?fault|} --no-collector.arp --no-collector.bcache --no-collector.contrack --no-collector.d|rbd --no-collector.edac --no-collector.infiband --no-collector.interrupts --no-collector.ipv6 --no-collector.ksm --no-collector.logind --no-collector.naddm --no-collector.mountstats --no-collector.netclass --no-collector.nfs --no-collector.nfsd --no-collector.ntp --no-collector.qdisc --no-collector.runit --no-collector.sockstat --no-collector.supervisord --no-collector.systemd --no-collector.tepstat --no-collector.timex --no-collector.wifi --no-collector.xfs --no-collector.zfs --web.disable-exporter=Metrics --web.listen-address=:42801
```

The exporter parses all files in these directories that match the filename wildcard expression `*.prom` using a simple text-based [exposition format](#). Metrics are stored on the PMM Server-side with additional labels related to this Node.

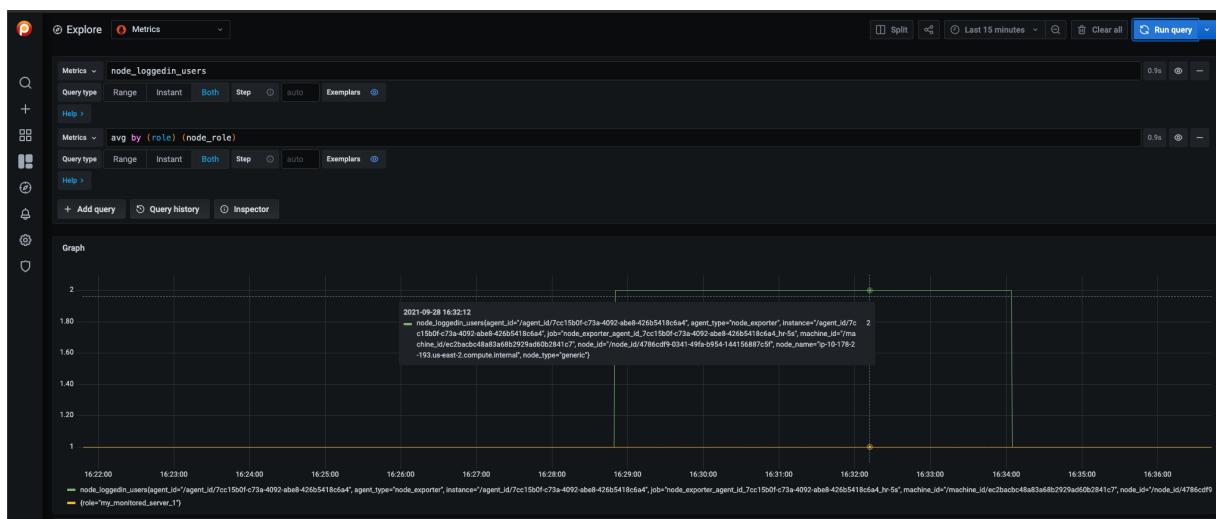
### 4.9.2 Examples of shell commands for custom metrics

To statically set roles for a machine using labels:

```
echo 'node_role{role="my_monitored_server_1"} 1' > /usr/local/percona/pmm2/collectors/textfile-collector/low-resolution/node_role.prom
```

Here's an example of a cron job that automatically pushes logged-in users:

```
$ cat /etc/cron.d/loggedin_users
*/1 * * * *      root    /usr/bin/who | /usr/bin/wc -l | sed -ne 's/^/node_loggedin_users /p'
> /usr/local/percona/pmm2/collectors/textfile-collector/high-resolution/node_users.prom
```



Last update: 2021-09-30

## 4.10 Troubleshoot

### 4.10.1 Update

If PMM server wasn't updated properly, or if you have concerns about the release, you can force the update process in 2 ways:

1. From the UI - Home panel: click with the Alt key on the reload icon in the Update panel to make the Update Button visible even if you are on the same version as available for update. Pressing this button will force the system to rerun the update so that any broken or not installed components can be installed. In this case, you'll go through the usual update process with update logs and successful messages at the end.
2. By API call (if UI not available): You can call the Update API directly with:

```
curl --user admin:admin --request POST 'http://PMM_SERVER/v1/Updates/Start'
```

Replace `admin:admin` with your username/password, and replace `PMM_SERVER` with your server address.

You will not see the logs using this method.

Refresh The Home page in 2-5 minutes and you should see that PMM was updated.

### 4.10.2 Client-server connections

There are many causes of broken network connectivity.

When [using Docker](#) the container is constrained by the host-level routing and firewall rules. For example, your hosting provider might have default `iptables` rules on their hosts that block communication between PMM Server and PMM Client, resulting in `DOWN` targets in VictoriaMetrics. If this happens, check the firewall and routing settings on the Docker host.

PMM is also able to generate diagnostics data which can be examined and/or shared with our support team to help solve an issue. You can get collected logs from PMM Client using the `pmm-admin summary` command.

Logs obtained in this way includes PMM Client logs and logs which were received from the PMM Server, stored separately in the `client` and `server` folders. The `server` folder also contains its own `client` subfolder with the self-monitoring client information collected on the PMM Server.

Beginning with [PMM 2.4.0](#), there is a flag that enables the fetching of `pprof` debug profiles and adds them to the diagnostics data. To enable, run `pmm-admin summary --pprof`.

You can get PMM Server logs with either of these methods:

#### Direct download

In a browser, visit <https://<address-of-your-pmm-server>/logs.zip>.

#### From Settings page

1. Select `Configuration` → `Settings`.
2. Click `Download server diagnostics`. (See [Diagnostics in PMM Settings](#).)

### 4.10.3 Connection difficulties

#### Passwords

When adding a service, the host might not be detected if the password contains special symbols (e.g. @, %, etc.).

In such cases, you should convert any password, replacing special characters with their escape sequence equivalents.

One way to do this is to use the `encodeURIComponent` JavaScript function in your browser's web console (commonly found under a *Development Tools* menu). Run the function with your password as the parameter. For example:

```
> encodeURIComponent("s3cR#tpa$$worD")
```

will give:

```
"s3cR%23tpa%24%24worD"
```

### 4.10.4 Integrated Alerting

#### No Integrated Alerting icon

You are not logged in as a privileged user. You need either Admin or Editor roles to work with Integrated Alerting.

##### Integrated Alerting icon but no submenu

Integrated Alerting isn't activated.

1. Go to *Configuration* → *Settings* → *Advanced Settings*.
2. Enable *Integrated Alerting*.

#### Unreachable external IP addresses

**When I get an email or page from my system the IP is not reachable from outside my organization how do I fix this?**

To configure your PMM Server's Public Address, select *Configuration* → *Settings* → *Advanced Settings*, and supply an address to use in your alert notifications.

#### What is 'Alertmanager integration'?

**There's already an Alertmanager integration tab without me turning it on, I know because I was using your existing Alertmanager integration.**

This will continue to work but will be renamed *External Alertmanager*.

#### Notification channels not working

**I tried to setup a Slack/Email channel but nothing happened.**

Before you can use a notification channel you must provide your connection details.

1. Go to *PMM* → *PMM Settings* → *Communication*.
2. Define your SMTP server or Slack incoming webhook URL.

For PagerDuty you can configure in the notification channel tab of Integrated Alerting by supplying your server/routing key.

### What's the difference: Username/Password vs Identity/Secret

**In configuring my email server I'm being asked for a Username and Password as well as Identity and Secret. What is the difference between these and which do I use or do I need both?**

It depends on what kind of authentication your system uses:

- LOGIN : Use Username/Password.
- PLAIN : Use either Username or Identity and Password.
- CRAM-MD5 : Use Username and Secret.

### Alert Rule Templates is disabled

Built-In alerts are not editable, but you can copy them and edit the copies. (In [PMM 2.14.0](#) and above).

If you create a custom alert rule template you will have access to edit.

### Creating rules

**I'm ready to create my first rule! I've chosen a template and given it a name...what is the format of the fields?**

- Threshold - float value, it has different meanings depending on what template is used.
- Duration - The duration the condition must be satisfied in seconds.
- Filters - A Key, Evaluator, and Value. E.g. `service_name=ps5.7`
  - Key must be an exact match. You can find a complete list of keys by using the [Explore](#) main menu item in PMM.
  - Evaluator can be: `=` `=~`.
  - Value is an exact match or when used with a 'fuzzy' evaluator (`=~`) can be a regular expression. E.g. `service_name=~ps.*`

### Variables in Templates

**The concept of *template* implies features like variable substitutions...where can I use these? Where can I find a complete list of them?**

Here is a guide to creating templates for Alertmanager: [https://prometheus.io/docs/prometheus/latest/configuration/template\\_examples/](https://prometheus.io/docs/prometheus/latest/configuration/template_examples/)

## 4.10.5 Missing data

### Why don't I see the whole query?

To reduce space usage, long query examples and fingerprints can be truncated to 1024 symbols. In this case, the query explains section will not work.

Last update: 2021-08-27

## 5. Details

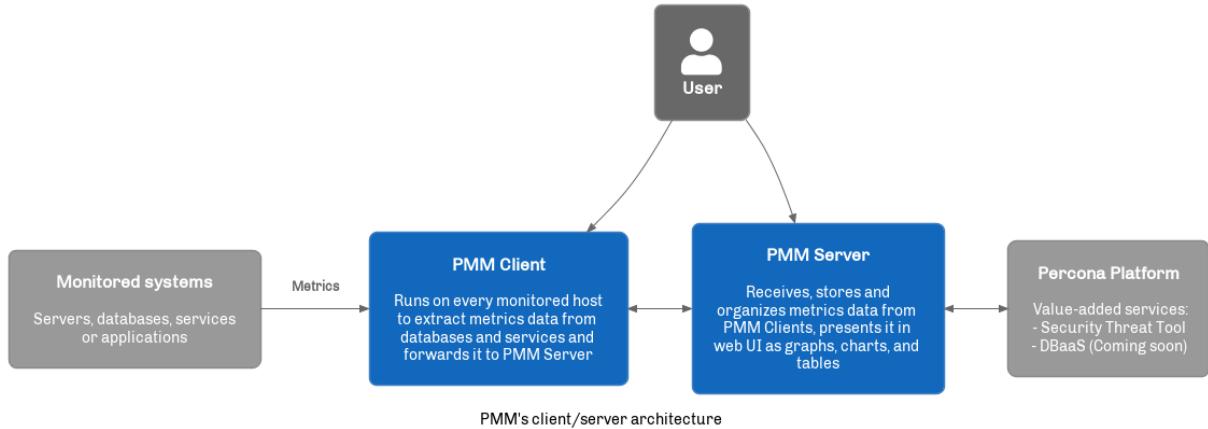
### 5.1 Details

- [Architecture](#): high-level architecture and main components.
  - [User interface components](#): Descriptions of the main menus and icons.
  - [Dashboards reference](#): A complete list of dashboards by category, with screenshots.
  - Commands:
    - [pmm-admin](#): The manual page for the PMM administration tool.
    - [pmm-agent](#): The manual page for the PMM Client agent program.
  - [API](#): How to access the Swagger API.
  - [VictoriaMetrics](#): the third-party monitoring solution and time-series database that replaced Prometheus in [PMM 2.12.0](#).
  - [Glossary](#): A list of obscure terms and definitions.
- 

Last update: 2021-07-07

## 5.2 Architecture

PMM is a client/server application built by us with our own and third-party open-source tools.



### 5.2.1 PMM Server

PMM Server is the heart of PMM. It receives data from clients, collates it and stores it. Metrics are drawn as tables, charts and graphs within [dashboards](#), each a part of the web-based [user interface](#).

### 5.2.2 PMM Client

PMM Client is a collection of agents and exporters that run on the host being monitored.

PMM Client runs on every database host or node you want to monitor. The client collects server metrics, general system metrics, and query analytics data, and sends it to the server. Except when monitoring AWS RDS instances, a PMM Client must be running on the host to be monitored.

### 5.2.3 Percona Platform

[Percona Platform](#) (in development) provides value-added services for PMM.

### 5.2.4 PMM context

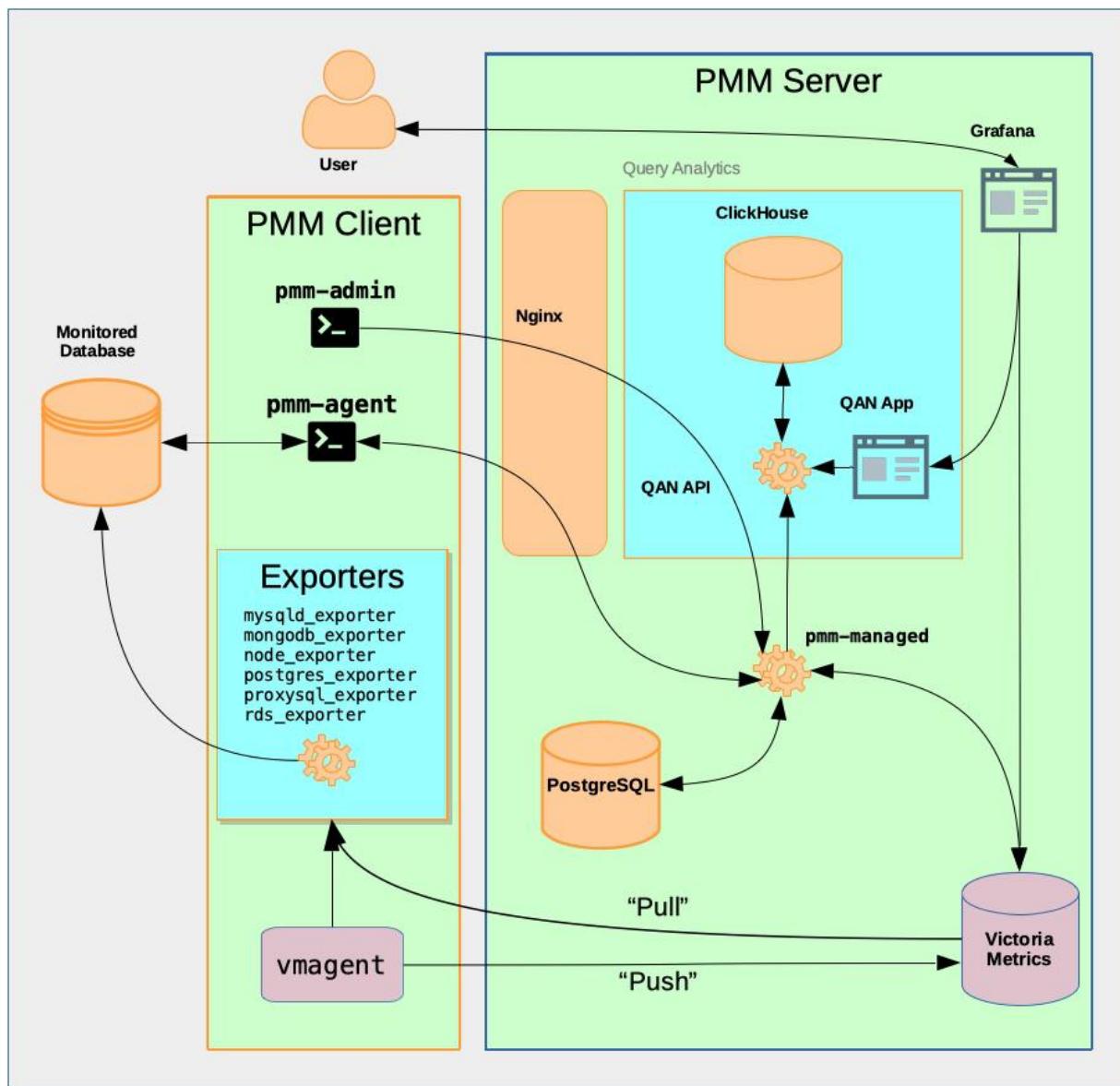
The PMM Client package provides:

- Exporters for each database and service type. When an exporter runs, it connects to the database or service instance, runs the metrics collection routines, and sends the results to PMM Server.
- `pmm-agent` : Run as a daemon process, it starts and stops exporters when instructed.
- `vmaagent` : A VictoriaMetrics daemon process that sends metrics data (*pushes*) to PMM Server.

The PMM Server package provides:

- `pmm-managed` ;
- Query Analytics;
- Grafana;
- VictoriaMetrics.

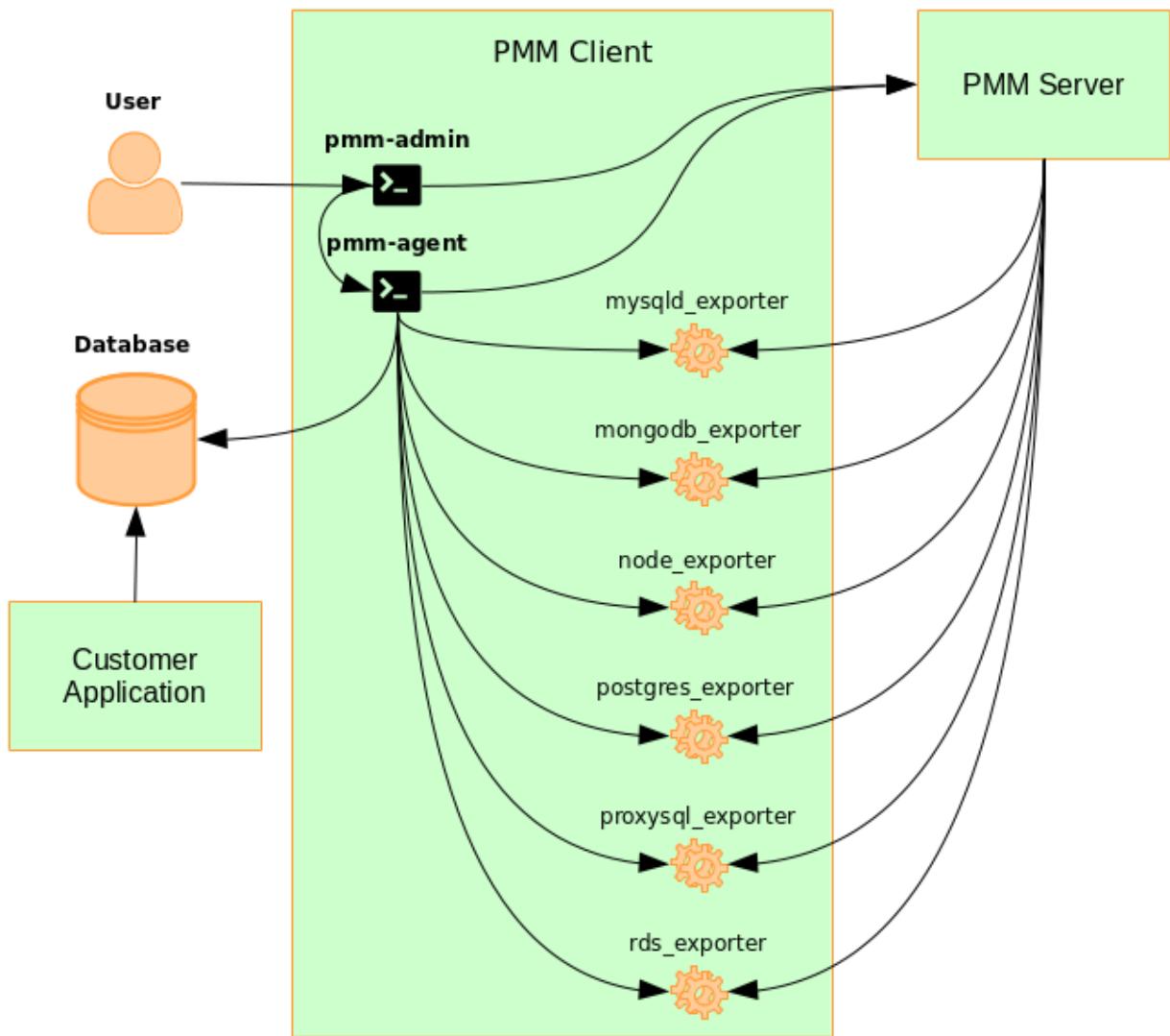
#### PMM Server



PMM Server includes the following tools:

- Query Analytics (QAN) enables you to analyze MySQL query performance over periods of time. In addition to the client-side QAN agent, it includes the following:
  - QAN API is the back-end for storing and accessing query data collected by the QAN agent running on a PMM Client.
  - QAN Web App is a web application for visualizing collected Query Analytics data.
- Metrics Monitor provides a historical view of metrics that are critical to a MySQL or MongoDB server instance. It includes the following:
  - VictoriaMetrics, a scalable time-series database. (Replaced Prometheus in PMM 2.12.0.)
  - ClickHouse is a third-party column-oriented database that facilitates the Query Analytics functionality.
  - Grafana is a third-party dashboard and graph builder for visualizing data aggregated (by VictoriaMetrics or Prometheus) in an intuitive web interface.
  - Percona Dashboards is a set of dashboards for Grafana developed by us.

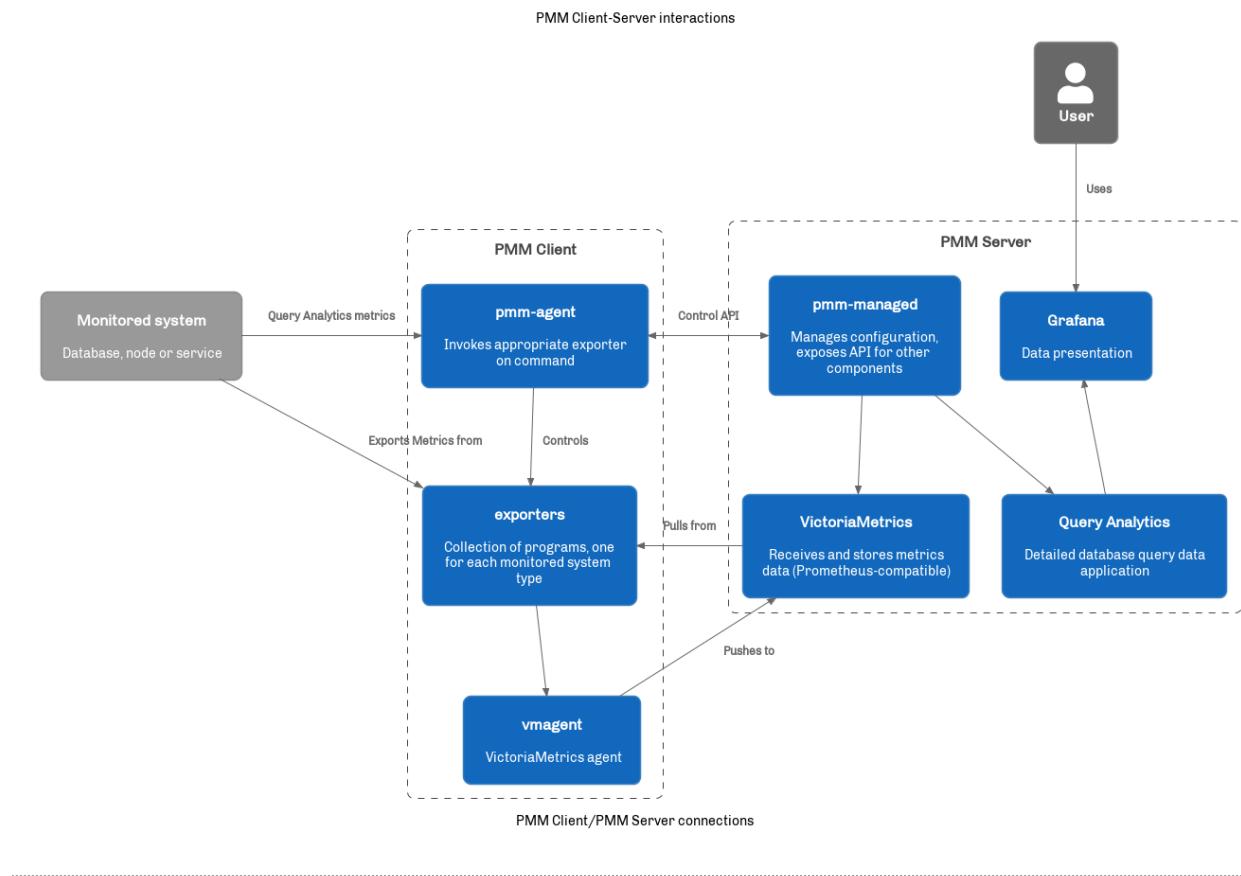
#### PMM Client



The PMM Client package consists of the following:

- `pmm-admin` is a command-line tool for managing PMM Client, for example, adding and removing database instances that you want to monitor. ([Read more](#)).
- `pmm-agent` is a client-side component a minimal command-line interface, which is a central entry point in charge for bringing the client functionality: it carries on client's authentication, gets the client configuration stored on the PMM Server, manages exporters and other agents.
- `node_exporter` is an exporter that collects general system metrics.
- `mysqld_exporter` is an exporter that collects MySQL server metrics.
- `mongodb_exporter` is an exporter that collects MongoDB server metrics.
- `postgres_exporter` is an exporter that collects PostgreSQL performance metrics.
- `proxysql_exporter` is an exporter that collects ProxySQL performance metrics.
- `rds_exporter` is an exporter that collects Amazon RDS performance metrics.
- `azure_database_exporter` is an exporter that collects Azure database performance metrics.

To make data transfer from PMM Client to PMM Server secure, all exporters are able to use SSL/TLS encrypted connections, and their communication with the PMM server is protected by the HTTP basic authentication.



Last update: 2021-12-09

## 5.3 UI components

The screenshot shows the PMM dashboard with several numbered callouts:

- 1**: A vertical sidebar menu on the left.
- 2**: The title bar "Insight / Home Dashboard" with a star icon and a refresh button.
- 3**: The top navigation bar with icons for Home, Query Analytics, Services, and PMM.
- 4**: View selector dropdowns for Interval (auto), Environment (All), Node Name (All).
- 5**: The main content area titled "Percona Monitoring and Management".

**General information**

**Overview**  
Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL and MongoDB performance, and provides time-based analysis to ensure that your data works as efficiently as possible.

**Documentation**  
Please consult the official [PMM documentation](#) to learn more about PMM. Also of interest are the [Release notes](#) and [FAQ](#) for common questions about PMM.

**Community and Blogs**  
On the [PMM Community Forums](#) you will find help from Perconians and the Community at large. Further, we publish PMM announcements and use cases regularly on the [Percona Database Performance Blog](#).

**Environment Overview**

| Host              | CPU B... | Mem ... | Disk R... | Disk ...  | Netwo...  | DB Co... | DB QPS  | Virtual... | RAM        | lost u... | JB upt...  |
|-------------------|----------|---------|-----------|-----------|-----------|----------|---------|------------|------------|-----------|------------|
| All               | 28%      | 52%     | 2.0 MB/s  | 37.2 MB/s | 62.3 MB/s | 454      | 15.85 K | 62         | 127.56 GiB | 4.5 weeks | 2.9 hours  |
| AzureDB-mysql     | N/A      | N/A     | N/A       | N/A       | N/A       | 19       | 142.95  | N/A        | N/A        | N/A       | 15.3 weeks |
| load-gen-sysbench | 60%      | 45%     | 0 B/s     | 2.3 kB/s  | 25.4 MB/s | N/A      | N/A     | 1          | 1.77 GiB   | 4.5 weeks | N/A        |

**Percona News**

- Using MySQL 8 Dual Passwords (Jul 01)
- Installing Percona Server for MySQL on Rocky Linux 8 (Jul 01)

**Starred & Recently us...**

- Home Dashboard
- PMM Query Analytics
- ProxySQL Instance Summary
- VictoriaMetrics Agents Overview
- VictoriaMetrics

**Monitored nod...** 30

**Monitored DB ...** 28

**Failed security checks** 2 / 9 / 2

**PMM Upgrade**  
Current version: 2.19.0 (June 30, 2021)  
You are up to date  
Last check: July 04, 16:02



Key

1. Main menu (also *Grafana menu, side menu*)
2. Navigation bar
3. View controls
4. View selectors (dynamic contents)
5. Shortcut menu (dynamic contents)

### 5.3.1 Main menu

The main menu is part of the Grafana framework and is visible on every page.

| Item (Top)                                                                          | Name                | Description                                                                         |
|-------------------------------------------------------------------------------------|---------------------|-------------------------------------------------------------------------------------|
|    | Home                | Link to home dashboard.                                                             |
|                                                                                     | Search              | Search dashboards by name.                                                          |
|                                                                                     | Create              | Create dashboards or <a href="#">folders</a> , import dashboards.                   |
|                                                                                     | Dashboards          | Manage dashboards, create playlists, manage snapshots.                              |
|    | PMM Dashboards      | Replacement for <a href="#">shortcut menu</a> .                                     |
|                                                                                     | Explore             | Run queries with PromQL.                                                            |
|                                                                                     | Alerting            | Alerting, <a href="#">Integrated Alerting</a> , Alert Rules, Notification Channels. |
|                                                                                     | Configuration       |                                                                                     |
|                                                                                     | Server Admin        |                                                                                     |
|                                                                                     | Backup Management   | <a href="#">Backup management and storage location configuration</a> .              |
|   | PMM Database Checks |                                                                                     |
|  | DBaaS               |                                                                                     |

#### Tip

The DBaaS icon appears only if a server feature flag has been set.

The Backup Management icon appears when *Backup Management* is activated in [Configuration](#) → [Settings](#) → [Advanced Settings](#).

| Icon (Bottom)  | Description |
|----------------|-------------|
| (Profile icon) | User menu   |
|                | Help        |

### 5.3.2 Navigation bar



| Item (left) | Description             |
|-------------|-------------------------|
|             | (Display only.)         |
| (Name) /    | (Optional) Folder name. |
| (Name)      | Dashboard name.         |
|             | Mark as favorite.       |
|             | Share dashboard.        |

### 5.3.3 View controls

| Item (right)    | Description          |
|-----------------|----------------------|
|                 | Dashboard settings.  |
|                 | Cycle view mode.     |
| ⌚ (time range)  | Time range selector. |
|                 | Time range zoom out. |
|                 | Refresh dashboard.   |
| (Time interval) | Refresh period.      |

### 5.3.4 View selectors

This menu bar is context sensitive; it changes according to the page you are on. (With wide menus on small screens, items may wrap to the next row.)



| Item            | Description                        |
|-----------------|------------------------------------|
| Interval        | Data interval.                     |
| Region          | Filter by region.                  |
| Environment     | Filter by environment.             |
| Cluster         | Filter by cluster.                 |
| Replication Set | Filter by replication set.         |
| Node Name       | Filter by node name.               |
| Service Name    | Filter by service name.            |
| PMM Annotations | <a href="#">View annotations</a> . |

### 5.3.5 Shortcut menu

This menu contains shortcuts to other dashboards. The list changes according to the page you're on.

This menu will be removed in future releases. Its function will be replaced by the  PMM Dashboards main menu entry.

| Item            | Description                                             |
|-----------------|---------------------------------------------------------|
| Home            | Home dashboard.                                         |
| Query Analytics | Query Analytics.                                        |
| Compare         | Nodes compare.                                          |
| (Service Type)  | Service type menu (see <a href="#">Services menu</a> ). |
| HA              | HA dashboards.                                          |
| Services        | Services menu.                                          |
| PMM             | PMM menu.                                               |

### Tip

The *Compare* menu links to the Instances Overview dashboard for the current service type.

## Services menu

The *Services* menu choice determines the Service Type menu.

| Menu     | Item                          | Service type menu | Description            |
|----------|-------------------------------|-------------------|------------------------|
| Services |                               |                   |                        |
|          | MongoDB Instances Overview    | MongoDB           | MongoDB dashboards.    |
|          | MySQL Instances Overview      | MySQL             | MySQL dashboards.      |
|          | Nodes Overview                | OS                | OS dashboards.         |
|          | PostgreSQL Instances Overview | PostgreSQL        | PostgreSQL dashboards. |

## PMM menu

This item lists shortcuts to utility pages.

| Menu | Item                |
|------|---------------------|
| PMM  |                     |
|      | PMM Add Instance    |
|      | PMM Database Checks |
|      | PMM Inventory       |
|      | PMM Settings        |

Last update: 2021-07-07

## 5.4 Dashboards

### 5.4.1 Dashboards

| Category   | Dashboard                              | Elements |
|------------|----------------------------------------|----------|
| Insight    | Advanced Data Exploration              | 7        |
| Insight    | Home Dashboard                         | 26       |
| Insight    | Prometheus Exporter Status             | 57       |
| Insight    | Prometheus Exporters Overview          | 27       |
| Insight    | VictoriaMetrics                        | 52       |
| Insight    | VictoriaMetrics Agents Overview        | 58       |
| PMM        | PMM Inventory                          | 3        |
| PMM        | Environment Overview                   | 0        |
| PMM        | Environment Summary                    | 0        |
| DBaaS      | DB Cluster Summary                     | 0        |
| OS         | CPU Utilization Details                | 21       |
| OS         | Disk Details                           | 34       |
| OS         | Network Details                        | 70       |
| OS         | Memory Details                         | 116      |
| OS         | Node Temperature Details               | 6        |
| OS         | Nodes Compare                          | 74       |
| OS         | Nodes Overview                         | 115      |
| OS         | Node Summary                           | 67       |
| OS         | NUMA Details                           | 72       |
| OS         | Processes Details                      | 35       |
| Prometheus | Prometheus Exporter Status             | 57       |
| Prometheus | Prometheus Exporters Overview          | 27       |
| MySQL      | MySQL Amazon Aurora Details            | 20       |
| MySQL      | MySQL Command/Handler Counters Compare | 11       |
| MySQL      | MySQL InnoDB Compression Details       | 41       |
| MySQL      | MySQL InnoDB Details                   | 339      |
| MySQL      | MySQL MyISAM/Aria Details              | 55       |
| MySQL      | MySQL MyRocks Details                  | 101      |
| MySQL      | MySQL Instance Summary                 | 90       |
| MySQL      | MySQL Instances Compare                | 70       |
| MySQL      | MySQL Instances Overview               | 96       |
| MySQL      | MySQL Wait Event Analyses Details      | 42       |
| MySQL      | MySQL Performance Schema Details       | 48       |
| MySQL      | MySQL Query Response Time Details      | 49       |

| Category          | Dashboard                                       | Elements |
|-------------------|-------------------------------------------------|----------|
| MySQL             | <a href="#">MySQL Replication Summary</a>       | 50       |
| MySQL             | <a href="#">MySQL Group Replication Summary</a> | 18       |
| MySQL             | <a href="#">MySQL Table Details</a>             | 45       |
| MySQL             | <a href="#">MySQL User Details</a>              | 62       |
| MySQL             | <a href="#">MySQL TokuDB Details</a>            | 172      |
| MongoDB           | <a href="#">MongoDB Cluster Summary</a>         | 55       |
| MongoDB           | <a href="#">MongoDB Instance Summary</a>        | 42       |
| MongoDB           | <a href="#">MongoDB Instances Overview</a>      | 100      |
| MongoDB           | <a href="#">MongoDB Instances Compare</a>       | 19       |
| MongoDB           | <a href="#">MongoDB ReplSet Summary</a>         | 130      |
| MongoDB           | <a href="#">MongoDB InMemory Details</a>        | 46       |
| MongoDB           | <a href="#">MongoDB MMAPv1 Details</a>          | 52       |
| MongoDB           | <a href="#">MongoDB WiredTiger Details</a>      | 54       |
| PostgreSQL        | <a href="#">PostgreSQL Instances Overview</a>   | 114      |
| PostgreSQL        | <a href="#">PostgreSQL Instance Summary</a>     | 67       |
| PostgreSQL        | <a href="#">PostgreSQL Instances Compare</a>    | 89       |
| ProxySQL          | <a href="#">ProxySQL Instance Summary</a>       | 55       |
| High-availability | <a href="#">PXC/Galera Node Summary</a>         | 32       |
| High-availability | <a href="#">PXC/Galera Cluster Summary</a>      | 19       |
| High-availability | <a href="#">PXC/Galera Nodes Compare</a>        | 55       |
| High-availability | <a href="#">HAProxy Instance Summary</a>        | 113      |

Last update: 2022-02-08

## 5.4.2 Insight

### Home Dashboard

The Home Dashboard is a high-level overview of your environment, the starting page of the PMM portal from which you can open the tools of PMM, and browse to online resources.

On the PMM home page, you can also find the version number and a button to update your PMM Server.

#### GENERAL INFORMATION

This section contains links to online resources, such as PMM documentation, releases notes, and blogs.

#### SHARED AND RECENTLY USED DASHBOARDS

This section is automatically updated to show the most recent dashboards that you worked with. It also contains the dashboards that you have bookmarked.

## STATISTICS

This section shows the total number of hosts added to PMM and the total number of database instances being monitored. This section also current the version number. Use the *Check for Updates Manually* button to see if you are using the most recent version of PMM.

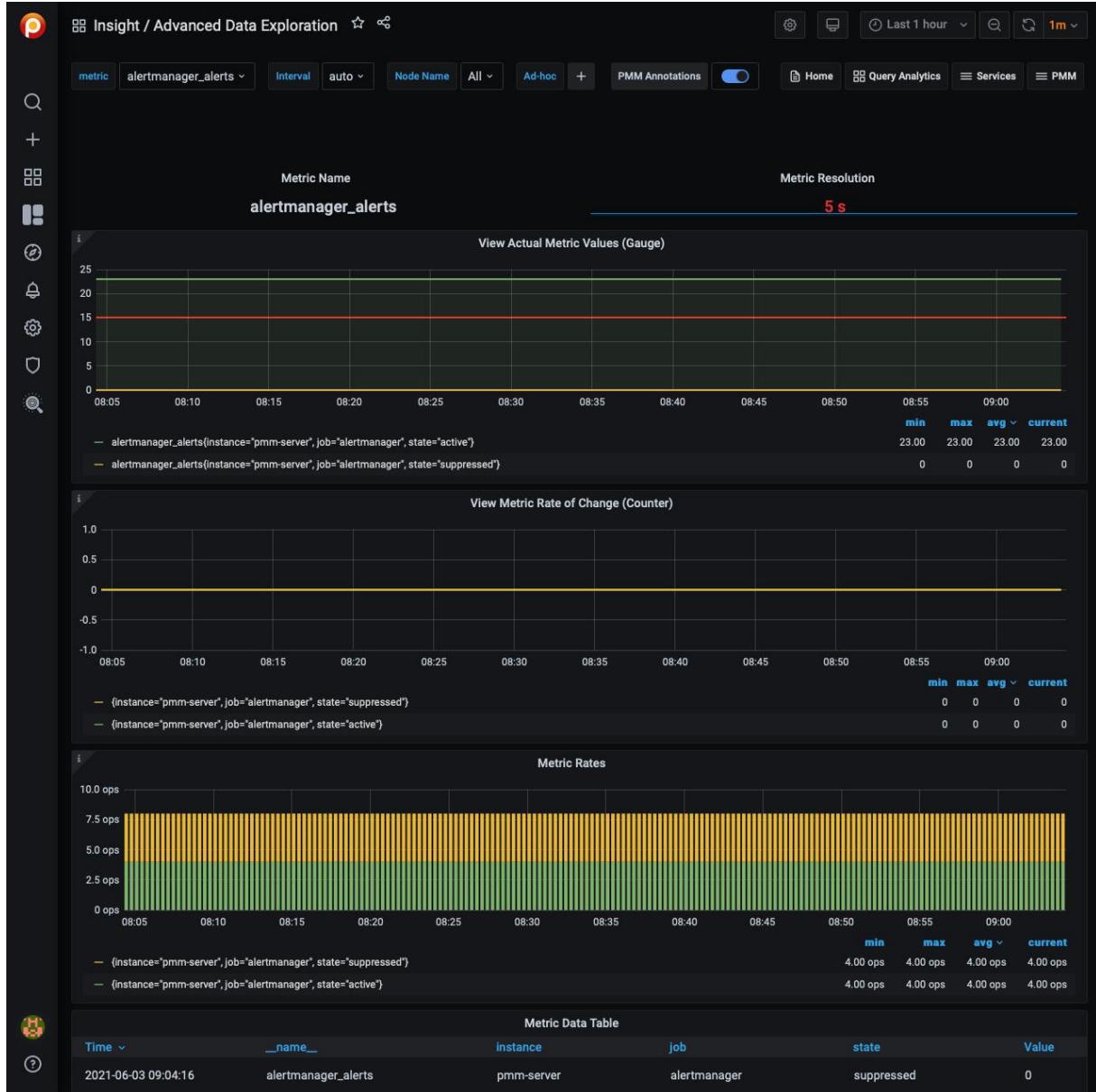
## ENVIRONMENT OVERVIEW

This section lists all added hosts along with essential information about their performance. For each host, you can find the current values of the following metrics:

- CPU Busy
  - Memory Available
  - Disk Reads
  - Disk Writes
  - Network IO
  - DB Connections
  - DB QPS
  - Virtual CPUs
  - RAM
  - Host Uptime
  - DB Uptime
- 

Last update: 2022-01-10

## Advanced Data Exploration



The *Advanced Data Exploration* dashboard provides detailed information about the progress of a single Prometheus metric across one or more hosts.

### VIEW ACTUAL METRIC VALUES (GAUGE)

A gauge is a metric that represents a single numerical value that can arbitrarily go up and down.

Gauges are typically used for measured values like temperatures or current memory usage, but also “counts” that can go up and down, like the number of running goroutines.

### VIEW METRIC RATE OF CHANGE (COUNTER)

A counter is a cumulative metric that represents a single numerical value that only ever goes up. A counter is typically used to count requests served, tasks completed, errors occurred, etc. Counters should not be used to expose current counts of items whose number can also go down, e.g. the number of currently running goroutines. Use gauges for this use case.

#### METRIC RATES

Shows the number of samples Per second stored for a given interval in the time series.

This dashboard supports metrics related to NUMA. The names of all these metrics start with `node_memory_numa`.

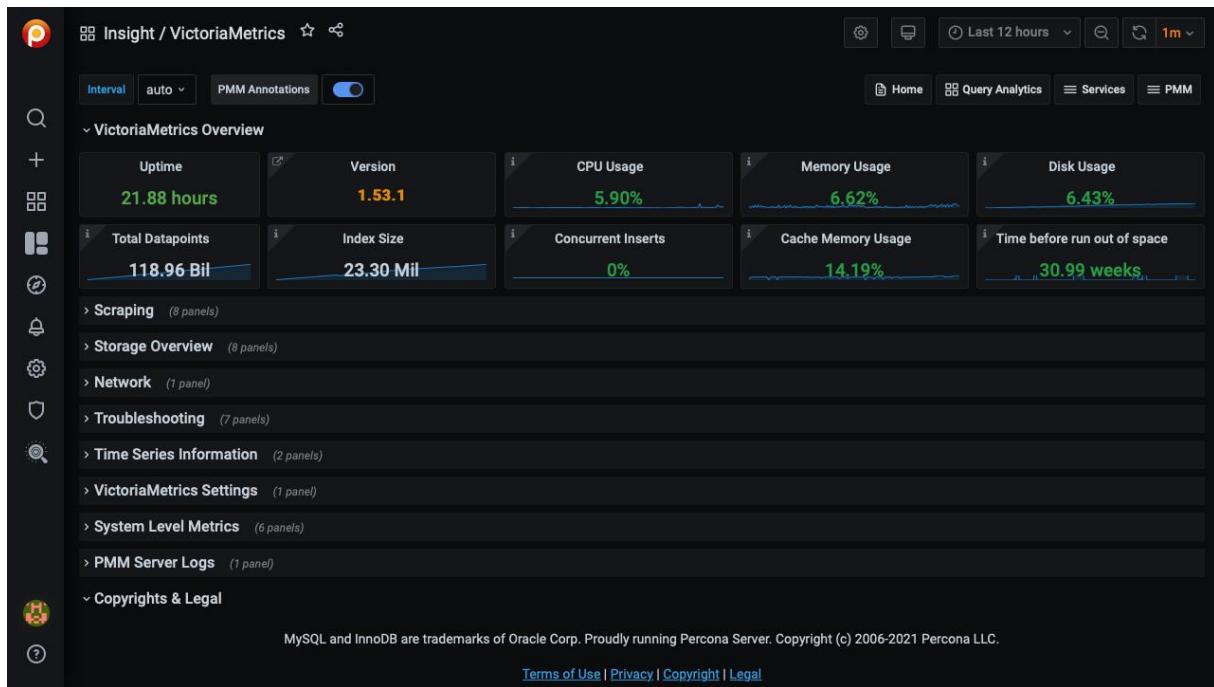
The screenshot shows a search interface for metrics. The search bar contains the query `node_memory_numa`. Below the search bar, there are several dropdown and selection buttons: `Interval` (set to `auto`), `Host` (set to `All`), and a checked checkbox for `PMM Annotations`. A vertical scroll bar is visible on the right side of the metric list. The list itself contains 14 items, each starting with `node_memory_numa_`:

- `node_memory_numa_Active`
- `node_memory_numa_Active_anon`
- `node_memory_numa_Active_file`
- `node_memory_numa_AnonHugePages`
- `node_memory_numa_AnonPages`
- `node_memory_numa_Bounce`
- `node_memory_numa_Dirty`
- `node_memory_numa_FilePages`
- `node_memory_numa_HugePages_Free`
- `node_memory_numa_HugePages_Surp`
- `node_memory_numa_HugePages_Total`
- `node_memory_numa_Inactive`
- `node_memory_numa_Inactive_anon`
- `node_memory_numa_Inactive_file`

---

Last update: 2021-06-03

## VictoriaMetrics

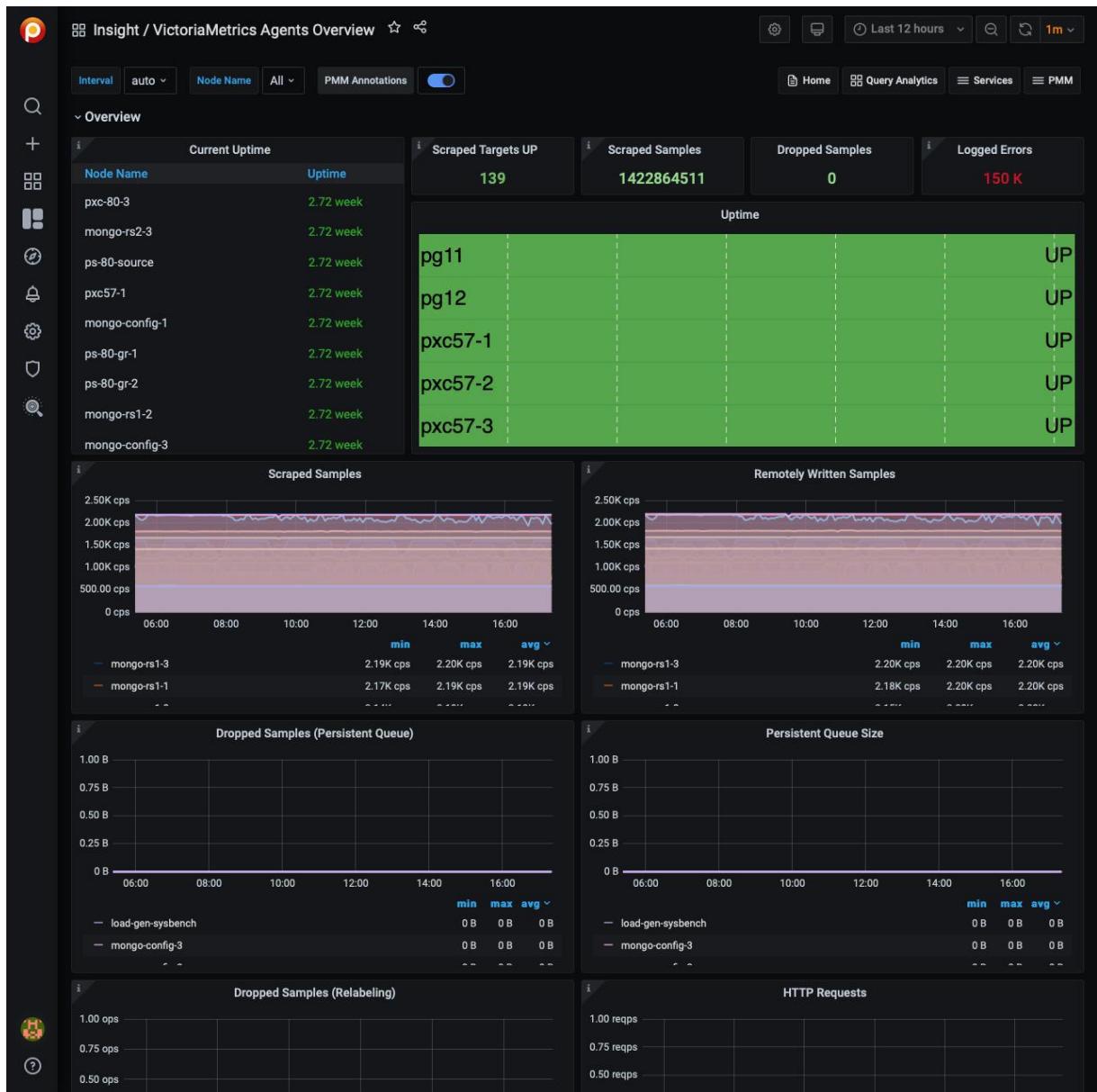


No description

---

Last update: 2021-05-11

## VictoriaMetrics Agents Overview



No description

Last update: 2021-05-11

### 5.4.3 PMM

#### PMM Inventory

The *Inventory* dashboard is a high level overview of all objects registered by PMM.

To see it select *Configuration* → *PMM Inventory* → *Inventory list*.

|                          | ID                                               | Service Type | Service name           | Node ID                                       | Addresses                                | Port  | Other Details                                                                                                  |
|--------------------------|--------------------------------------------------|--------------|------------------------|-----------------------------------------------|------------------------------------------|-------|----------------------------------------------------------------------------------------------------------------|
| <input type="checkbox"/> | /service_id/c400c998-292a-437b-9237-a581aba39e84 | MySQL        | AzureDB-mysql          | /node_id/b86dd014-f9f5-4054-ab2a-0ca3acab5d6d | pmmdemo-azuredb.mysql.database.azure.com | 3306  |                                                                                                                |
| <input type="checkbox"/> | /service_id/04bc17d5-18dd-441d-b4c4-f54565258eee | MySQL        | mariadb-105-mysql      | /node_id/a80a1fe7-cca3-4236-8c03-bd642d8e7038 | 127.0.0.1                                | 3306  | environment: Prod                                                                                              |
| <input type="checkbox"/> | /service_id/cc7c3b4d-7f02-4c0b-b2a9-b27e3744eebc | MongoDB      | mongo-config-1-mongodb | /node_id/741e26a3-6dd5-44f2-9f0a-8642c66bc48c | 127.0.0.1                                | 27017 | environment: Staging cluster: MongoDBCluster1 replication_set: MongoDBReplicaSet3 az: sfo1 city: San_Francisco |
| <input type="checkbox"/> | /service_id/40f99ebd-af4b-4b29-8efb-51a45fd37a1f | MongoDB      | mongo-config-2-mongodb | /node_id/588dcc23-60a2-44b5-b276-ec086b9d81d2 | 127.0.0.1                                | 27017 | environment: Staging cluster: MongoDBCluster1 replication_set: MongoDBReplicaSet3 az: sfo1 city: San_Francisco |
| <input type="checkbox"/> | /service_id/8752e76b-d5f1-4f29-885e-f783257a2ad4 | MongoDB      | mongo-config-3-mongodb | /node_id/68501ed5-4e66-4098-9055-23c8426877c5 | 127.0.0.1                                | 27017 | environment: Staging cluster: MongoDBCluster1 replication_set: MongoDBReplicaSet3 az: sfo1 city: San_Francisco |
| <input type="checkbox"/> | /service_id/f7a72752-6897-4038-803e-c3eeda55acc  | MongoDB      | mongo-mongos           | /node_id/5761899d-9c5c-4c8d-87ef-394841fcdb53 | 127.0.0.1                                | 27017 | environment: Staging cluster: MongoDBCluster1 replication_set: MongoDBReplicaSet3 az: sfo1 city: San_Francisco |
| <input type="checkbox"/> | /service_id/ce81acd7-93ae-476d-ae13-54063cab846  | MongoDB      | mongo-rs1-1-mongodb    | /node_id/fbc3368c-2e3f-4f6c-9f5b-29480a40def7 | 127.0.0.1                                | 27018 | environment: Staging cluster: MongoDBCluster1 replication_set: MongoDBReplicaSet3 az: sfo1                     |

Inventory objects form a hierarchy with Node at the top, then Service and Agents assigned to a Node.

There are three tabs where items for each type are listed with their details:

- *Services*

Individual service names and where they run, against which agents will be assigned. Each instance of a service gets a `service_id` value that is related to a `node_id`. Examples are MySQL, Amazon Aurora MySQL. This feature also allows to support multiple mysqld instances on a single node, with different service names, e.g. `mysql1-3306`, and `mysql1-3307`.

- *Agents*

Each binary (exporter, agent) running on a client will get an `agent_id` value. Examples:

- `pmm-agent` one is the top of the tree, assigned to a `node_id`
- `node_exporter` is assigned to pmm-agent `agent_id`
- `mysqld_exporter` and QAN MySQL Perfschema are assigned to a `service_id`.

- *Nodes*

Where the service and agents will run. Assigned a `node_id`, associated with a `machine_id` (from `/etc/machine-id`). Some examples are bare metal, virtualized, container.

## REMOVING ITEMS FROM THE INVENTORY

You can remove items from the inventory.

1. Select **Configuration** → **PMM Inventory** → **Inventory list**.

2. In the first column, select the items to be removed.

The screenshot shows the PMM Inventory list page. On the left is a sidebar with icons for Home Dashboard, Configuration, Services, Agents, Nodes, and Help. The main area has tabs for Services, Agents, and Nodes, with Services selected. A red 'Delete' button is at the top right. Below is a table with columns: ID, Service Type, Service name, Node ID, Addresses, Port, and Other Details. Two rows are selected: one for a MongoDB service with ID /service\_id/86b91725-be64-442e-a54e-f789da48d4a0 and another for a MongoDB service with ID /service\_id/b0dda75b-5825-4be9-91c6-97761845cf92.

3. Click **Delete**. The interface will ask you to confirm the operation:

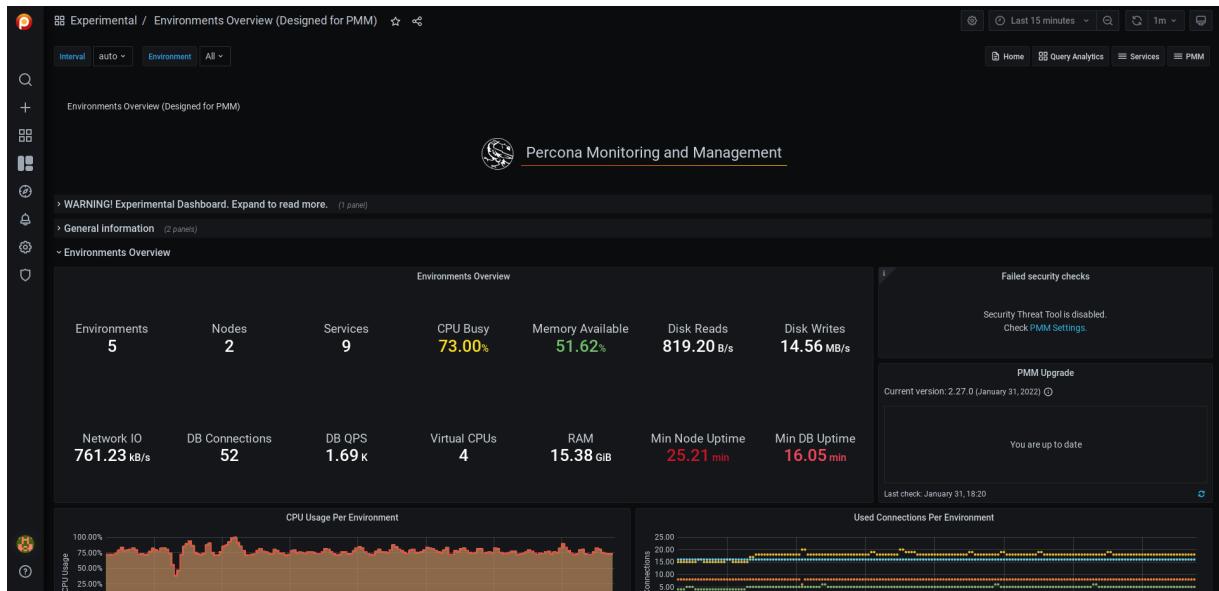
The screenshot shows the PMM Inventory list page with a modal dialog titled 'Confirm action'. The dialog asks 'Are you sure that you want to permanently delete 1 service?'. It includes a 'Force mode' checkbox with the subtext 'Force mode is going to delete all associated agents'. At the bottom are 'Cancel' and 'Proceed' buttons. The background table is identical to the one in the previous screenshot, showing the same two MongoDB services selected for deletion.

Last update: 2021-07-29

## Environment Overview

### ⚠ Disclaimer

This is an Experimental Dashboard that is not part of the official Percona Monitoring and Management (PMM) deployment and might be updated. We ship this Dashboard to obtain feedback from our users.



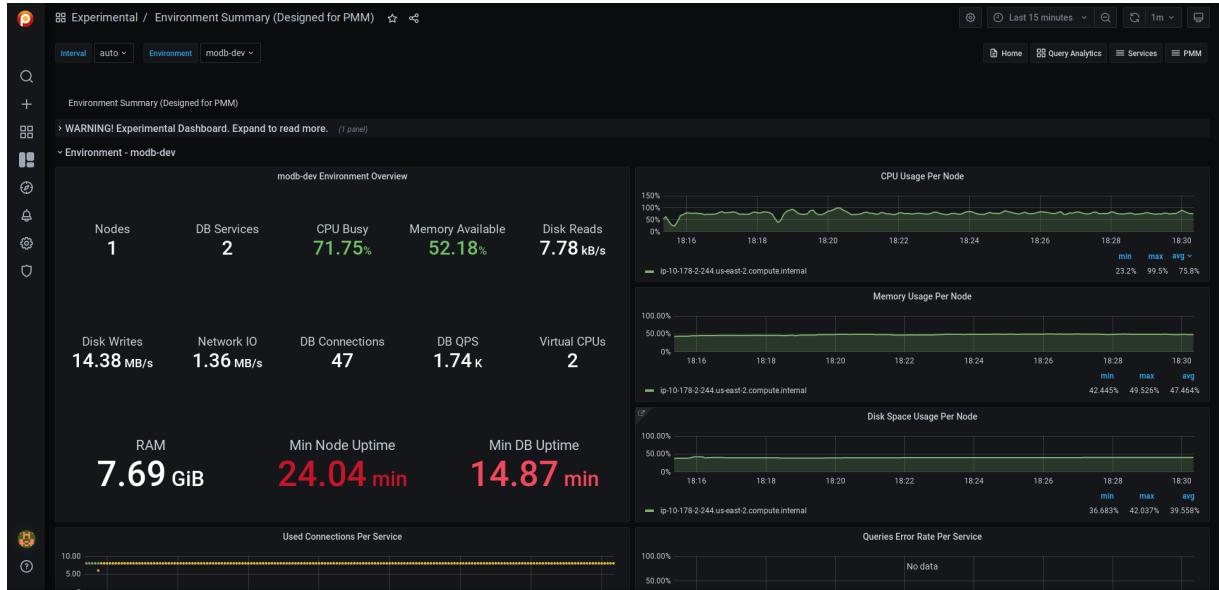
The Dashboard provides the user with a high-level view of all the environments in PMM.

Last update: 2022-02-08

## Environment Summary

### ⚠ Disclaimer

This is an Experimental Dashboard that is not part of the official Percona Monitoring and Management (PMM) deployment and might be updated. We ship this Dashboard to obtain feedback from our users.

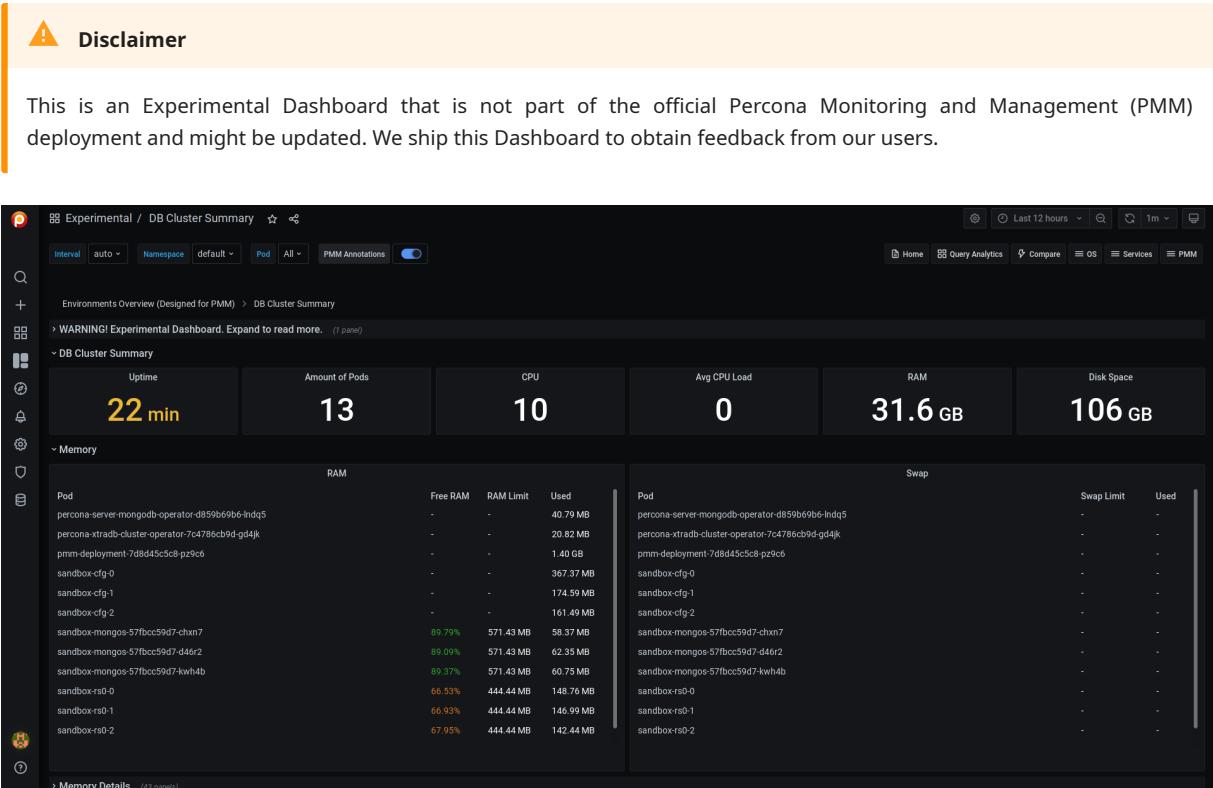


The Environment Summary Dashboard provides an at-a-glance view specific to the selected environment in PMM, including an overview of the services and nodes running in that environment.

Last update: 2022-02-08

## 5.4.4 DBaaS

### DB Cluster Summary



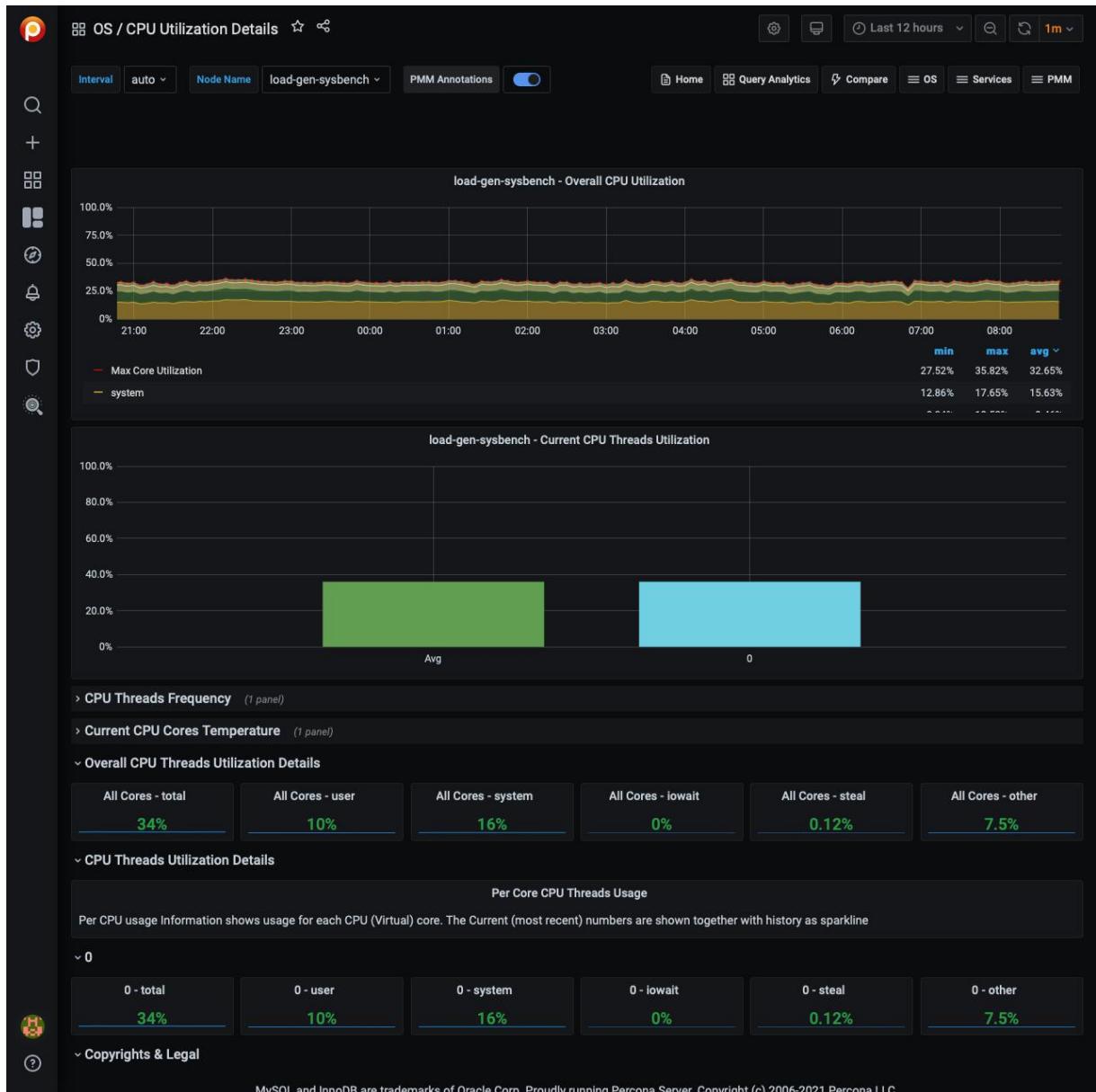
This Dashboard is a part of [DBaaS solution](#) inside PMM.

This Dashboard is designed to show the resource consumption inside K8s Cluster.

Last update: 2022-02-08

## 5.4.5 OS Dashboards

### CPU Utilization Details



#### OVERALL CPU UTILIZATION

The Overall CPU Utilization metric shows how much of the overall CPU time is used by the server. It has these components:

##### Max Core Utilization

No description

##### System

This component the proportion of time the CPUs spent inside the Linux kernel for operations like context switching, memory allocation and queue handling.

**User**

This component is the time spent in the user space. Normally, most of the MySQL CPU time is in user space. A high value of user time indicates a CPU bound workload.

**Softirq**

This component is the portion of time the CPU spent servicing software interrupts generated by the device drivers. A high value of softirq may indicate a poorly configured device. The network devices are generally the main source of high softirq values.

**Steal**

When multiple virtual machines share the same physical host, some virtual machines may be allowed to use more of their share of CPU and that CPU time is accounted as Steal by the virtual machine from which the time is taken.

**Iowait**

This component is the time the CPU spent waiting for disk IO requests to complete. A high value of iowait indicates a disk bound load.

**Nice**

No description

In addition, sampling of the Max utilization of a single core is shown.

This metric presents global values: while there may be a lot of unused CPU, a single core may be saturated. Look at the Max Core Utilization to see if any core is reaching close to 100%.

**CURRENT CPU THREADS UTILIZATION**

This shows the total utilization of each CPU core along with the average utilization of all CPU cores. Watch for any core close to 100% utilization and investigate the root cause.

**CPU THREADS FREQUENCY**

No description

**CURRENT CPU CORES TEMPERATURE**

No description

**OVERALL CPU THREADS UTILIZATION DETAILS**

No description

Last update: 2021-06-03

## Disk Details



### MOUNT POINT USAGE

Shows the percentage of disk space utilization for every mount point defined on the system. Having some of the mount points close to 100% space utilization is not good because of the risk of a “disk full” error that can block one of the services or even cause a crash of the entire system.

In cases where the mount point is close to 100% consider removing unused files or expanding the space allocated to the mount point.

### MOUNT POINT

Shows information about the disk space usage of the specified mount point.

**Used** is the amount of space used.

**Free** is the amount of space not in use.

**Used+Free** is the total disk space allocated to the mount point.

Having *Free* close to 0 B is not good because of the risk of a “disk full” error that can block one of the services or even cause a crash of the entire system.

In cases where *Free* is close to 0 B consider removing unused files or expanding the space allocated to the mount point.

#### DISK LATENCY

Shows average latency for Reads and Writes IO Devices. Higher than typical latency for highly loaded storage indicates saturation (overload) and is frequent cause of performance problems. Higher than normal latency also can indicate internal storage problems.

#### DISK OPERATIONS

Shows amount of physical IOs (reads and writes) different devices are serving. Spikes in number of IOs served often corresponds to performance problems due to IO subsystem overload.

#### DISK BANDWIDTH

Shows volume of reads and writes the storage is handling. This can be better measure of IO capacity usage for network attached and SSD storage as it is often bandwidth limited. Amount of data being written to the disk can be used to estimate Flash storage life time.

#### DISK LOAD

Shows how much disk was loaded for reads or writes as average number of outstanding requests at different period of time. High disk load is a good measure of actual storage utilization. Different storage types handle load differently - some will show latency increases on low loads others can handle higher load with no problems.

#### DISK IO UTILIZATION

Shows disk Utilization as percent of the time when there was at least one IO request in flight. It is designed to match utilization available in iostat tool. It is not very good measure of true IO Capacity Utilization. Consider looking at IO latency and Disk Load Graphs instead.

#### AVG DISKS OPERATIONS MERGE RATIO

Shows how effectively Operating System is able to merge logical IO requests into physical requests. This is a good measure of the IO locality which can be used for workload characterization.

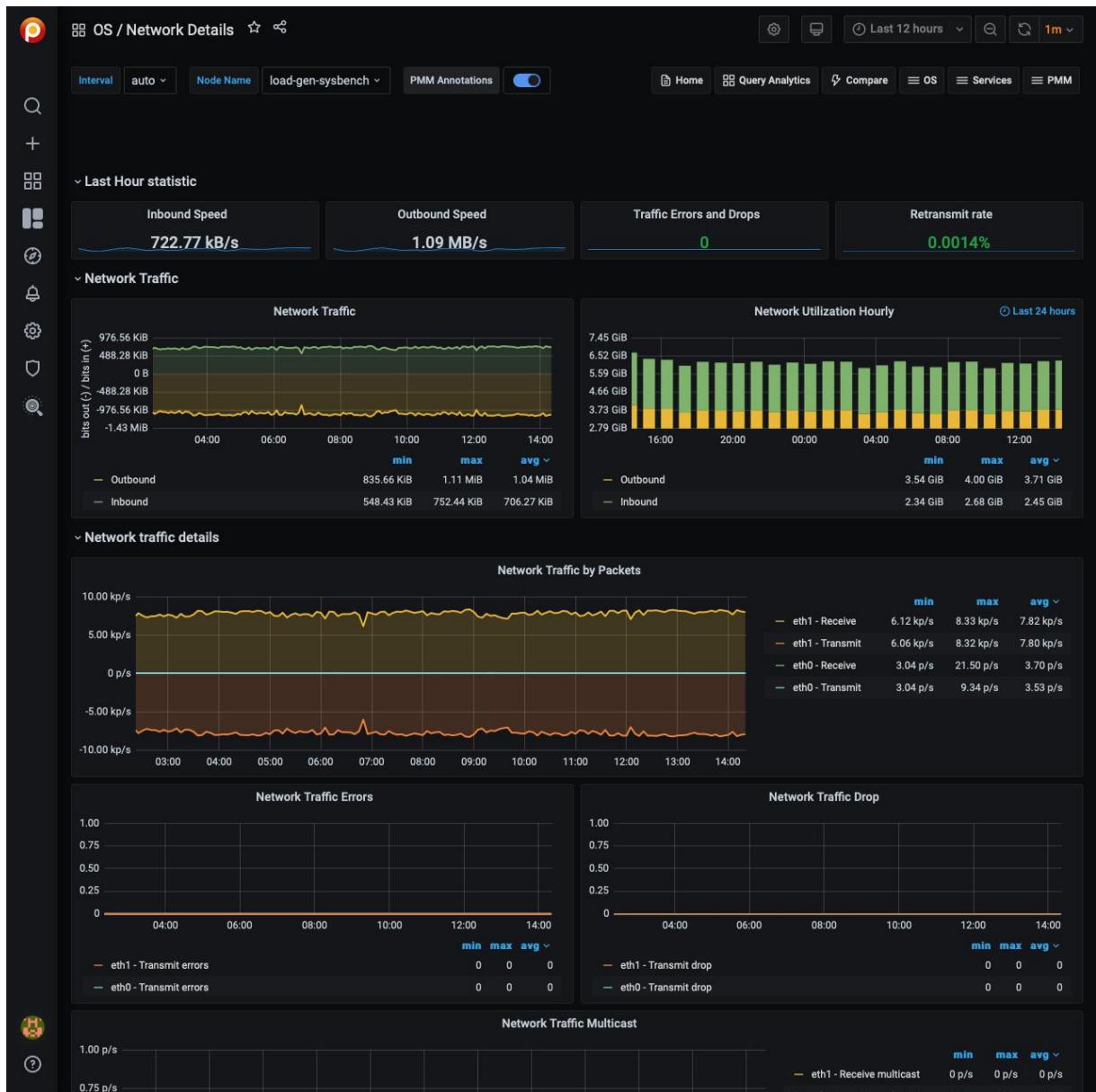
#### DISK IO SIZE

Shows average size of a single disk operation.

---

Last update: 2021-05-11

## Network Details



### LAST HOUR STATISTIC

This section reports the *inbound speed*, *outbound speed*, *traffic errors and drops*, and *retransmit rate*.

### NETWORK TRAFFIC

This section contains the *Network traffic* and *network utilization hourly* metrics.

### NETWORK TRAFFIC DETAILS

This section offers the following metrics:

- Network traffic by packets
- Network traffic errors
- Network traffic drop
- Network traffic multicast

**NETWORK NETSTAT TCP**

This section offers the following metrics:

- Timeout value used for retransmitting
- Min TCP retransmission timeout
- Max TCP retransmission timeout
- Netstat: TCP
- TCP segments

**NETWORK NETSTAT UDP**

In this section, you can find the following metrics:

- Netstat: UDP
- UDP Lite

The graphs in the *UDP Lite* metric give statistics about:

**InDatagrams**

Packets received

**OutDatagrams**

Packets sent

**InCsumErrors**

Datagrams with checksum errors

**InErrors**

Datagrams that could not be delivered to an application

**RcvbufErrors**

Datagrams for which not enough socket buffer memory to receive

**SndbufErrors**

Datagrams for which not enough socket buffer memory to transmit

**NoPorts**

Datagrams received on a port with no listener

**ICMP**

This section has the following metrics:

- ICMP Errors
- Messages/Redirects
- Echos
- Timestamps/Mask Requests

**ICMP Errors****InErrors**

Messages which the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.)

**OutErrors**

Messages which this entity did not send due to problems discovered within ICMP, such as a lack of buffers

**InDestUnreachs**

Destination Unreachable messages received

**OutDestUnreachs**

Destination Unreachable messages sent

**InType3**

Destination unreachable

**OutType3**

Destination unreachable

**InCsumErrors**

Messages with ICMP checksum errors

**InTimeExcds**

Time Exceeded messages received

**Messages/Redirects****InMsgs**

Messages which the entity received. Note that this counter includes all those counted by `icmpInErrors`

**InRedirects**

Redirect messages received

**OutMsgs**

Messages which this entity attempted to send. Note that this counter includes all those counted by `icmpOutErrors`

**OutRedirects**

Redirect messages sent. For a host, this object will always be zero, since hosts do not send redirects

**Echos****InEchoReps**

Echo Reply messages received

**InEchos**

Echo (request) messages received

**OutEchoReps**

Echo Reply messages sent

**OutEchos**

Echo (request) messages sent

**Timestamps/Mask Requests****InAddrMaskReps**

Address Mask Reply messages received

**InAddrMasks**

Address Mask Request messages received

**OutAddrMaskReps**

Address Mask Reply messages sent

**OutAddrMasks**

Address Mask Request messages sent

**InTimestampReps**

Timestamp Reply messages received

**InTimestamps**

Timestamp Request messages received

**OutTimestampReps**

Timestamp Reply messages sent

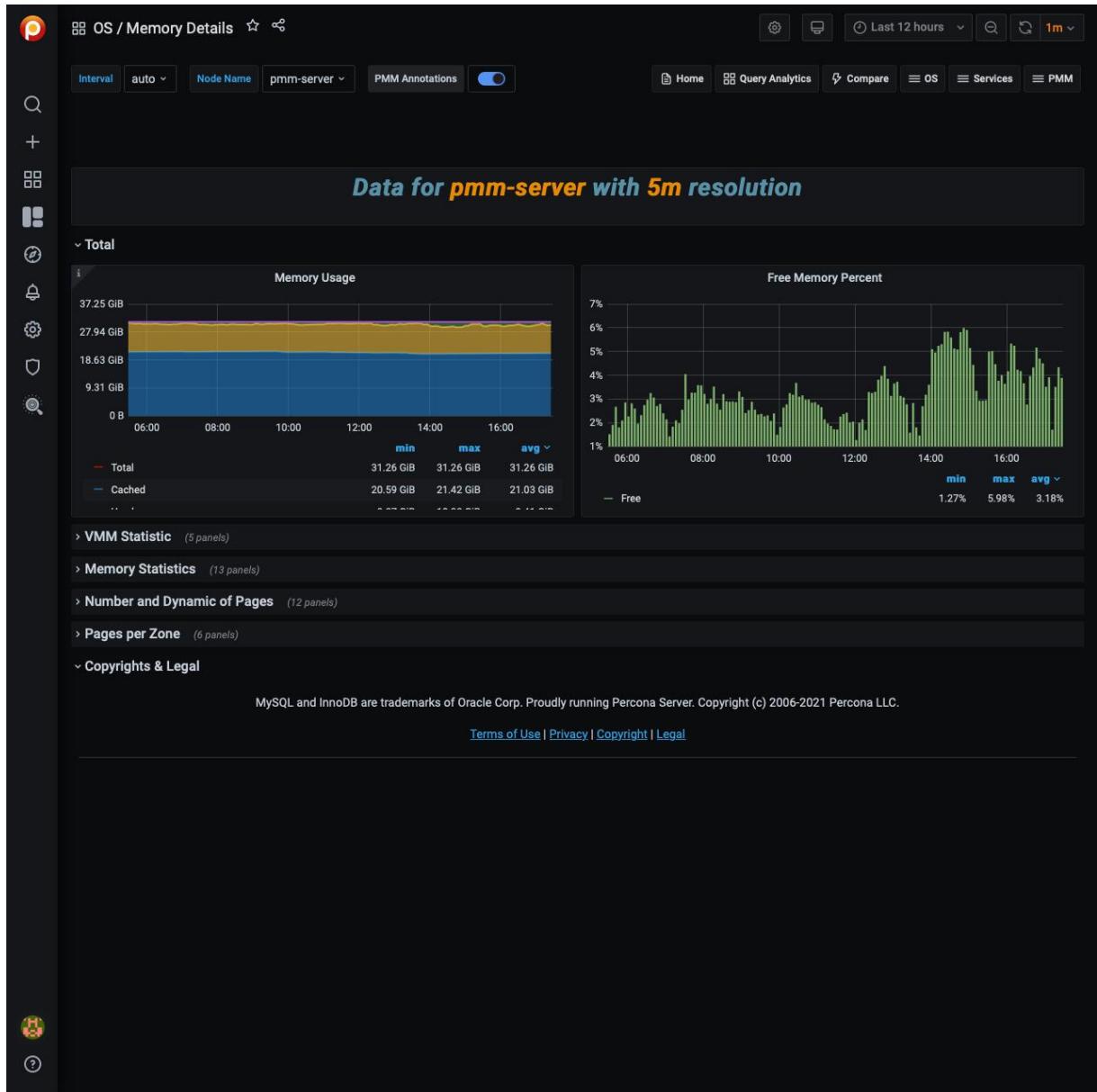
**OutTimestamps**

Timestamp Request messages sent

---

Last update: 2021-06-25

## Memory Details

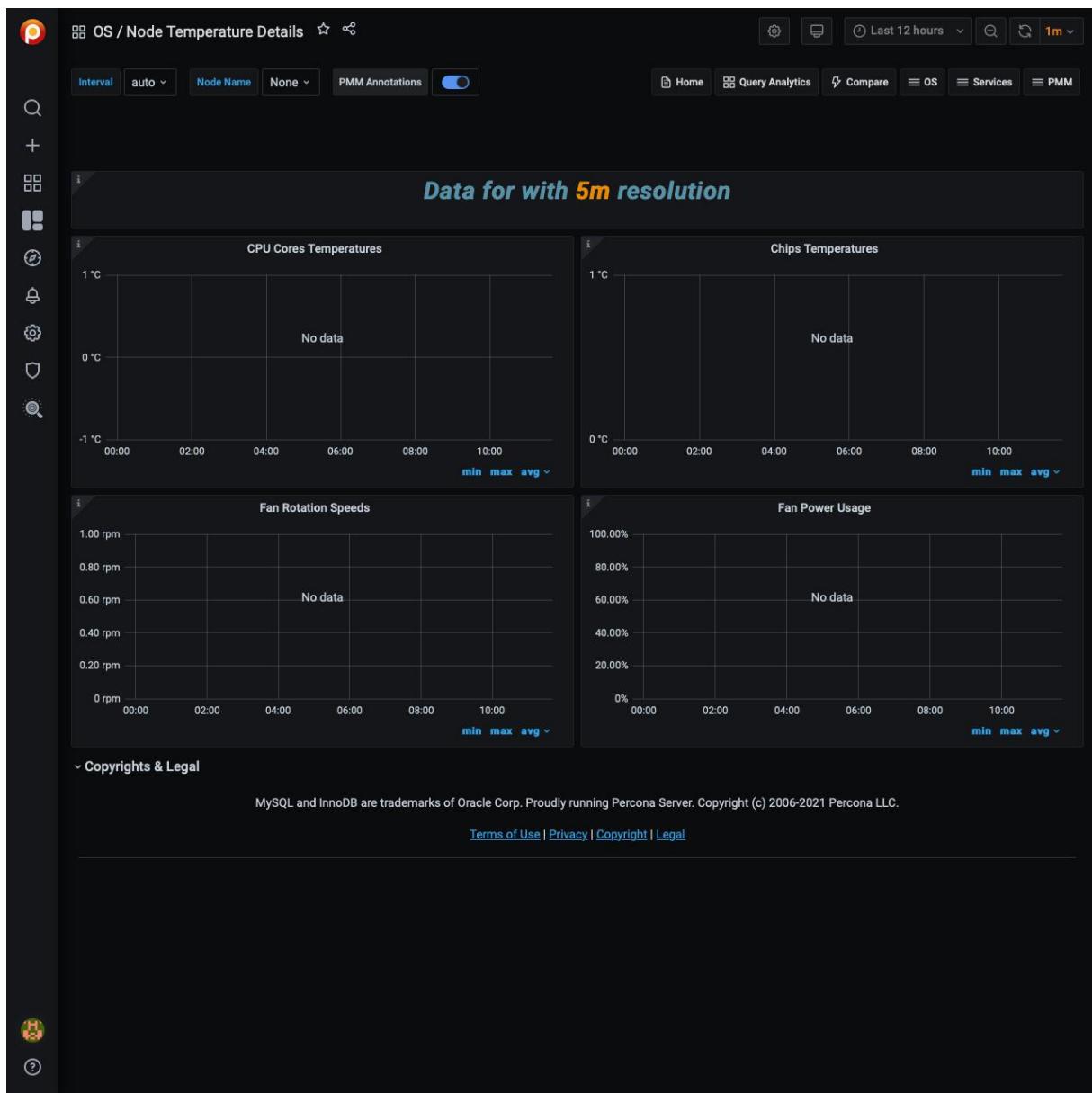


## MEMORY USAGE

No description

Last update: 2021-05-31

## Node Temperature Details



The Node Temperature Details dashboard exposes hardware monitoring and sensor data obtained through the `sysfs` virtual file system of the node.

Hardware monitoring devices attached to the CPU and/or other chips on the motherboard let you monitor the hardware health of a system. Most modern systems include several of such devices. The actual list can include temperature sensors, voltage sensors, fan speed sensors, and various additional features, such as the ability to control the rotation speed of the fans.

### CPU CORES TEMPERATURES

Presents data taken from the temperature sensors of the CPU

### CHIPS TEMPERATURES

Presents data taken from the temperature sensors connected to other system controllers

**FAN ROTATION SPEEDS**

Fan rotation speeds reported in RPM (rotations per minute).

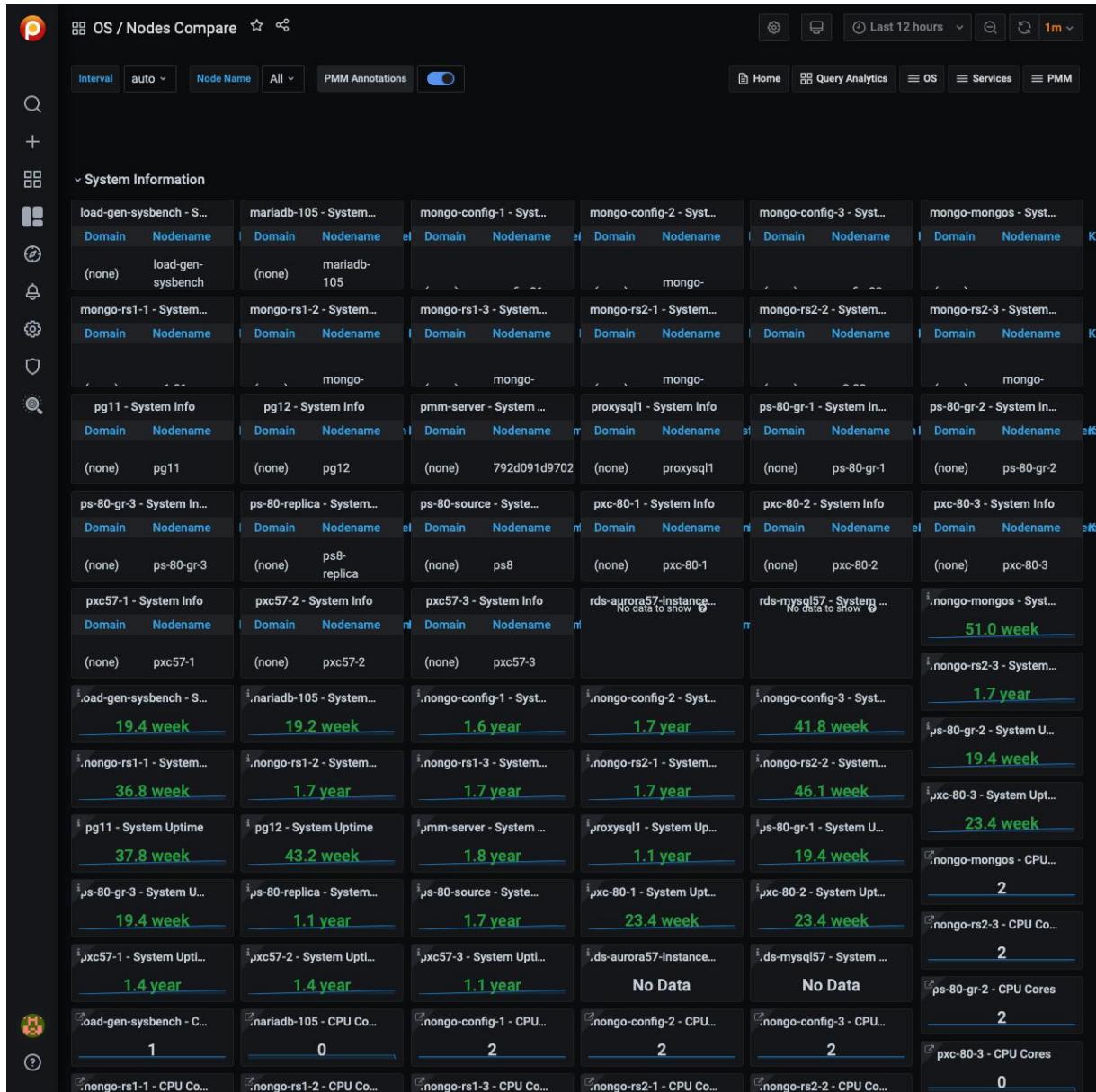
**FAN POWER USAGE**

Describes the pulse width modulation of the PWN-equipped fans. PWM operates like a switch that constantly cycles on and off, thereby regulating the amount of power the fan gains: 100% makes it rotate at full speed, while lower percentage slows rotation down proportionally.

---

Last update: 2021-06-03

## Nodes Compare



This dashboard lets you compare a wide range of parameters. Parameters of the same type are shown side by side for all servers, grouped into the following sections:

- System Information
- CPU
- Memory
- Disk Partitions
- Disk Performance
- Network

The *System Information* section shows the *System Info* summary of each server, as well as *System Uptime*, *CPU Cores*, *RAM*, *Saturation Metrics*, and *Load Average* gauges.

The *CPU* section offers the *CPU Usage*, *Interrupts*, and *Context Switches* metrics.

In the *Memory* section, you can find the *Memory Usage*, *Swap Usage*, and *Swap Activity* metrics.

The *Disk Partitions* section encapsulates two metrics, *Mountpoint Usage* and *Free Space*.

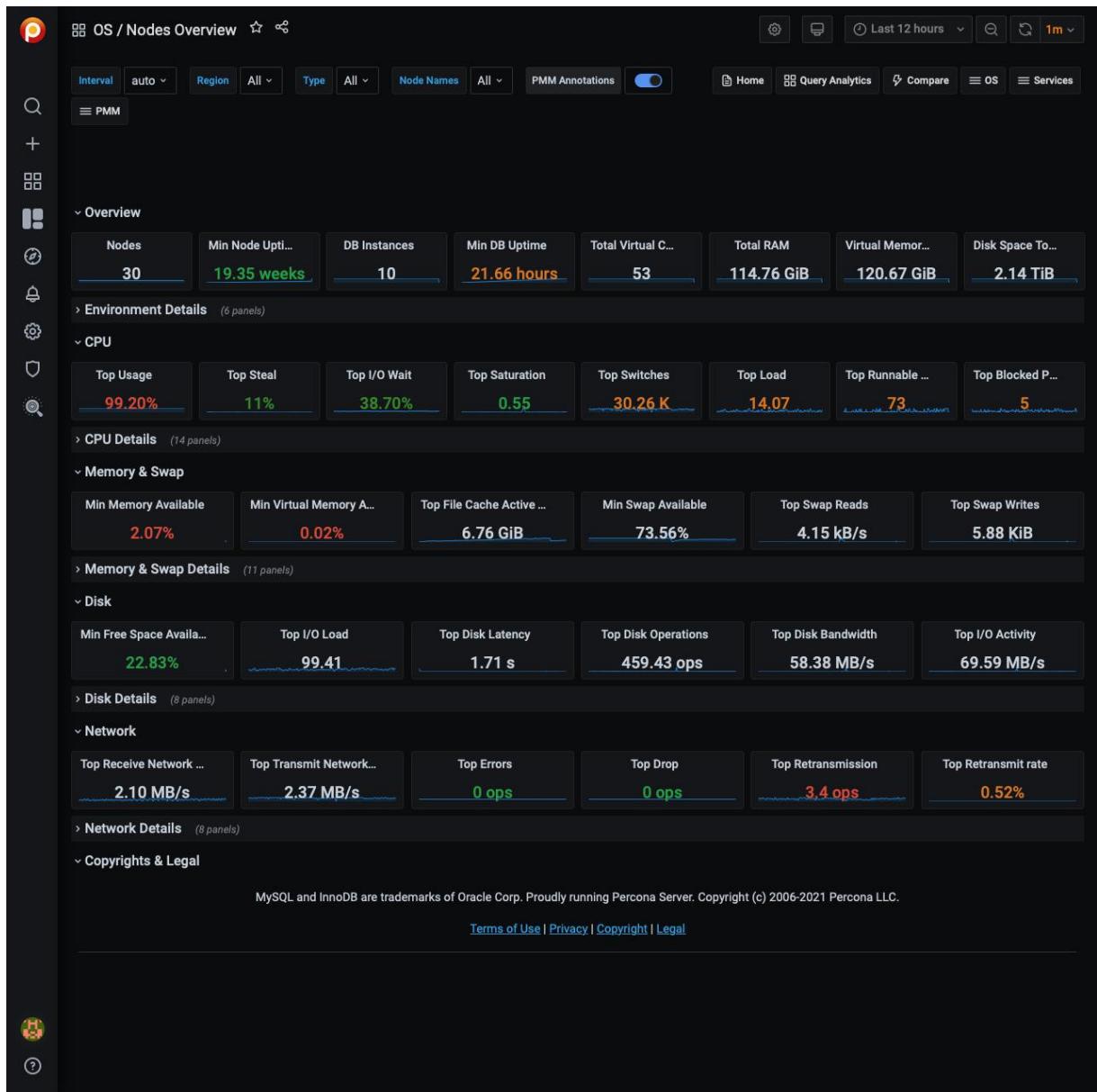
The *Disk Performance* section contains the *I/O Activity*, *Disk Operations*, *Disk Bandwidth*, *Disk IO Utilization*, *Disk Latency*, and *Disk Load* metrics.

Finally, *Network* section shows *Network Traffic*, and *Network Utilization Hourly* metrics.

---

Last update: 2021-05-11

## Nodes Overview



The Nodes Overview dashboard provides details about the efficiency of work of the following components. Each component is represented as a section in the dashboard.

- CPU
- Memory & Swap
- Disk
- Network

The *CPU* section offers the *CPU Usage*, *CPU Saturation* and *Max Core Usage*, *Interrupts* and *Context Switches*, and *Processes* metrics.

In the *Memory* section, you can find the *Memory Utilization*, *Virtual Memory Utilization*, *Swap Space*, and *Swap Activity* metrics.

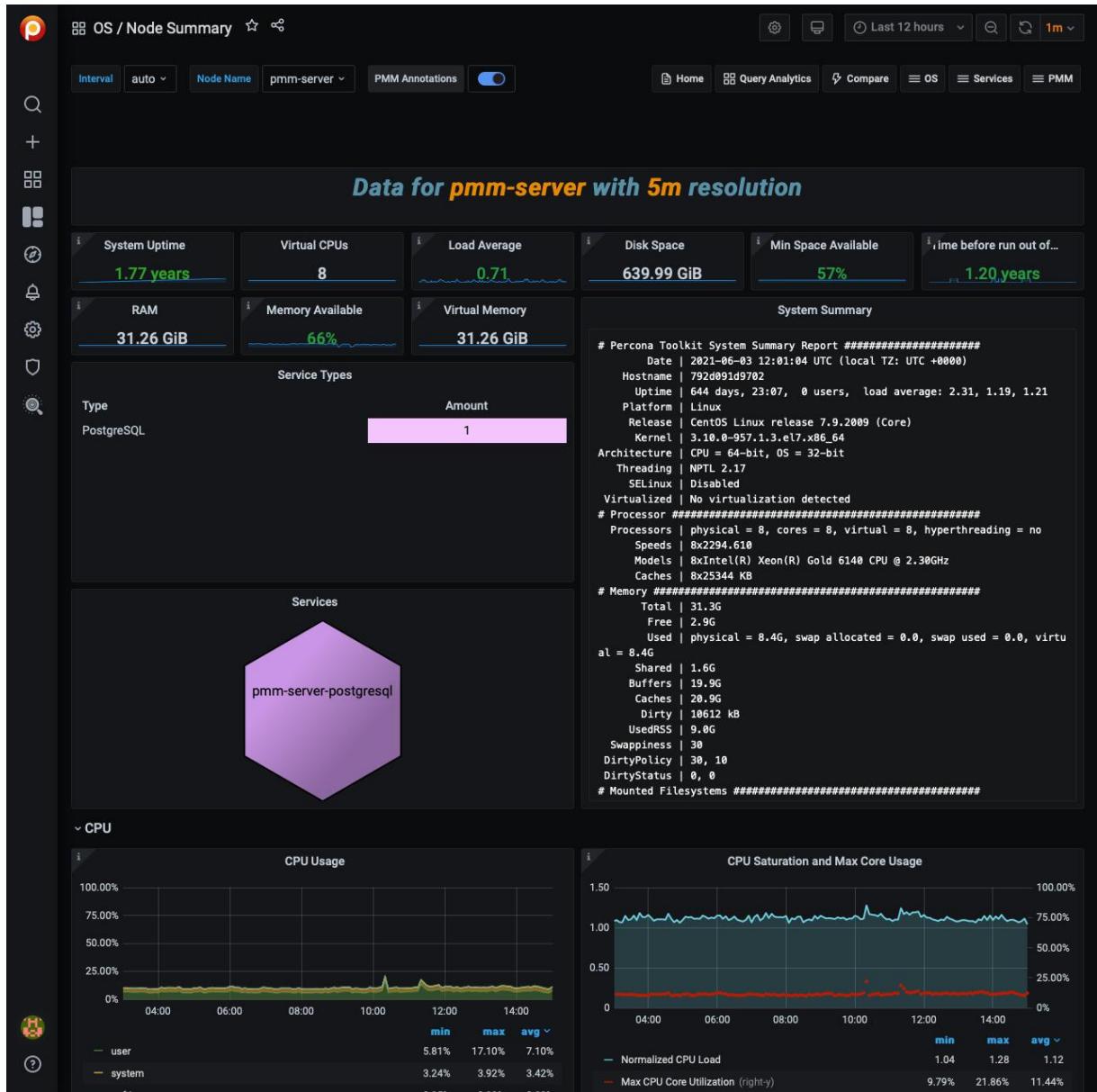
The *Disk* section contains the *I/O Activity*, *Global File Descriptors Usage*, *Disk IO Latency*, and *Disk IO Load* metrics.

In the *Network* section, you can find the *Network Traffic*, *Network Utilization Hourly*, *Local Network Errors*, and *TCP Retransmission* metrics.

---

Last update: 2021-06-25

## Node Summary



## SYSTEM SUMMARY

The output from `pt-summary`, one of the [Percona Toolkit utilities](#).

### CPU USAGE

The CPU time is measured in clock ticks or seconds. It is useful to measure CPU time as a percentage of the CPU's capacity, which is called the CPU usage.

### CPU SATURATION AND MAX CORE USAGE

When a system is running with maximum CPU utilization, the transmitting and receiving threads must all share the available CPU. This will cause data to be queued more frequently to cope with the lack of CPU. CPU Saturation may be measured as the length of a wait queue, or the time spent waiting on the queue.

**INTERRUPTS AND CONTEXT SWITCHES**

Interrupt is an input signal to the processor indicating an event that needs immediate attention. An interrupt signal alerts the processor and serves as a request for the processor to interrupt the currently executing code, so that the event can be processed in a timely manner.

Context switch is the process of storing the state of a process or thread, so that it can be restored and resume execution at a later point. This allows multiple processes to share a single CPU, and is an essential feature of a multitasking operating system.

**PROCESSES**

No description

**MEMORY UTILIZATION**

No description

**VIRTUAL MEMORY UTILIZATION**

No description

**SWAP SPACE**

No description

**SWAP ACTIVITY**

Swap Activity is memory management that involves swapping sections of memory to and from physical storage.

**I/O ACTIVITY**

Disk I/O includes read or write or input/output operations involving a physical disk. It is the speed with which the data transfer takes place between the hard disk drive and RAM.

**GLOBAL FILE DESCRIPTORS USAGE**

No description

**DISK IO LATENCY**

Shows average latency for Reads and Writes IO Devices. Higher than typical latency for highly loaded storage indicates saturation (overload) and is frequent cause of performance problems. Higher than normal latency also can indicate internal storage problems.

**DISK IO LOAD**

Shows how much disk was loaded for reads or writes as average number of outstanding requests at different period of time. High disk load is a good measure of actual storage utilization. Different storage types handle load differently - some will show latency increases on low loads others can handle higher load with no problems.

**NETWORK TRAFFIC**

Network traffic refers to the amount of data moving across a network at a given point in time.

**NETWORK UTILIZATION HOURLY**

No description

**LOCAL NETWORK ERRORS**

Total Number of Local Network Interface Transmit Errors, Receive Errors and Drops. Should be Zero

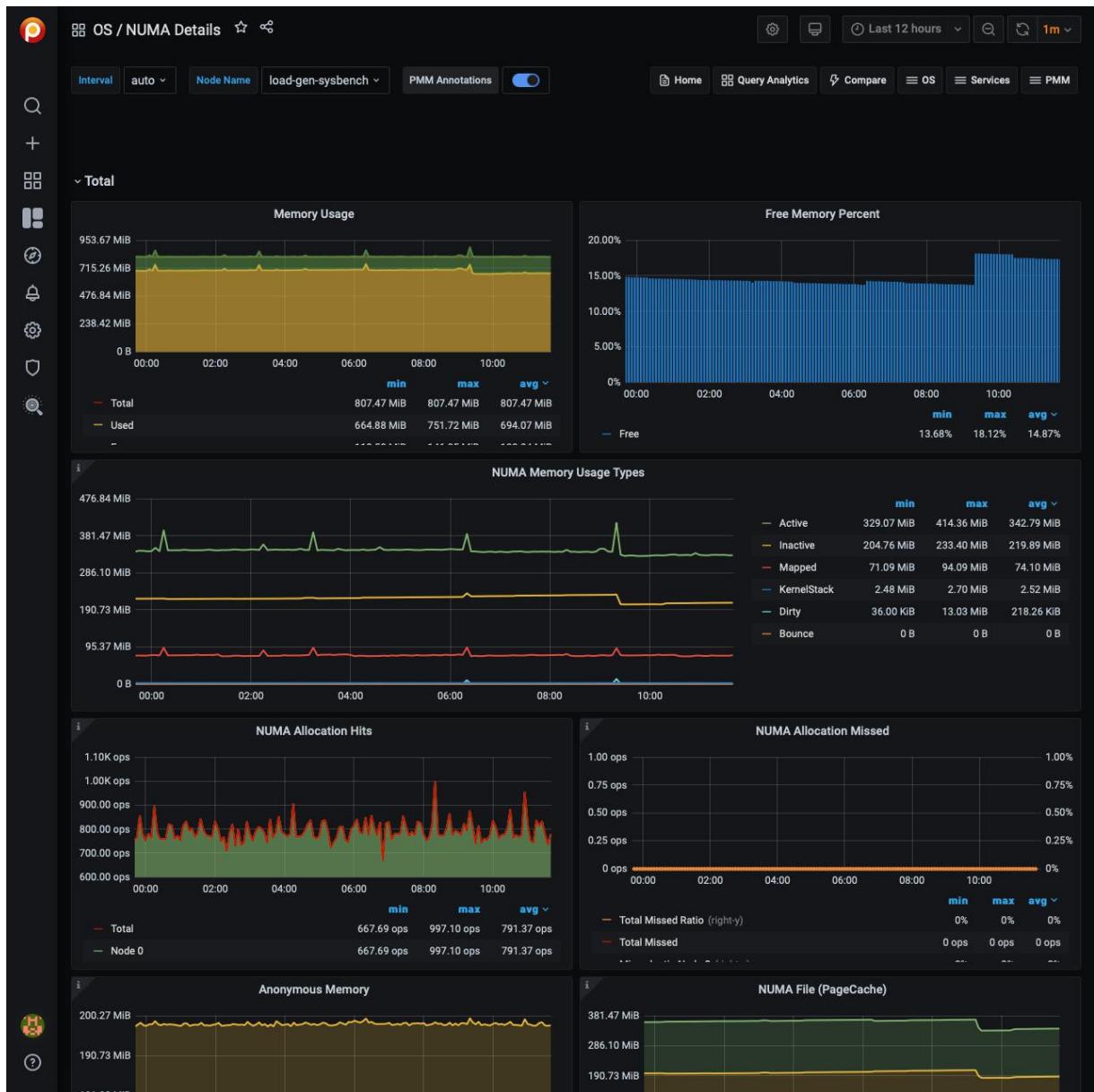
**TCP RETRANSMISSION**

Retransmission, essentially identical with Automatic repeat request (ARQ), is the resending of packets which have been either damaged or lost. Retransmission is one of the basic mechanisms used by protocols operating over a packet switched computer network to provide reliable communication (such as that provided by a reliable byte stream, for example TCP).

---

Last update: 2021-05-11

## NUMA Details



For each node, this dashboard shows metrics related to Non-uniform memory access (NUMA).

### MEMORY USAGE

Monitors over time the total, used, and free memory.

### FREE MEMORY PERCENT

Shows the free memory as the ratio to the total available memory.

### NUMA MEMORY USAGE TYPES

#### Dirty

Memory waiting to be written back to disk

#### Bounce

Memory used for block device bounce buffers

**Mapped**

Files which have been mapped, such as libraries

**KernelStack**

The memory the kernel stack uses. This is not reclaimable.

**NUMA ALLOCATION HITS**

Memory successfully allocated on this node as intended.

**NUMA ALLOCATION MISSED**

Memory missed is allocated on a node despite the process preferring some different node.

Memory foreign is intended for a node, but actually allocated on some different node.

**ANONYMOUS MEMORY**

## Active

Anonymous memory that has been used more recently and usually not swapped out.

## Inactive

Anonymous memory that has not been used recently and can be swapped out.

**NUMA FILE (PAGECACHE)**

Active(file) Pagecache memory that has been used more recently and usually not reclaimed until needed.

Inactive(file) Pagecache memory that can be reclaimed without huge performance impact.

**SHARED MEMORY**

Shmem Total used shared memory (shared between several processes, thus including RAM disks, SYS-V-IPC and BSD like SHMEM).

**HUGEPAGES STATISTICS**

## Total

Number of hugepages being allocated by the kernel (Defined with `vm.nr_hugepages` ).

## Free

The number of hugepages not being allocated by a process

**Surp**

The number of hugepages in the pool above the value in `vm.nr_hugepages`. The maximum number of surplus hugepages is controlled by `vm.nr_overcommit_hugepages`.

**LOCAL PROCESSES**

Memory allocated on a node while a process was running on it.

**REMOTE PROCESSES**

Memory allocated on a node while a process was running on some other node.

**SLAB MEMORY****Slab**

Allocation is a memory management mechanism intended for the efficient memory allocation of kernel objects.

**SReclaimable**

The part of the Slab that might be reclaimed (such as caches).

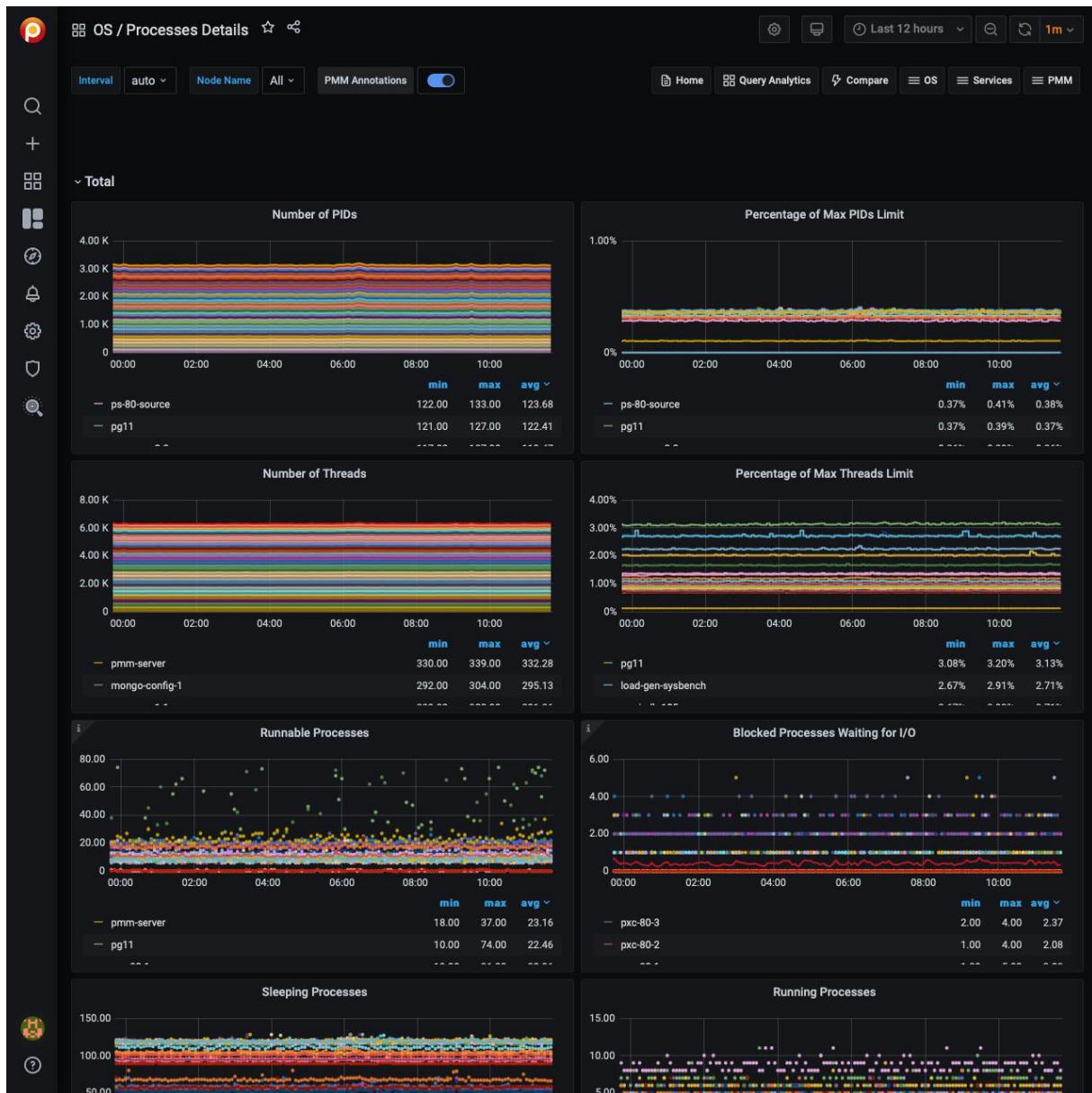
**SUnreclaim**

The part of the Slab that can't be reclaimed under memory pressure

---

Last update: 2021-05-11

## Processes Details



The Processes Details dashboard displays Linux process information - PIDs, Threads, and Processes. The dashboard shows how many processes/threads are either in the kernel run queue (runnable state) or in the blocked queue (waiting for I/O). When the number of process in the runnable state is constantly higher than the number of CPU cores available, the load is CPU bound. When the number of process blocked waiting for I/O is large, the load is disk bound. The running average of the sum of these two quantities is the basis of the `loadavg` metric.

The dashboard consists of two parts: the first section describes metrics for all hosts, and the second part provides charts for each host.

Charts for all hosts, available in the first section, are the following ones:

- States of Processes
- Number of PIDs
- Percentage of Max PIDs Limit
- Number of Threads
- Percentage of Max Threads Limit
- Runnable Processes
- Blocked Processes Waiting for I/O
- Sleeping Processes
- Running Processes
- Disk Sleep Processes
- Stopped Processes
- Zombie Processes
- Dead Processes

The following charts are present in the second part, available for each host:

- Processes
- States of Processes
- Number of PIDs
- Percentage of Max PIDs Limit
- Number of Threads
- Percentage of Max Threads Limit

#### NUMBER OF PIDS

No description

#### PERCENTAGE OF MAX PIDS LIMIT

No description

#### NUMBER OF THREADS

No description

#### PERCENTAGE OF MAX THREADS LIMIT

No description

#### RUNNABLE PROCESSES

##### Processes

The Processes graph shows how many processes/threads are either in the kernel run queue (runnable state) or in the blocked queue (waiting for I/O). When the number of process in the runnable state is constantly higher than the number of CPU cores available, the load is CPU bound. When the number of process blocked waiting for I/O is large, the load is disk bound. The running average of the sum of these two quantities is the basis of the `loadavg` metric.

#### BLOCKED PROCESSES WAITING FOR I/O

##### Processes

The Processes graph shows how many processes/threads are either in the kernel run queue (runnable state) or in the blocked queue (waiting for I/O). When the number of process in the runnable state is constantly higher than the

number of CPU cores available, the load is CPU bound. When the number of process blocked waiting for I/O is large, the load is disk bound. The running average of the sum of these two quantities is the basis of the `loadavg` metric.

**SLEEPING PROCESSES**

No description

**RUNNING PROCESSES**

No description

**DISK SLEEP PROCESSES**

No description

**STOPPED PROCESSES**

No description

**ZOMBIE PROCESSES**

No description

**DEAD PROCESSES**

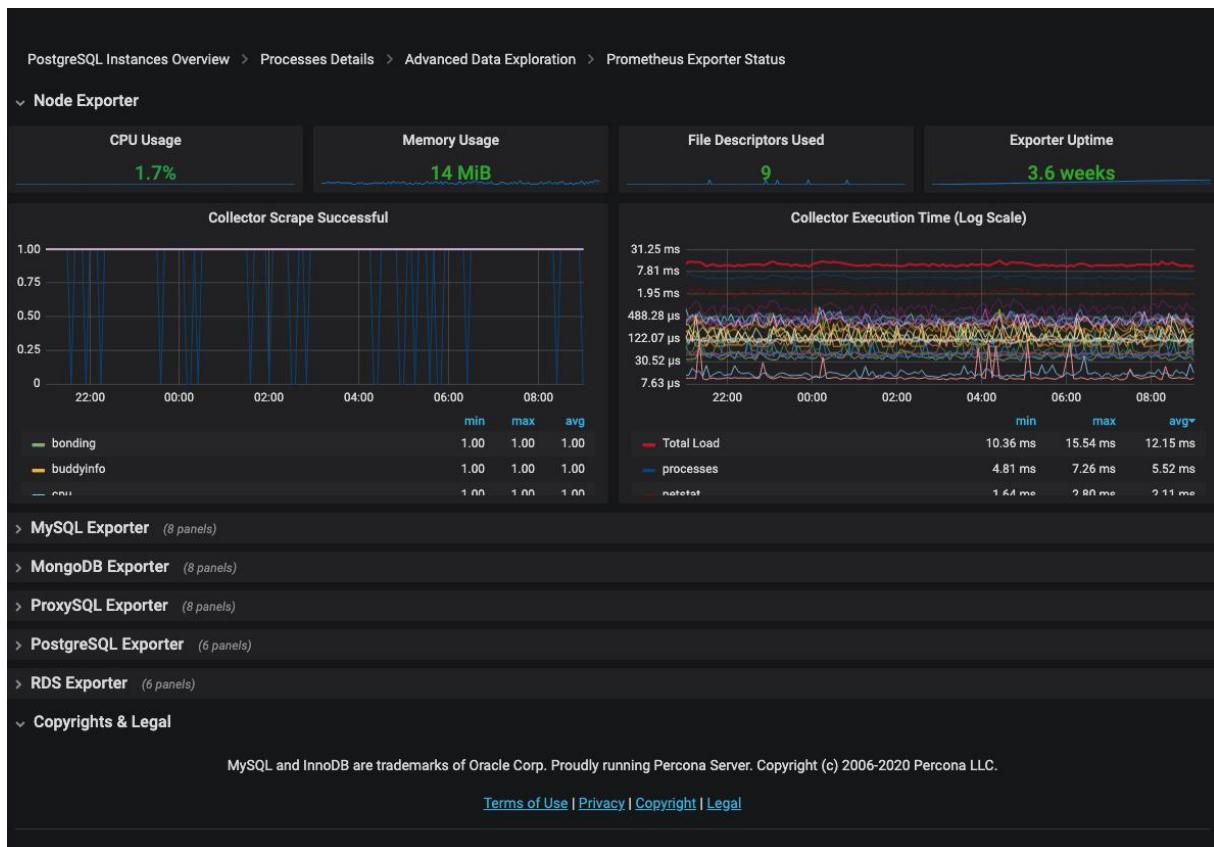
No description

---

Last update: 2021-06-25

## 5.4.6 Prometheus Dashboards

### Prometheus Exporter Status

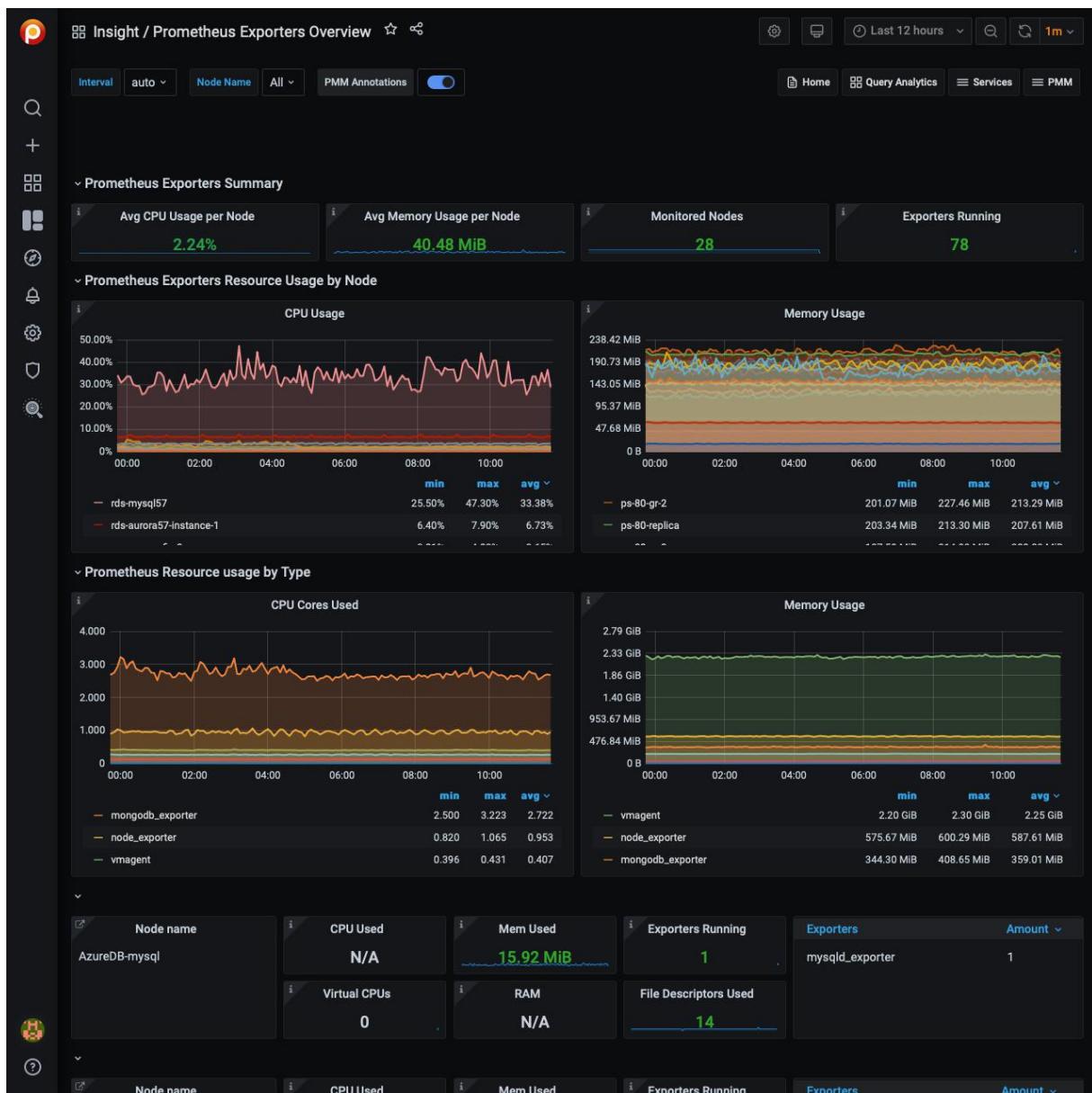


The Prometheus Exporter Status dashboard reports the consumption of resources by the Prometheus exporters used by PMM. For each exporter, this dashboard reveals the following information:

- CPU usage
- Memory usage
- File descriptors used
- Exporter uptime

Last update: 2021-06-25

## Prometheus Exporters Overview



### PROMETHEUS EXPORTERS SUMMARY

This section provides a summary of how exporters are used across the selected hosts. It includes the average usage of CPU and memory as well as the number of hosts being monitored and the total number of running exporters.

#### Avg CPU Usage per Host

Shows the average CPU usage in percent per host for all exporters.

#### Avg Memory Usage per Host

Shows the Exporters average Memory usage per host.

#### Monitored Hosts

Shows the number of monitored hosts that are running Exporters.

## Exporters Running

Shows the total number of Exporters running with this PMM Server instance.

The CPU usage and memory usage do not include the additional CPU and memory usage required to produce metrics by the application or operating system.

### PROMETHEUS EXPORTERS RESOURCE USAGE BY NODE

This section shows how resources, such as CPU and memory, are being used by the exporters for the selected hosts.

#### CPU Usage

Plots the Exporters' CPU usage across each monitored host (by default, All hosts).

#### Memory Usage

Plots the Exporters' Memory usage across each monitored host (by default, All hosts).

### PROMETHEUS EXPORTERS RESOURCE USAGE BY TYPE

This section shows how resources, such as CPU and memory, are being used by the exporters for host types: MySQL, MongoDB, ProxySQL, and the system.

#### CPU Cores Used

Shows the Exporters' CPU Cores used for each type of Exporter.

#### Memory Usage

Shows the Exporters' memory used for each type of Exporter.

### LIST OF HOSTS

At the bottom, this dashboard shows details for each running host.

#### CPU Used

Show the CPU usage as a percentage for all Exporters.

#### Mem Used

Shows total Memory Used by Exporters.

#### Exporters Running

Shows the number of Exporters running.

#### RAM

Shows the total amount of RAM of the host.

#### Virtual CPUs

Shows the total number of virtual CPUs on the host.

You can click the value of the *CPU Used*, *Memory Used*, or *Exporters Running* columns to open the [Prometheus Exporter Status](#) dashboard for further analysis.



#### See also

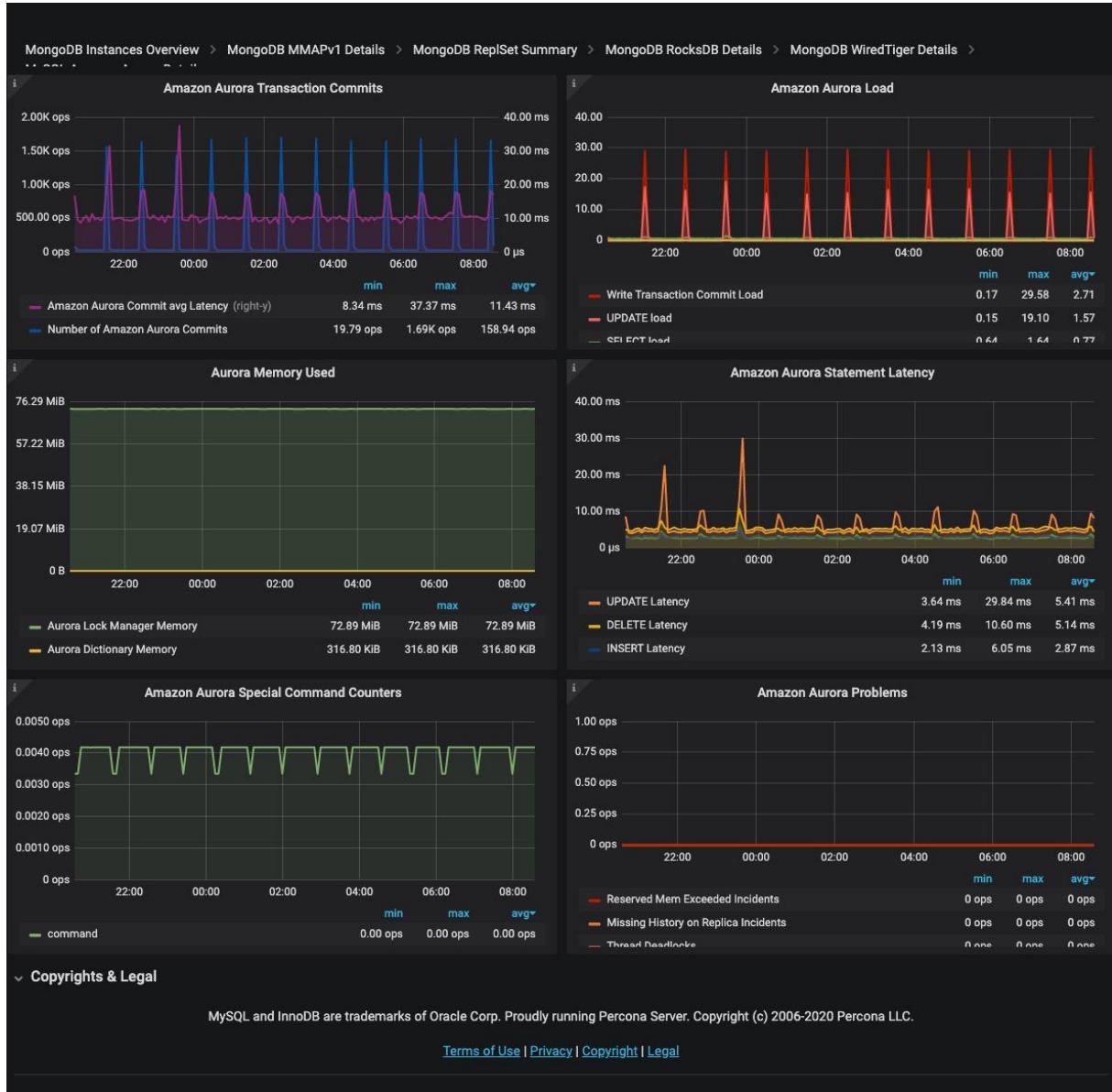
[Percona blog: Understand Your Prometheus Exporters with Percona Monitoring and Management \(PMM\)](#)

---

Last update: 2021-06-10

## 5.4.7 MySQL Dashboards

### MySQL Amazon Aurora Details



#### AMAZON AURORA TRANSACTION COMMITS

This graph shows the number of Commits which Amazon Aurora engine performed as well as average commit latency. Graph Latency does not always correlate with the number of performed commits and can be quite high in certain situations.

- **Number of Amazon Aurora Commits:** The average number of commit operations per second.
- **Amazon Aurora Commit avg Latency:** The average amount of latency for commit operations

#### AMAZON AURORA LOAD

This graph shows us what statements contribute most load on the system as well as what load corresponds to Amazon Aurora transaction commit.

- **Write Transaction Commit Load:** Load in Average Active Sessions per second for COMMIT operations
- **UPDATE load:** Load in Average Active Sessions per second for UPDATE queries
- **SELECT load:** Load in Average Active Sessions per second for SELECT queries
- **DELETE load:** Load in Average Active Sessions per second for DELETE queries
- **INSERT load:** Load in Average Active Sessions per second for INSERT queries

An *active session* is a connection that has submitted work to the database engine and is waiting for a response from it. For example, if you submit an SQL query to the database engine, the database session is active while the database engine is processing that query.

#### AURORA MEMORY USED

This graph shows how much memory is used by Amazon Aurora lock manager as well as amount of memory used by Amazon Aurora to store Data Dictionary.

- **Aurora Lock Manager Memory:** the amount of memory used by the Lock Manager, the module responsible for handling row lock requests for concurrent transactions.
- **Aurora Dictionary Memory:** the amount of memory used by the Dictionary, the space that contains metadata used to keep track of database objects, such as tables and indexes.

#### AMAZON AURORA STATEMENT LATENCY

This graph shows average latency for the most important types of statements. Latency spikes are often indicative of the instance overload.

- **DDL Latency:** Average time to execute DDL queries
- **DELETE Latency:** Average time to execute DELETE queries
- **UPDATE Latency:** Average time to execute UPDATE queries
- **SELECT Latency:** Average time to execute SELECT queries
- **INSERT Latency:** Average time to execute INSERT queries

#### AMAZON AURORA SPECIAL COMMAND COUNTERS

Amazon Aurora MySQL allows a number of commands which are not available in standard MySQL. This graph shows usage of such commands. Regular `unit_test` calls can be seen in default Amazon Aurora install, the rest will depend on your workload.

- `show_volume_status` : The number of executions per second of the command SHOW VOLUME STATUS. The SHOW VOLUME STATUS query returns two server status variables, Disks and Nodes. These variables represent the total number of logical blocks of data and storage nodes, respectively, for the DB cluster volume.
- `awslambda` : The number of AWS Lambda calls per second. AWS Lambda is an event-drive, server-less computing platform provided by AWS. It is a compute service that runs codes in response to an event. You can run any kind of code from Aurora invoking Lambda from a stored procedure or a trigger.
- `alter_system` : The number of executions per second of the special query ALTER SYSTEM, that is a special query to simulate an instance crash, a disk failure, a disk congestion or a replica failure. It's a useful query for testing the system.

#### AMAZON AURORA PROBLEMS

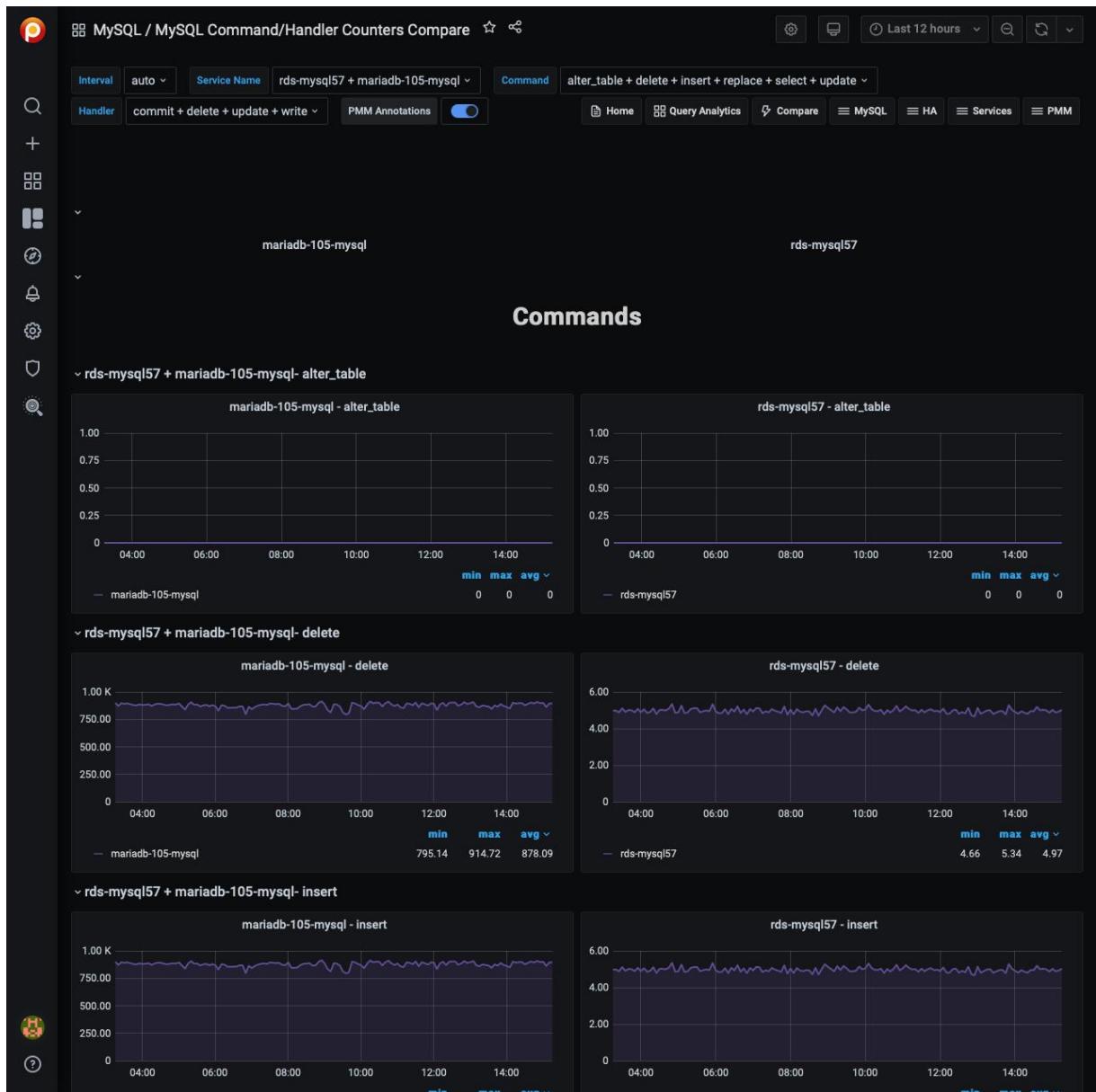
This graph shows different kinds of Internal Amazon Aurora MySQL Problems which generally should be zero in normal operation.

Anything non-zero is worth examining in greater depth.

---

Last update: 2021-05-11

## MySQL Command/Handler Counters Compare



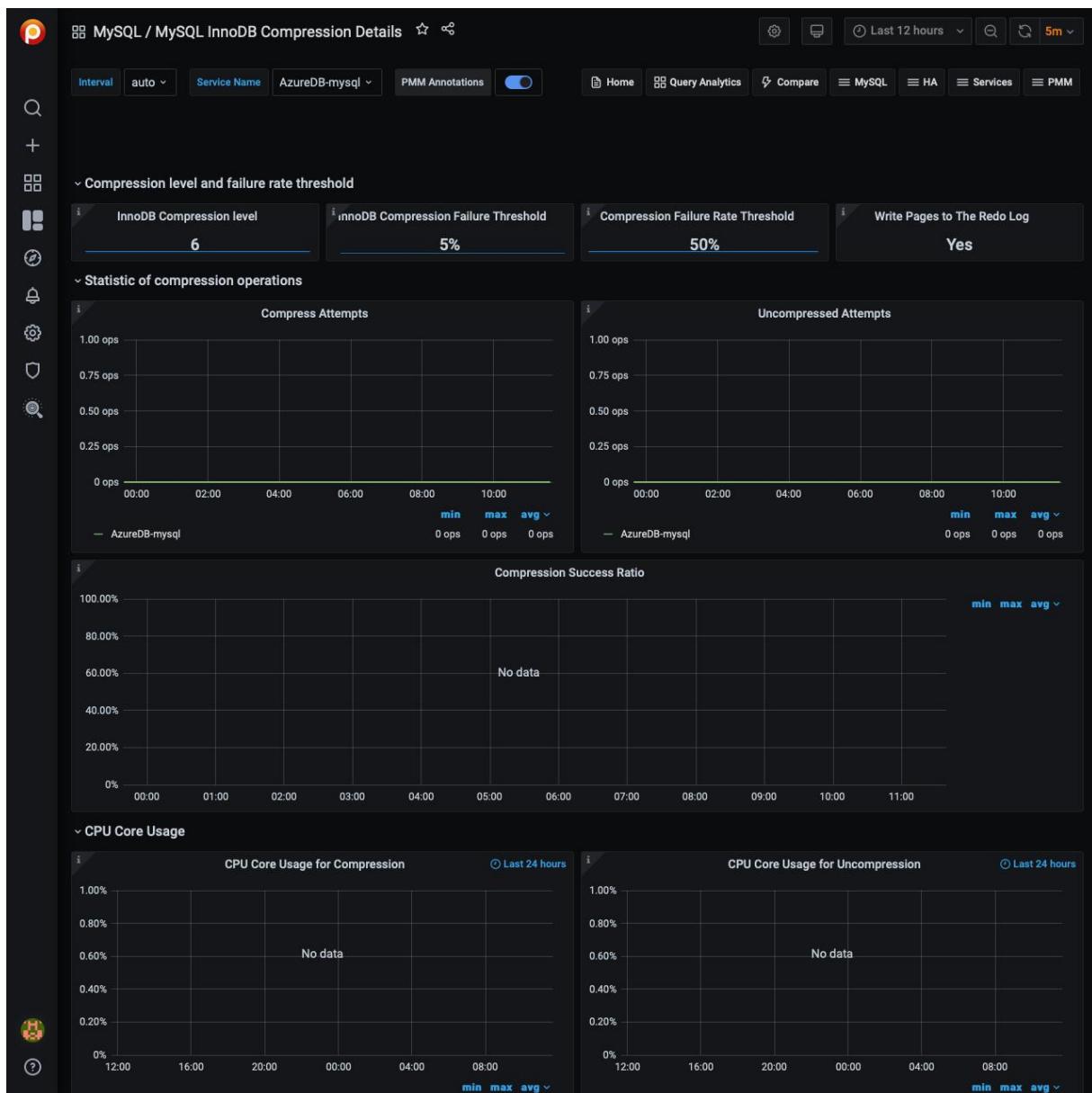
This dashboard shows server status variables. On this dashboard, you may select multiple servers and compare their counters simultaneously.

Server status variables appear in two sections: *Commands* and *Handlers*. Choose one or more variables in the *Command* and *Handler* fields in the top menu to select the variables which will appear in the *COMMANDS* or *HANDLERS* section for each host. Your comparison may include from one up to three hosts.

By default or if no item is selected in the menu, PMM displays each command or handler respectively.

Last update: 2021-05-11

## MySQL InnoDB Compression Details



This dashboard helps you analyze the efficiency of InnoDB compression.

### COMPRESSION LEVEL AND FAILURE RATE THRESHOLD

#### InnoDB Compression Level

The level of zlib compression to use for InnoDB compressed tables and indexes.

#### InnoDB Compression Failure Threshold

The compression failure rate threshold for a table.

#### Compression Failure Rate Threshold

The maximum percentage that can be reserved as free space within each compressed page, allowing room to reorganize the data and modification log within the page when a compressed table or index is updated and the data might be recompressed.

## Write Pages to the Redo Log

Specifies whether images of re-compressed pages are written to the redo log. Re-compression may occur when changes are made to compressed data.

### STATISTIC OF COMPRESSION OPERATIONS

#### Compress Attempts

Number of compression operations attempted. Pages are compressed whenever an empty page is created or the space for the uncompressed modification log runs out.

#### Uncompressed Attempts

Number of uncompression operations performed. Compressed InnoDB pages are uncompressed whenever compression fails, or the first time a compressed page is accessed in the buffer pool and the uncompressed page does not exist.

### CPU CORE USAGE

#### CPU Core Usage for Compression

Shows the time in seconds spent by InnoDB Compression operations.

#### CPU Core Usage for Uncompression

Shows the time in seconds spent by InnoDB Uncompression operations.

### BUFFER POOL TOTAL

#### Total Used Pages

Shows the total amount of used compressed pages into the InnoDB Buffer Pool split by page size.

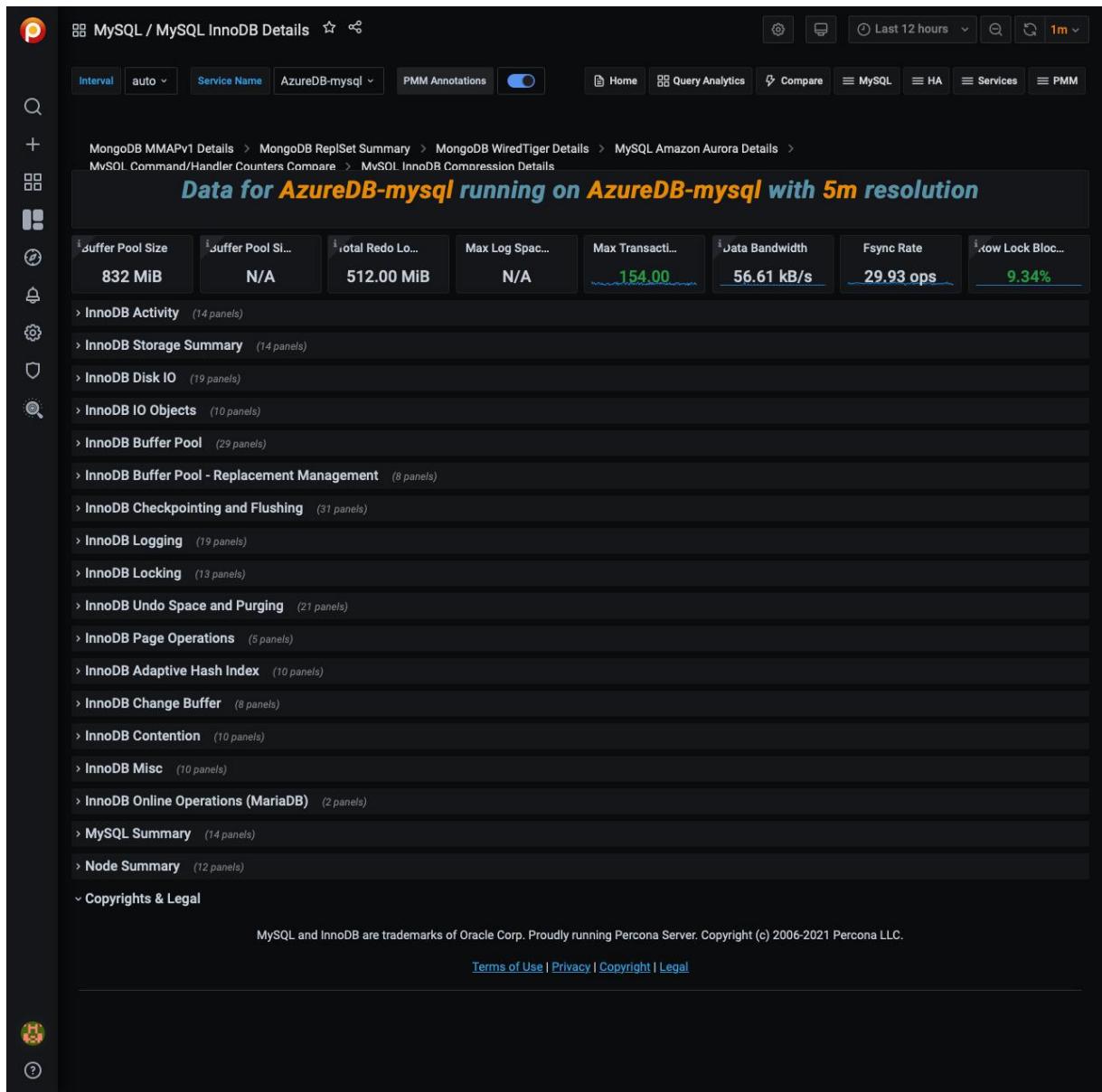
#### Total Free Pages

Shows the total amount of free compressed pages into the InnoDB Buffer Pool split by page size.

---

Last update: 2021-05-11

## MySQL InnoDB Details



### Tip

If metrics are missing, try running: `SET GLOBAL innodb_monitor_enable=all;` in the MySQL client.

## INNODB ACTIVITY

Writes (Rows)

Writes (Transactions)

Row Writes per Trx

Rows Written Per Transactions which modify rows. This is better indicator of transaction write size than looking at all transactions which did not do any writes as well.

**Rows Read Per Trx****Log Space per Trx****Rollbacks**

Percent of Transaction Rollbacks (as portion of read-write transactions).

**BP Reqs Per Row**

Number of Buffer Pool requests per Row Access. High numbers here indicate going through long undo chains, deep trees and other inefficient data access. It can be less than zero due to several rows being read from single page.

**Log Fsync Per Trx**

Log Fsync Per Transaction.

**InnoDB Row Reads****InnoDB Row Operations**

This graph allows you to see which operations occur and the number of rows affected per operation. A graph like Queries Per Second will give you an idea of queries, but one query could effect millions of rows.

**InnoDB Row Writes****InnoDB Row Operations**

This graph allows you to see which operations occur and the number of rows affected per operation. A graph like Queries Per Second will give you an idea of queries, but one query could effect millions of rows.

**InnoDB Read-Only Transactions****InnoDB Read-Write Transactions****InnoDB Transactions Information (RW)**

The InnoDB Transactions Information graph shows details about the recent transactions. Transaction IDs Assigned represents the total number of transactions initiated by InnoDB. RW Transaction Commits are the number of transactions not read-only. Insert-Update Transactions Commits are transactions on the Undo entries. Non Locking RO Transaction Commits are transactions commit from select statement in auto-commit mode or transactions explicitly started with "start transaction read only".

**Misc InnoDB Transactions Information**

Additional InnoDB Transaction Information

**INNODB STORAGE SUMMARY****InnoDB Tables**

Current Number of InnoDB Tables in database

**Data Buffer Pool Fit**

Buffer Pool Size as Portion of the Data

**Avg Row Size**

Amount of Data Per Row

**Index Size Per Row**

Index Size Per Row shows how much space we're using for indexes on per row basics

**InnoDB Data Summary****Space Allocated**

Total Amount of Space Allocated. May not exactly match amount of space used on file system but provided great guidance.

**Space Used**

Space used in All InnoDB Tables. Reported Allocated Space Less Free Space.

**Data Length**

Space Used by Data (Including Primary Key).

**Index Length**

Space Used by Secondary Indexes.

**Estimated Rows**

Estimated number of Rows in InnoDB Storage Engine. It is not exact value and it can change abruptly as information is updated.

**Indexing Overhead**

How Much Indexes Take Compared to Data.

**Free Space Percent**

How Much Space is Free. Too high value wastes space on disk.

**Free**

Allocated Space not currently used by Data or Indexes.

**InnoDB File Per Table**

If Enabled, By Default every Table will have its own Tablespace represented as its own `.idb` file rather than all tables stored in single system tablespace.

**INNODB DISK IO****InnoDB Page Size****Avg Data Read Rq Size****Avg Data Write Rq Size****Avg Log Write Rq Size****Data Written Per Fsync****Log Written Per Fsync****Data Read Per Row Read****Data Written Per Row Written**

Due to difference in timing of Row Write and Data Write the value may be misleading on short intervals.

**InnoDB Data I/O****InnoDB I/O**

- Data Writes - The total number of InnoDB data writes.
- Data Reads - The total number of InnoDB data reads (OS file reads).
- Log Writes - The number of physical writes to the InnoDB redo log file.
- Data Fsyncs - The number of `fsync()` operations. The frequency of `fsync()` calls is influenced by the setting of the `innodb_flush_method` configuration option.

**InnoDB Data Bandwidth****InnoDB Log IO****InnoDB I/O**

- Data Writes - The total number of InnoDB data writes.
- Data Reads - The total number of InnoDB data reads (OS file reads).
- Log Writes - The number of physical writes to the InnoDB redo log file.
- Data Fsyncs - The number of `fsync()` operations. The frequency of `fsync()` calls is influenced by the setting of the `innodb_flush_method` configuration option.

**InnoDB FSyncs****InnoDB Pending IO****InnoDB Pending Fsyncs****InnoDB Auto Extend Increment**

When Growing InnoDB System Tablespace extend it by this size at the time.

**InnoDB Double Write**

Whether InnoDB Double Write Buffer is enabled. Doing so doubles amount of writes InnoDB has to do to storage but is required to avoid potential data corruption during the crash on most storage subsystems.

**InnoDB Fast Shutdown**

Fast Shutdown means InnoDB will not perform complete Undo Space and Change Buffer cleanup on shutdown, which is faster but may interfere with certain major upgrade operations.

**InnoDB Open Files**

Maximum Number of Files InnoDB is Allowed to use.

**InnoDB File Use**

Portion of Allowed InnoDB Open Files Use.

**INNODB IO OBJECTS****InnoDB IO Targets Write Load**

Write Load Includes both Write and fsync (referred as misc).

**INNODB BUFFER POOL****Buffer Pool Size****InnoDB Buffer Pool Size**

InnoDB maintains a storage area called the buffer pool for caching data and indexes in memory. Knowing how the InnoDB buffer pool works, and taking advantage of it to keep frequently accessed data in memory, is one of the most

important aspects of MySQL tuning. The goal is to keep the working set in memory. In most cases, this should be between 60%-90% of available memory on a dedicated database host, but depends on many factors.

#### **Buffer Pool Size of Total RAM**

InnoDB maintains a storage area called the buffer pool for caching data and indexes in memory. Knowing how the InnoDB buffer pool works, and taking advantage of it to keep frequently accessed data in memory, is one of the most important aspects of MySQL tuning. The goal is to keep the working set in memory. In most cases, this should be between 60%-90% of available memory on a dedicated database host, but depends on many factors.

#### **NUMA Interleave**

Interleave Buffer Pool between NUMA zones to better support NUMA systems.

#### **Buffer Pool Activity**

Combined value of Buffer Pool Read and Write requests.

#### **BP Data**

Percent of Buffer Pool Occupied by Cached Data.

#### **BP Data Dirty**

Percent of Data which is Dirty.

#### **BP Miss Ratio**

How often buffer pool read requests have to do read from the disk. Keep this percent low for good performance.

#### **BP Write Buffering**

Number of Logical Writes to Buffer Pool Per logical Write.

#### **InnoDB Buffer Pool LRU Sub-Chain Churn**

#### **Buffer Pool Chunk Size**

Size of the “Chunk” for buffer pool allocation. Allocation of buffer pool will be rounded by this number. It also affects the performance impact of online buffer pool resize.

#### **Buffer Pool Instances**

Number of Buffer Pool Instances. Higher values allow to reduce contention but also increase overhead.

#### **Read Ahead IO Percent**

Percent of Reads Caused by InnoDB Read Ahead.

#### **Read Ahead Wasted**

Percent of Pages Fetched by Read Ahead Evicted Without Access.

#### **Dump Buffer Pool on Shutdown**

#### **Load Buffer Pool at Startup**

#### **Portion of Buffer Pool To Dump/Load**

Larger Portion increases dump/load time but get more of original buffer pool content and hence may reduce warmup time.

#### **Include Buffer Pool in Core Dump**

Whenever to Include Buffer Pool in Crash Core Dumps. Doing so may dramatically increase core dump file slow down restart. Only makes a difference if core dumping on crash is enabled.

**InnoDB Old Blocks**

Percent of The Buffer Pool To be Reserved for “Old Blocks” - which has been touched repeatedly over period of time.

**InnoDB Old Blocks Time**

The Time which has to pass between multiple touches for the block for it to qualify as old block.

**InnoDB Random Read Ahead**

Is InnoDB Random ReadAhead Enabled.

**InnoDB Random Read Ahead**

The Threshold (in Pages) to trigger Linear Read Ahead.

**InnoDB Read IO Threads**

Number of Threads used to Schedule Reads.

**InnoDB Write IO Threads**

Number of Threads used to Schedule Writes.

**InnoDB Native AIO Enabled**

Whether Native Asynchronous IO is enabled. Strongly recommended for optimal performance.

**INNODB BUFFER POOL - REPLACEMENT MANAGEMENT****LRU Scan Depth****InnoDB LRU Scan Depth**

This variable defines InnoDB Free Page Target per buffer pool. When number of free pages falls below this number this number page cleaner will make required amount of pages free, flushing or evicting pages from the tail of LRU as needed.

**LRU Clean Page Searches**

When Page is being read (or created) the Page need to be allocated in Buffer Pool.

**Free List Miss Rate**

The most efficient way to get a clean page is to grab one from free list. However if no pages are available in Free List the LRU scan needs to be performed.

**LRU Get Free Loops**

If Free List was empty LRU Get Free Loop will be performed. It may perform LRU scan or may use some other heuristics and shortcuts to get free page.

**LRU Scans**

If Page could not be find any Free list and other shortcuts did not work, free page will be searched by scanning LRU chain which is not efficient.

**Pages Scanned in LRU Scans**

Pages Scanned Per Second while doing LRU scans. If this value is large (thousands) it means a lot of resources are wasted.

**Pages scanned per LRU Scan**

Number of pages scanned per LRU scan in Average. Large number of scans can consume a lot of resources and also introduce significant addition latency to queries.

**LRU Get Free Waits**

If InnoDB could not find a free page in LRU list and had to sleep. Should be zero.

**INNODB CHECKPOINTING AND FLUSHING****Pages Flushed from Flush List**

Number of Pages Flushed from "Flush List" This combines Pages Flushed through Adaptive Flush and Background Flush.

**Page Flush Batches Executed**

InnoDB Flush Cycle typically Runs on 1 second intervals. If too far off from this number it can indicate an issue.

**Pages Flushed Per Batch**

How many pages are flushed per Batch. Large Batches can "choke" IO subsystem and starve other IO which needs to happen.

**Neighbor Flushing Enabled**

Neighbor Flushing is Optimized for Rotational Media and unless you're Running spinning disks you should disable it.

**InnoDB Checkpoint Age**

The maximum checkpoint age is determined by the total length of all transaction log files ( `innodb_log_file_size` ).

When the checkpoint age reaches the maximum checkpoint age, blocks are flushed synchronously. The rule of the thumb is to keep one hour of traffic in those logs and let the check-pointing perform its work as smooth as possible. If you don't do this, InnoDB will do synchronous flushing at the worst possible time, i.e., when you are busiest.

**Pages Flushed (Adaptive)**

Adaptive Flush Flushes pages from Flush List based on the need to advance Checkpoint (driven by Redo Generation Rate) and by maintaining number of dirty pages within set limit.

**Adaptive Flush Batches Executed****Pages Per Batch (Adaptive)**

Pages Flushed Per Adaptive Batch.

**Neighbor Flushing**

To optimize IO for rotational Media InnoDB may flush neighbor pages. It can cause significant wasted IO for flash storage. Generally for flash you should run with `innodb_flush_neighbors=0` but otherwise this shows how much IO you're wasting.

**Pages Flushed (LRU)**

Flushing from the tail of the LRU list is needed to keep readily-available free pages for new data to be read when data does not fit in the buffer pool.

**LRU Flush Batches Executed****Pages Per Batch (LRU)**

Pages Flushed Per Neighbor.

**LSN Age Flush Batch Target**

Target for Pages to Flush due to LSN Age.

**Pages Flushed (Neighbor)**

Number of Neighbor pages flushed (If neighbor flushing is enabled) from Flush List and LRU List Combined.

**Neighbor Flush Batches Executed****Pages Per Batch (Neighbor)**

Pages Flushed Per Neighbor.

**Sync Flush Waits**

If InnoDB could not keep up with Checkpoint Flushing and had to trigger Sync flush. This should never happen.

**Pages Flushed (Background)**

Pages Flushed by Background Flush which is activated when server is considered to be idle.

**Background Flush Batches Executed****Pages Per Batch (Background)**

Pages Flushed Per Background Batch.

**Redo Generation Rate**

Rate at which LSN (Redo) is Created. It may not match how much data is written to log files due to block size rounding.

**InnoDB Flushing by Type****Pages Evicted (LRU)**

This correspond to number of clean pages which were evicted (made free) from the tail of LRU buffer.

**Page Eviction Batches****Pages Evicted per Batch****Max Log Space Used****Single Page Flushes**

Single Page flushes happen in rare case, then clean page could not be found in LRU list. It should be zero for most workloads.

**Single Page Flush Pages Scanned****Pages Scanned Per Single Page Flush****InnoDB IO Capacity**

Estimated number of IOPS storage system can provide. Is used to scale background activities. Do not set it to actual storage capacity.

**InnoDB IO Capacity Max**

InnoDB IO Capacity to use when falling behind and need to catch up with Flushing.

**INNODB LOGGING****Total Log Space**

Number of InnoDB Log Files Multiplied by Their Size.

**Log Buffer Size****InnoDB Log Buffer Size**

The size of buffer InnoDB uses for buffering writes to log files.

**At Transaction Commit**

What to do with Log file At Transaction Commit. Do nothing and wait for timeout to flush the data from Log Buffer, Flush it to OS Cache but not FSYNC or Flush only.

**Flush Transaction Log Every**

Every Specified Number of Seconds Flush Transaction Log.

**InnoDB Write Ahead Block Size**

This variable can be seen as minimum IO alignment InnoDB will use for Redo log file. High Values cause waste, low values can make IO less efficient.

**Log Write Amplification**

How much Writes to Log Are Amplified compared to how much Redo is Generated.

**Log Fsync Rate****Redo Generated per Trx**

Amount of Redo Generated Per Write Transaction. This is a good indicator of transaction size.

**InnoDB Log File Usage Hourly**

Along with the buffer pool size, `innodb_log_file_size` is the most important setting when we are working with InnoDB. This graph shows how much data was written to InnoDB's redo logs over each hour. When the InnoDB log files are full, InnoDB needs to flush the modified pages from memory to disk.

The rule of thumb is to keep one hour of traffic in those logs and let the checkpointing perform its work as smooth as possible. If you don't do this, InnoDB will do synchronous flushing at the worst possible time, i.e., when you are busiest.

This graph can help guide you in setting the correct `innodb_log_file_size`.

**Log Padding Written**

Amount of Log Padding Written.

**InnoDB Log File Size****InnoDB Log Files**

Number of InnoDB Redo Log Files.

**Log Bandwidth****Redo Generation Rate**

Rate at which LSN (Redo) is Created. It may not match how much data is written to log files due to block size rounding.

**InnoDB Group Commit Batch Size**

The InnoDB Group Commit Batch Size graph shows how many bytes were written to the InnoDB log files per attempt to write. If many threads are committing at the same time, one of them will write the log entries of all the waiting threads and flush the file. Such process reduces the number of disk operations needed and enlarges the batch size.

**INNODB LOCKING****Lock Wait Timeout****InnoDB Lock Wait Timeout**

How long to wait for row lock before timing out.

**InnoDB Deadlock Detection**

If Disabled InnoDB Will not detect deadlocks but rely on timeouts.

**InnoDB Auto Increment Lock Mode**

Will Define How much locking will come from working with Auto Increment Columns.

**Rollback on Timeout**

Whenever to rollback all transaction on timeout or just last statement.

**Row Lock Blocking**

Percent of Active Sessions which are blocked due to waiting on InnoDB Row Locks.

**Row Writes per Trx**

Rows Written Per Transactions which modify rows. This is better indicator of transaction write size than looking at all transactions which did not do any writes as well.

**Rollbacks**

Percent of Transaction Rollbacks (as portion of read-write transactions).

**InnoDB Row Lock Wait Activity****InnoDB Row Lock Wait Time****InnoDB Row Lock Wait Load**

Average Number of Sessions blocked from proceeding due to waiting on row level lock.

**InnoDB Row Locks Activity****InnoDB Table Lock Activity****Current Locks****INNODB UNDO SPACE AND PURGING****Undo Tablespaces****Max Undo Log Size****InnoDB Undo Log Truncate****Purge Threads****Max Purge Lag**

Maximum number of Unpurged Transactions, if this number exceeded delay will be introduced to incoming DDL statements.

**Max Purge Lag Delay****Current Purge Delay**

The Delay Injected due to Purge Thread(s) unable to keep up with purge progress.

**Rollback Segments****InnoDB Purge Activity**

The InnoDB Purge Performance graph shows metrics about the page purging process. The purge process removed the undo entries from the history list and cleanup the pages of the old versions of modified rows and effectively remove deleted rows.

**Transactions and Undo Records****InnoDB Undo Space Usage**

The InnoDB Undo Space Usage graph shows the amount of space used by the Undo segment. If the amount of space grows too much, look for long running transactions holding read views opened in the InnoDB status.

**Transaction History****InnoDB Purge Throttling****Records Per Undo Log Page**

How Many Undo Operations Are Handled Per Each Undo Log Page.

**Purge Invoked**

How Frequently Purge Operation is Invoked.

**Ops Per Purge**

Home Many Purge Actions are done Per invocation.

**Undo Slots Used**

Number of Undo Slots Used.

**Max Transaction History Length****Purge Batch Size****Rseg Truncate Frequency****INNODB PAGE OPERATIONS****InnoDB Page Splits and Merges**

The InnoDB Page Splits graph shows the InnoDB page maintenance activity related to splitting and merging pages. When an InnoDB page, other than the top most leaf page, has too much data to accept a row update or a row insert, it has to be split in two. Similarly, if an InnoDB page, after a row update or delete operation, ends up being less than half full, an attempt is made to merge the page with a neighbor page. If the resulting page size is larger than the InnoDB page size, the operation fails. If your workload causes a large number of page splits, try lowering the `innodb_fill_factor` variable (5.7+).

**Page Merge Success Ratio****InnoDB Page Reorg Attempts**

The InnoDB Page Reorgs graph shows information about the page reorganization operations. When a page receives an update or an insert that affect the offset of other rows in the page, a reorganization is needed. If the reorganization process finds out there is not enough room in the page, the page will be split. Page reorganization can only fail for compressed pages.

**InnoDB Page Reorgs Failures**

The InnoDB Page Reorgs graph shows information about the page reorganization operations. When a page receives an update or an insert that affect the offset of other rows in the page, a reorganization is needed. If the reorganization process finds out there is not enough room in the page, the page will be split. Page reorganization can only fail for compressed pages.

**InnoDB Fill Factor**

The portion of the page to fill then doing sorted Index Build. Lowering this value will worsen space utilization but will reduce need to split pages when new data is inserted in the index.

**INNODB ADAPTIVE HASH INDEX****Adaptive Hash Index Enabled**

Adaptive Hash Index helps to optimize index Look-ups but can be severe hotspot for some workloads.

**Adaptive Hash Index Partitions**

How many Partitions Used for Adaptive Hash Index (to reduce contention).

**Percent of Pages Hashed**

Number of Pages Added to AHI vs Number of Pages Added to Buffer Pool.

**AHI Miss Ratio**

Percent of Searches which could not be resolved through AHI.

**Rows Added Per Page**

Number of Rows "Hashed" Per Each Page which needs to be added to AHI.

**AHI ROI**

How Many Successful Searches using AHI are performed per each row maintenance operation.

**InnoDB AHI Usage**

The InnoDB AHI Usage graph shows the search operations on the InnoDB adaptive hash index and its efficiency. The adaptive hash index is a search hash designed to speed access to InnoDB pages in memory. If the Hit Ratio is small, the working data set is larger than the buffer pool, the AHI should likely be disabled.

**InnoDB AHI Miss Ratio****InnoDB AHI Churn - Rows****InnoDB AHI Churn - Pages****INNODB CHANGE BUFFER****Change Buffer Max Size**

The Maximum Size of Change Buffer (as Percent of Buffer Pool Size).

**Change Buffer Max Size**

The Maximum Size of Change Buffer (Bytes).

**InnoDB Change Buffer Merge Load**

Number of Average of Active Merge Buffer Operations in Process.

**INNODB CONTENTION****InnoDB Thread Concurrency**

If Enabled limits number of Threads allowed inside InnoDB Kernel at the same time.

**InnoDB Commit Concurrency**

If Enabled limits number of Threads allowed inside InnoDB Kernel at the same time during Commit Stage.

**InnoDB Thread Sleep Delay**

The Time the thread will Sleep before Re-Entering InnoDB Kernel if high contention.

**InnoDB Adaptive Max Sleep Delay**

If Set to Non-Zero Value InnoDB Thread Sleep Delay will be adjusted automatically depending on the load up to the value specified by this variable.

**InnoDB Concurrency Tickets**

Number of low level operations InnoDB can do after it entered InnoDB kernel before it is forced to exit and yield to another thread waiting.

**InnoDB Spin Wait Delay****InnoDB Spin Wait Pause Multiplier****InnoDB Sync Spin Loops****InnoDB Contention - OS Waits**

The InnoDB Contention - OS Waits graph shows the number of time an OS wait operation was required while waiting to get the lock. This happens once the spin rounds are exhausted.

**InnoDB Contention - Spin Rounds**

The InnoDB Contention - Spin Rounds graph shows the number of spin rounds executed to get a lock. A spin round is a fast retry to get the lock in a loop.

**INNODB MISC****InnoDB Main Thread Utilization**

The InnoDB Main Thread Utilization graph shows the portion of time the InnoDB main thread spent at various task.

**InnoDB Activity**

The InnoDB Activity graph shows a measure of the activity of the InnoDB threads.

**InnoDB Dedicated Server**

InnoDB automatically optimized for Dedicated Server Environment (auto scaling cache and some other variables).

**InnoDB Sort Buffer Size**

This Buffer is used for Building InnoDB Indexes using Sort algorithm.

**InnoDB Stats Auto Recalc****Update Stats when Metadata Queried**

Refresh InnoDB Statistics when meta-data queries by `SHOW TABLE STATUS` or `INFORMATION_SCHEMA` queries. If Enabled can cause severe performance issues.

**Index Condition Pushdown (ICP)**

Index Condition Pushdown (ICP) is an optimization for the case where MySQL retrieves rows from a table using an index. Without ICP, the storage engine traverses the index to locate rows in the base table and returns them to the MySQL server which evaluates the WHERE condition for the rows. With ICP enabled, and if parts of the WHERE condition can be evaluated by using only columns from the index, the MySQL server pushes this part of the WHERE condition down to the storage engine. The storage engine then evaluates the pushed index condition by using the index entry and only if this is satisfied is the row read from the table. ICP can reduce the number of times the storage engine must access the base table and the number of times the MySQL server must access the storage engine.

**InnoDB Persistent Statistics****InnoDB Persistent Sample Pages**

Number of Pages To Sample if Persistent Statistics are Enabled.

**InnoDB Transient Sample Pages**

Number of Pages To Sample if Persistent Statistics are Disabled.

**INNODB ONLINE OPERATIONS (MARIADB)****InnoDB Defragmentation**

The InnoDB Defragmentation graph shows the status information related to the InnoDB online defragmentation feature of MariaDB for the optimize table command. To enable this feature, the variable `innodb-defragment` must be set to 1 in the configuration file.

Currently available only on a MariaDB server.

**InnoDB Online DDL**

The InnoDB Online DDL graph shows the state of the online DDL (alter table) operations in InnoDB. The progress metric is estimate of the percentage of the rows processed by the online DDL.

Currently available only on a MariaDB server.

**MYSQL SUMMARY****MySQL Uptime****MySQL Uptime**

The amount of time since the last restart of the MySQL server process.

**Current QPS****Current QPS**

Based on the queries reported by MySQL's `SHOW STATUS` command, it is the number of statements executed by the server within the last second. This variable includes statements executed within stored programs, unlike the `Questions` variable. It does not count `COM_PING` or `COM_STATISTICS` commands.

**File Handlers Used****Table Open Cache Miss Ratio****Table Open Cache Size****Table Definition Cache Size****MySQL Connections****Max Connections**

`Max Connections` is the maximum permitted number of simultaneous client connections. By default, this is 151. Increasing this value increases the number of file descriptors that mysqld requires. If the required number of descriptors are not available, the server reduces the value of `Max Connections`.

mysqld actually permits `Max Connections` + 1 clients to connect. The extra connection is reserved for use by accounts that have the SUPER privilege, such as root.

`Max Used Connections` is the maximum number of connections that have been in use simultaneously since the server started.

`Connections` is the number of connection attempts (successful or not) to the MySQL server.

**MySQL Client Thread Activity****MySQL Active Threads**

`Threads Connected` is the number of open connections, while `Threads Running` is the number of threads not sleeping.

## MySQL Handlers

### MySQL Handlers

Handler statistics are internal statistics on how MySQL is selecting, updating, inserting, and modifying rows, tables, and indexes.

This is in fact the layer between the Storage Engine and MySQL.

- `read_rnd_next` is incremented when the server performs a full table scan and this is a counter you don't really want to see with a high value.
- `read_key` is incremented when a read is done with an index.
- `read_next` is incremented when the storage engine is asked to 'read the next index entry'. A high value means a lot of index scans are being done.

## Top Command Counters

### Top Command Counters

The `Com_` statement counter variables indicate the number of times each `xxx` statement has been executed. There is one status variable for each type of statement. For example, `Com_delete` and `Com_update` count `DELETE` and `UPDATE` statements, respectively. `Com_delete_multi` and `Com_update_multi` are similar but apply to `DELETE` and `UPDATE` statements that use multiple-table syntax.

## MySQL Network Traffic

### MySQL Network Traffic

Here we can see how much network traffic is generated by MySQL. Outbound is network traffic sent from MySQL and Inbound is network traffic MySQL has received.

## NODE SUMMARY

### System Uptime

The parameter shows how long a system has been up and running without a shut down or restart.

### Load Average

The system load is a measurement of the computational work the system is performing. Each running process either using or waiting for CPU resources adds 1 to the load.

## RAM

RAM (Random Access Memory) is the hardware in a computing device where the operating system, application programs and data in current use are kept so they can be quickly reached by the device's processor.

### Memory Available

#### Percent of Memory Available

On Modern Linux Kernels amount of Memory Available for application is not the same as Free+Cached+Buffers.

## Virtual Memory

### RAM + SWAP

## Disk Space

Sum of disk space on all partitions.

It can be significantly over-reported in some installations.

#### Min Space Available

Lowest percent of the disk space available.

#### CPU Usage

The CPU time is measured in clock ticks or seconds. It is useful to measure CPU time as a percentage of the CPU's capacity, which is called the CPU usage.

#### CPU Saturation and Max Core Usage

When a system is running with maximum CPU utilization, the transmitting and receiving threads must all share the available CPU. This will cause data to be queued more frequently to cope with the lack of CPU. CPU Saturation may be measured as the length of a wait queue, or the time spent waiting on the queue.

#### Disk I/O and Swap Activity

Disk I/O includes read or write or input/output operations involving a physical disk. It is the speed with which the data transfer takes place between the hard disk drive and RAM.

Swap Activity is memory management that involves swapping sections of memory to and from physical storage.

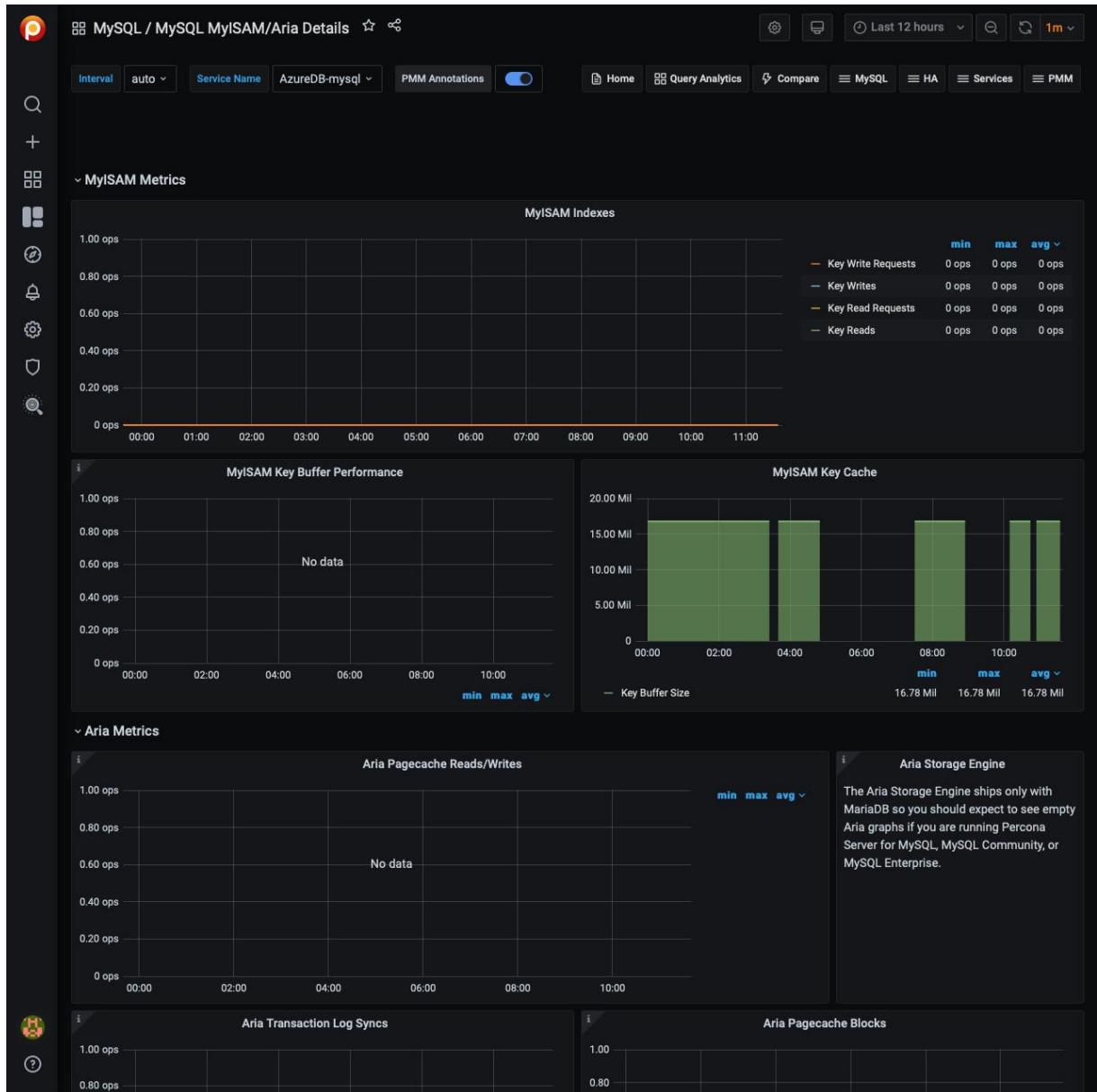
#### Network Traffic

Network traffic refers to the amount of data moving across a network at a given point in time.

---

Last update: 2021-06-25

## MySQL MyISAM/Aria Details



### MYISAM KEY BUFFER PERFORMANCE

The **Key Read Ratio** (`Key_reads / Key_read_requests`) ratio should normally be less than 0.01.

The **Key Write Ratio** (`Key_writes / Key_write_requests`) ratio is usually near 1 if you are using mostly updates and deletes, but might be much smaller if you tend to do updates that affect many rows at the same time or if you are using the `DELAY_KEY_WRITE` table option.

### ARIA PAGECACHE READS/WRITES

This graph is similar to InnoDB buffer pool reads/writes. `aria-pagecache-buffer-size` is the main cache for the Aria storage engine. If you see high reads/writes (physical IO), i.e. reads are close to read requests and/or writes are close to write requests you may need to increase the `aria-pagecache-buffer-size` (may need to decrease other buffers: `key_buffer_size`, `innodb_buffer_pool_size`, etc.)

#### ARIA TRANSACTION LOG SYNCs

This is similar to InnoDB log file syncs. If you see lots of log syncs and want to relax the durability settings you can change `aria_checkpoint_interval` (in seconds) from 30 (default) to a higher number. It is good to look at the disk IO dashboard as well.

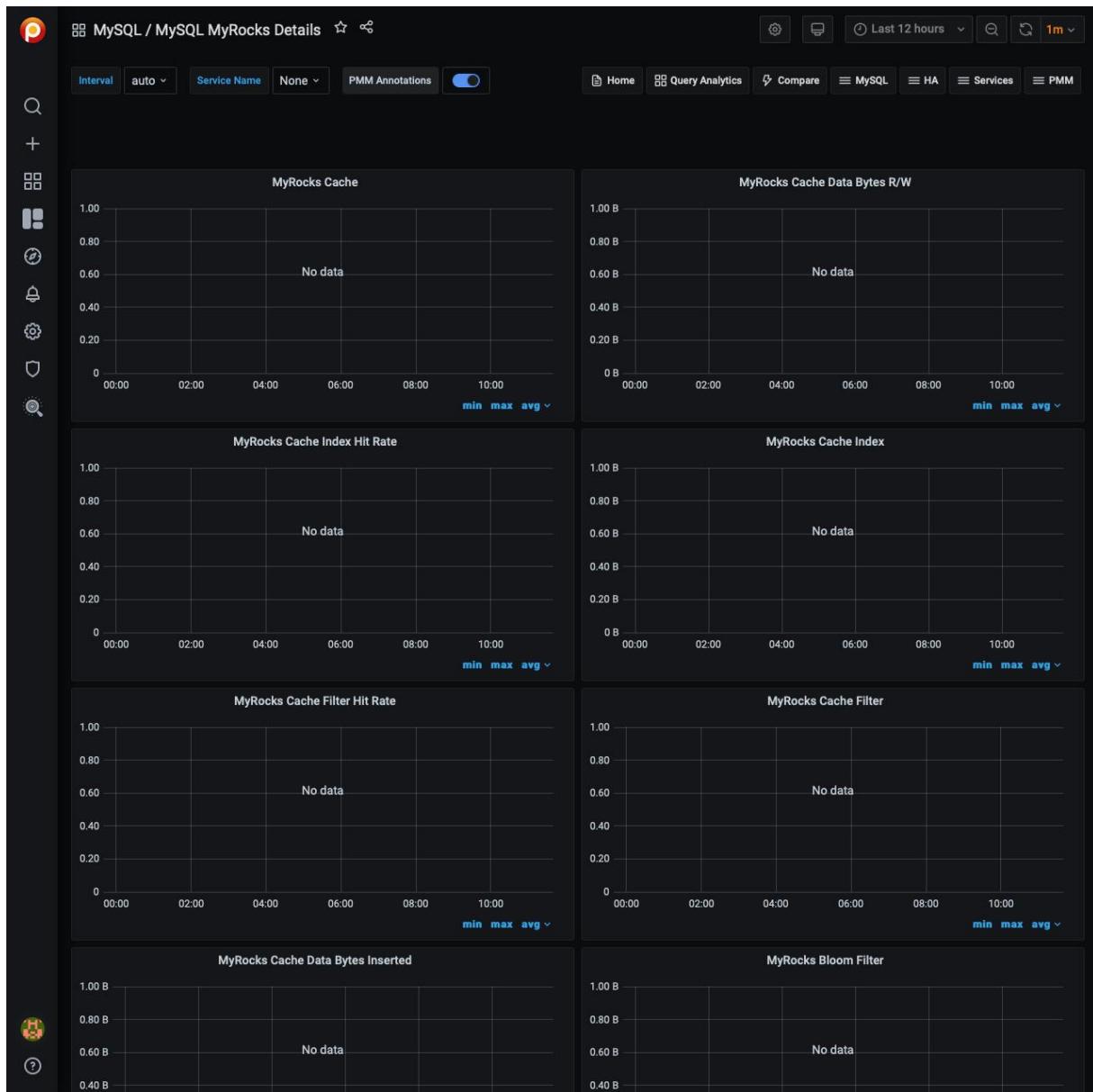
#### ARIA PAGECACHE BLOCKS

This graph shows the utilization for the Aria pagecache. This is similar to InnoDB buffer pool graph. If you see all blocks are used you may consider increasing `aria-pagecache-buffer-size` (may need to decrease other buffers: `key_buffer_size`, `innodb_buffer_pool_size`, etc.)

---

Last update: 2021-05-11

## MySQL MyRocks Details



The [MyRocks](#) storage engine developed by Facebook based on the RocksDB storage engine is applicable to systems which primarily interact with the database by writing data to it rather than reading from it. RocksDB also features a good level of compression, higher than that of the InnoDB storage engine, which makes it especially valuable when optimizing the usage of hard drives.

PMM collects statistics on the MyRocks storage engine for MySQL in the Metrics Monitor information for this dashboard comes from the *Information Schema* tables.

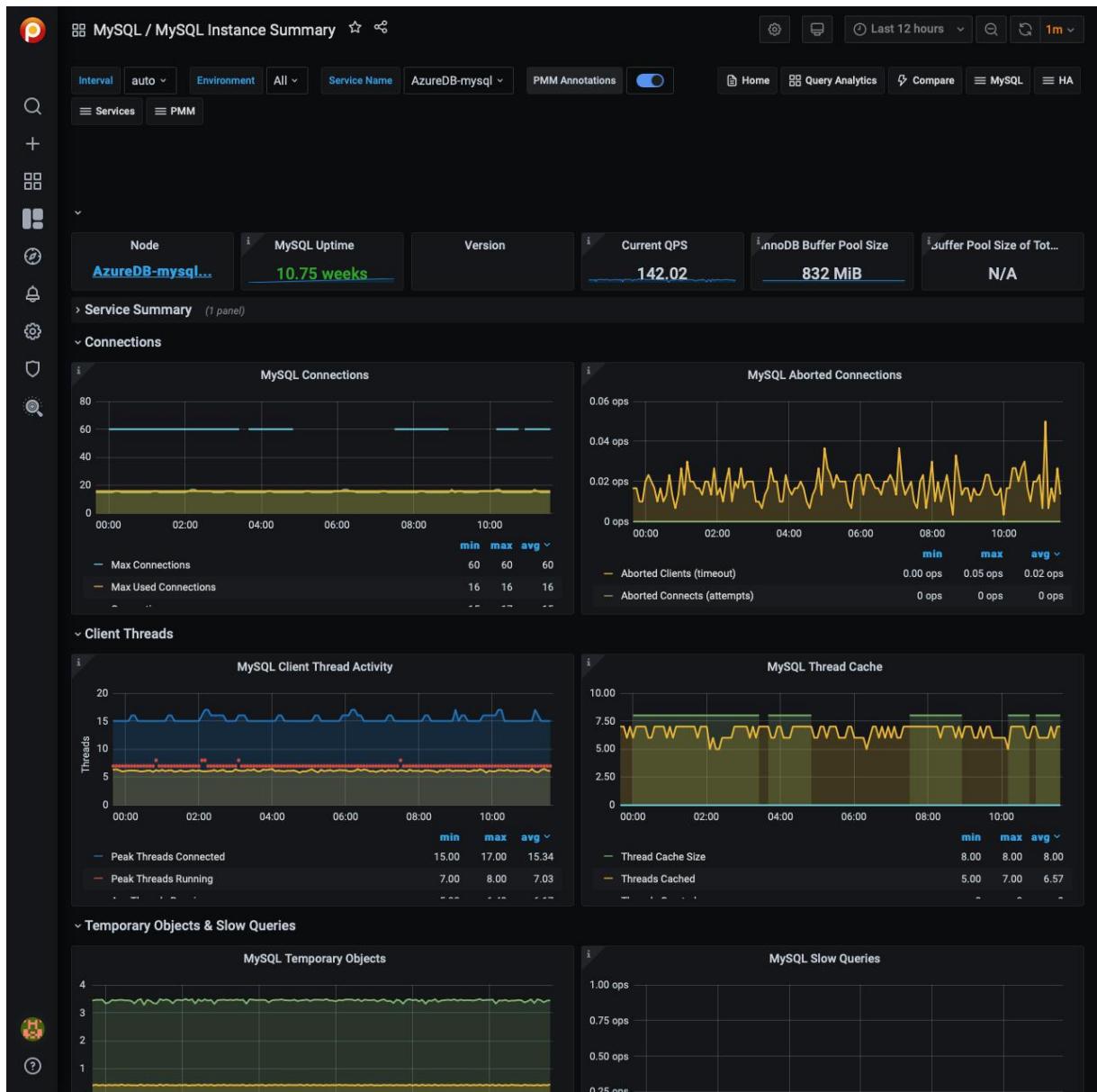
### METRICS

- MyRocks cache
- MyRocks cache data bytes R/W
- MyRocks cache index hit rate
- MyRocks cache index
- MyRocks cache filter hit rate

- MyRocks cache filter
  - MyRocks cache data bytes inserted
  - MyRocks bloom filter
  - MyRocks memtable
  - MyRocks memtable size
  - MyRocks number of keys
  - MyRocks cache L0/L1
  - MyRocks number of DB ops
  - MyRocks R/W
  - MyRocks bytes read by iterations
  - MyRocks write ops
  - MyRocks WAL
  - MyRocks number reseeks in iterations
  - RocksDB row operations
  - MyRocks file operations
  - RocksDB stalls
  - RocksDB stops/slowdowns
- 

Last update: 2021-06-25

## MySQL Instance Summary



### MYSQL CONNECTIONS

#### Max Connections

Max Connections is the maximum permitted number of simultaneous client connections. By default, this is 151. Increasing this value increases the number of file descriptors that `mysqld` requires. If the required number of descriptors are not available, the server reduces the value of Max Connections.

`mysqld` actually permits `Max Connections + 1` clients to connect. The extra connection is reserved for use by accounts that have the SUPER privilege, such as root.

Max Used Connections is the maximum number of connections that have been in use simultaneously since the server started.

Connections is the number of connection attempts (successful or not) to the MySQL server.

**MYSQL ABORTED CONNECTIONS****Aborted Connections**

When a given host connects to MySQL and the connection is interrupted in the middle (for example due to bad credentials), MySQL keeps that info in a system table (since 5.6 this table is exposed in `performance_schema`).

If the amount of failed requests without a successful connection reaches the value of `max_connect_errors`, `mysqld` assumes that something is wrong and blocks the host from further connection.

To allow connections from that host again, you need to issue the `FLUSH HOSTS` statement.

**MYSQL CLIENT THREAD ACTIVITY****MySQL Active Threads**

Threads Connected is the number of open connections, while Threads Running is the number of threads not sleeping.

**MYSQL THREAD CACHE****MySQL Thread Cache**

The `thread_cache_size` variable sets how many threads the server should cache to reuse. When a client disconnects, the client's threads are put in the cache if the cache is not full. It is auto-sized in MySQL 5.6.8 and above (capped to 100). Requests for threads are satisfied by reusing threads taken from the cache if possible, and only when the cache is empty is a new thread created.

- `threads_created` : The number of threads created to handle connections.
- `threads_cached` : The number of threads in the thread cache.

**MYSQL SLOW QUERIES****MySQL Slow Queries**

Slow queries are defined as queries being slower than the `long_query_time` setting. For example, if you have `long_query_time` set to 3, all queries that take longer than 3 seconds to complete will show on this graph.

**MYSQL SELECT TYPES****MySQL Select Types**

As with most relational databases, selecting based on indexes is more efficient than scanning an entire table's data. Here we see the counters for selects not done with indexes.

- **Select Scan** is how many queries caused full table scans, in which all the data in the table had to be read and either discarded or returned.
- **Select Range** is how many queries used a range scan, which means MySQL scanned all rows in a given range.
- **Select Full Join** is the number of joins that are not joined on an index, this is usually a huge performance hit.

**MYSQL SORTS****MySQL Sorts**

Due to a query's structure, order, or other requirements, MySQL sorts the rows before returning them. For example, if a table is ordered 1 to 10 but you want the results reversed, MySQL then has to sort the rows to return 10 to 1.

This graph also shows when sorts had to scan a whole table or a given range of a table to return the results and which could not have been sorted via an index.

**MYSQL TABLE LOCKS****Table Locks**

MySQL takes a number of different locks for varying reasons. In this graph we see how many Table level locks MySQL has requested from the storage engine. In the case of InnoDB, many times the locks could actually be row locks as it only takes table level locks in a few specific cases.

It is most useful to compare Locks Immediate and Locks Waited. If Locks waited is rising, it means you have lock contention. Otherwise, Locks Immediate rising and falling is normal activity.

**MYSQL QUESTIONS****MySQL Questions**

The number of statements executed by the server. This includes only statements sent to the server by clients and not statements executed within stored programs, unlike the Queries used in the QPS calculation.

This variable does not count the following commands:

- COM\_PING
- COM\_STATISTICS
- COM\_STMT\_PREPARE
- COM\_STMT\_CLOSE
- COM\_STMT\_RESET

**MYSQL NETWORK TRAFFIC****MySQL Network Traffic**

Here we can see how much network traffic is generated by MySQL. Outbound is network traffic sent from MySQL and Inbound is network traffic MySQL has received.

**MYSQL NETWORK USAGE HOURLY****MySQL Network Usage Hourly**

Here we can see how much network traffic is generated by MySQL per hour. You can use the bar graph to compare data sent by MySQL and data received by MySQL.

**MYSQL INTERNAL MEMORY OVERVIEW**

**System Memory:** Total Memory for the system.

**InnoDB Buffer Pool Data:** InnoDB maintains a storage area called the buffer pool for caching data and indexes in memory.

**TokuDB Cache Size:** Similar in function to the InnoDB Buffer Pool, TokuDB will allocate 50% of the installed RAM for its own cache.

**Key Buffer Size:** Index blocks for MyISAM tables are buffered and are shared by all threads. `key_buffer_size` is the size of the buffer used for index blocks.

**Adaptive Hash Index Size:** When InnoDB notices that some index values are being accessed very frequently, it builds a hash index for them in memory on top of B-Tree indexes.

**Query Cache Size:** The query cache stores the text of a SELECT statement together with the corresponding result that was sent to the client. The query cache has huge scalability problems in that only one thread can do an operation in the query cache at the same time.

**InnoDB Dictionary Size:** The data dictionary is InnoDB's internal catalog of tables. InnoDB stores the data dictionary on disk, and loads entries into memory while the server is running.

**InnoDB Log Buffer Size:** The MySQL InnoDB log buffer allows transactions to run without having to write the log to disk before the transactions commit.

#### TOP COMMAND COUNTERS

##### Top Command Counters

The `Com_xxx` statement counter variables indicate the number of times each `xxx` statement has been executed. There is one status variable for each type of statement. For example, `Com_delete` and `Com_update` count `DELETE` and `UPDATE` statements, respectively. `Com_delete_multi` and `Com_update_multi` are similar but apply to `DELETE` and `UPDATE` statements that use multiple-table syntax.

#### TOP COMMAND COUNTERS HOURLY

##### Top Command Counters Hourly

The `Com_xxx` statement counter variables indicate the number of times each `xxx` statement has been executed. There is one status variable for each type of statement. For example, `Com_delete` and `Com_update` count `DELETE` and `UPDATE` statements, respectively. `Com_delete_multi` and `Com_update_multi` are similar but apply to `DELETE` and `UPDATE` statements that use multiple-table syntax.

#### MYSQL HANDLERS

##### MySQL Handlers

Handler statistics are internal statistics on how MySQL is selecting, updating, inserting, and modifying rows, tables, and indexes.

This is in fact the layer between the Storage Engine and MySQL.

- `read_rnd_next` is incremented when the server performs a full table scan and this is a counter you don't really want to see with a high value.
- `read_key` is incremented when a read is done with an index.
- `read_next` is incremented when the storage engine is asked to 'read the next index entry'. A high value means a lot of index scans are being done.

#### MYSQL QUERY CACHE MEMORY

##### MySQL Query Cache Memory

The query cache has huge scalability problems in that only one thread can do an operation in the query cache at the same time. This serialization is true not only for SELECTs, but also for INSERT/UPDATE/DELETE.

This also means that the larger the `query_cache_size` is set to, the slower those operations become. In concurrent environments, the MySQL Query Cache quickly becomes a contention point, decreasing performance. MariaDB and AWS Aurora have done work to try and eliminate the query cache contention in their flavors of MySQL, while MySQL 8.0 has eliminated the query cache feature.

The recommended settings for most environments is to set:

- `query_cache_type=0`
- `query_cache_size=0`

#### Tip

While you can dynamically change these values, to completely remove the contention point you have to restart the database.

## MYSQl QUERY CACHE ACTIVITY

### MySQL Query Cache Activity

The query cache has huge scalability problems in that only one thread can do an operation in the query cache at the same time. This serialization is true not only for SELECTs, but also for INSERT/UPDATE/DELETE.

This also means that the larger the `query_cache_size` is set to, the slower those operations become. In concurrent environments, the MySQL Query Cache quickly becomes a contention point, decreasing performance. MariaDB and AWS Aurora have done work to try and eliminate the query cache contention in their flavors of MySQL, while MySQL 8.0 has eliminated the query cache feature.

The recommended settings for most environments is to set:

- `query_cache_type=0`
- `query_cache_size=0`



#### Tip

While you can dynamically change these values, to completely remove the contention point you have to restart the database.

## MYSQl TABLE OPEN CACHE STATUS

### MySQL Table Open Cache Status

The recommendation is to set the `table_open_cache_instances` to a loose correlation to virtual CPUs, keeping in mind that more instances means the cache is split more times. If you have a cache set to 500 but it has 10 instances, each cache will only have 50 cached.

The `table_definition_cache` and `table_open_cache` can be left as default as they are auto-sized MySQL 5.6 and above (i.e., do not set them to any value).

## MYSQl OPEN TABLES

### MySQL Open Tables

The recommendation is to set the `table_open_cache_instances` to a loose correlation to virtual CPUs, keeping in mind that more instances means the cache is split more times. If you have a cache set to 500 but it has 10 instances, each cache will only have 50 cached.

The `table_definition_cache` and `table_open_cache` can be left as default as they are auto-sized MySQL 5.6 and above (i.e., do not set them to any value).

## MYSQl TABLE DEFINITION CACHE

### MySQL Table Definition Cache

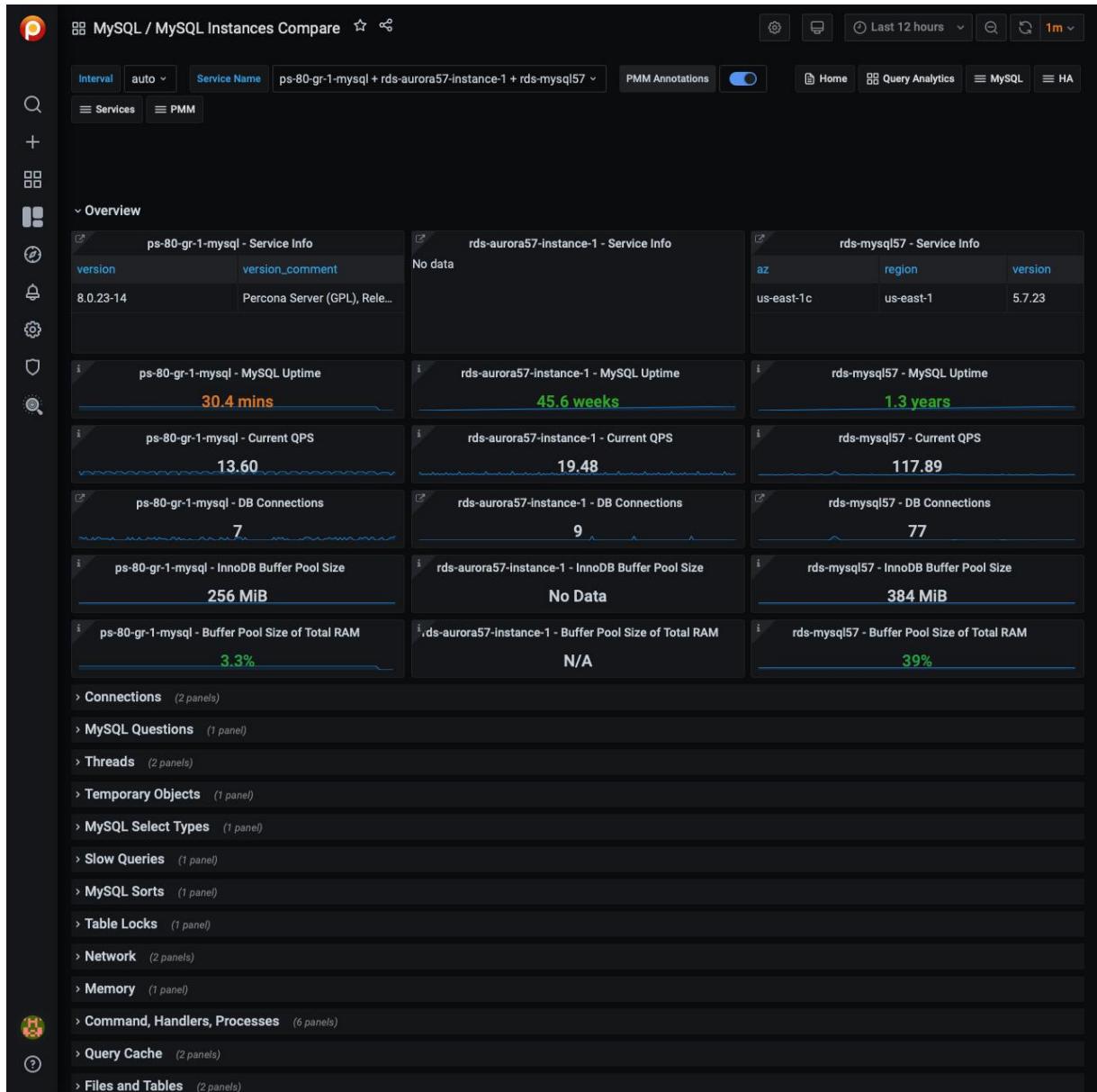
The recommendation is to set the `table_open_cache_instances` to a loose correlation to virtual CPUs, keeping in mind that more instances means the cache is split more times. If you have a cache set to 500 but it has 10 instances, each cache will only have 50 cached.

The `table_definition_cache` and `table_open_cache` can be left as default as they are auto-sized MySQL 5.6 and above (i.e., do not set them to any value).

---

Last update: 2021-06-25

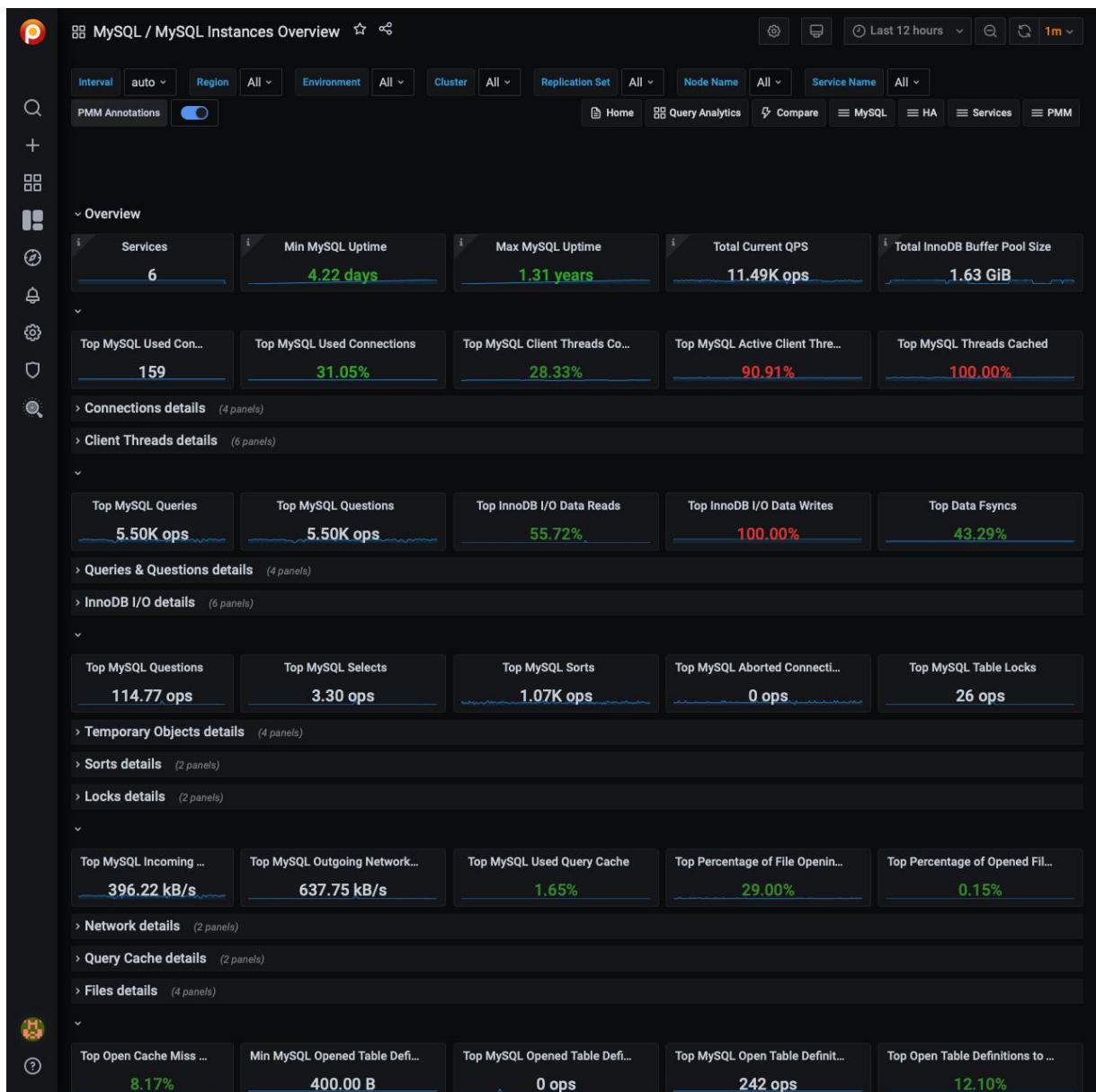
## MySQL Instances Compare



No description

Last update: 2021-05-11

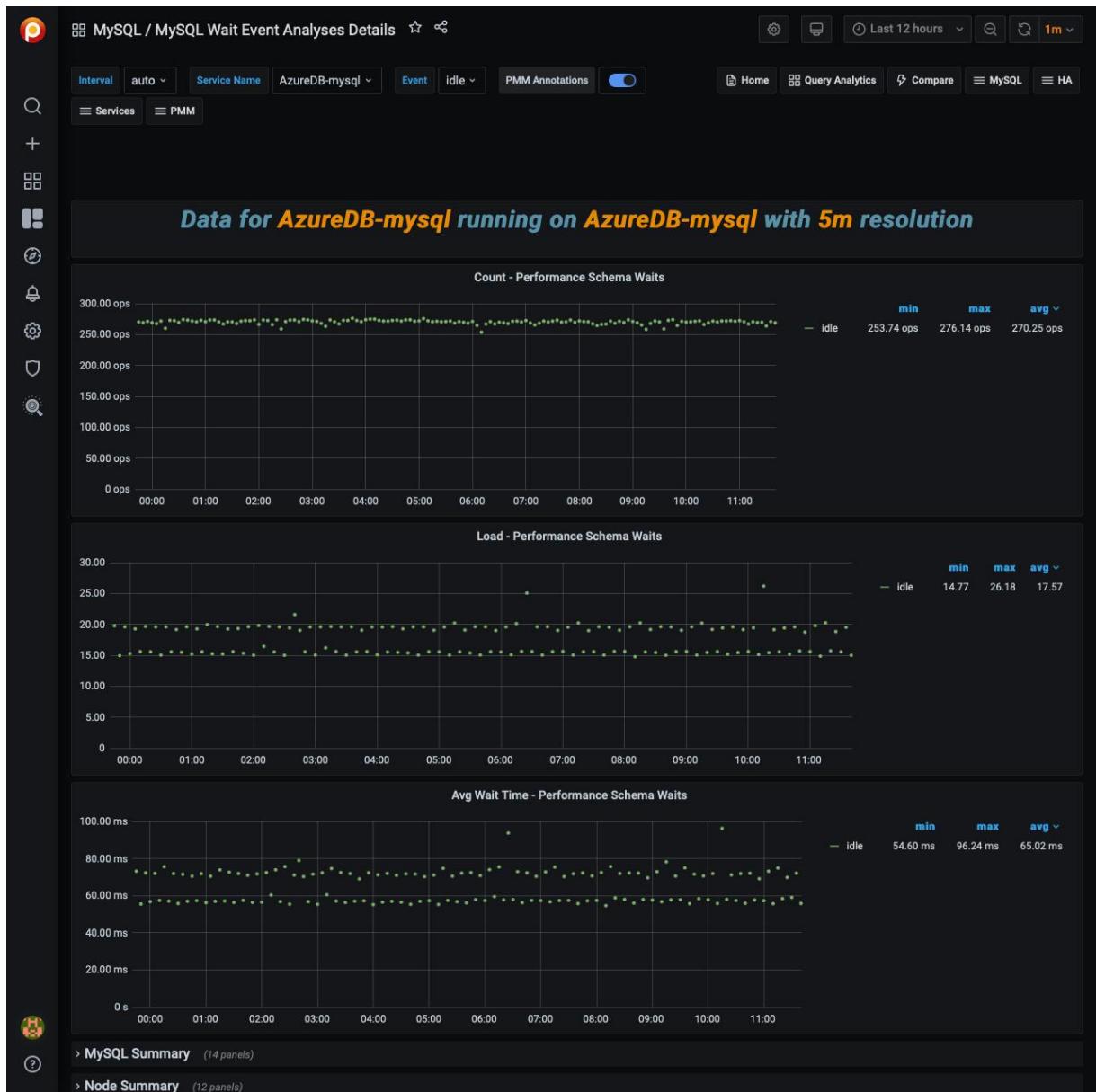
## MySQL Instances Overview



No description

Last update: 2021-06-03

## MySQL Wait Event Analyses Details

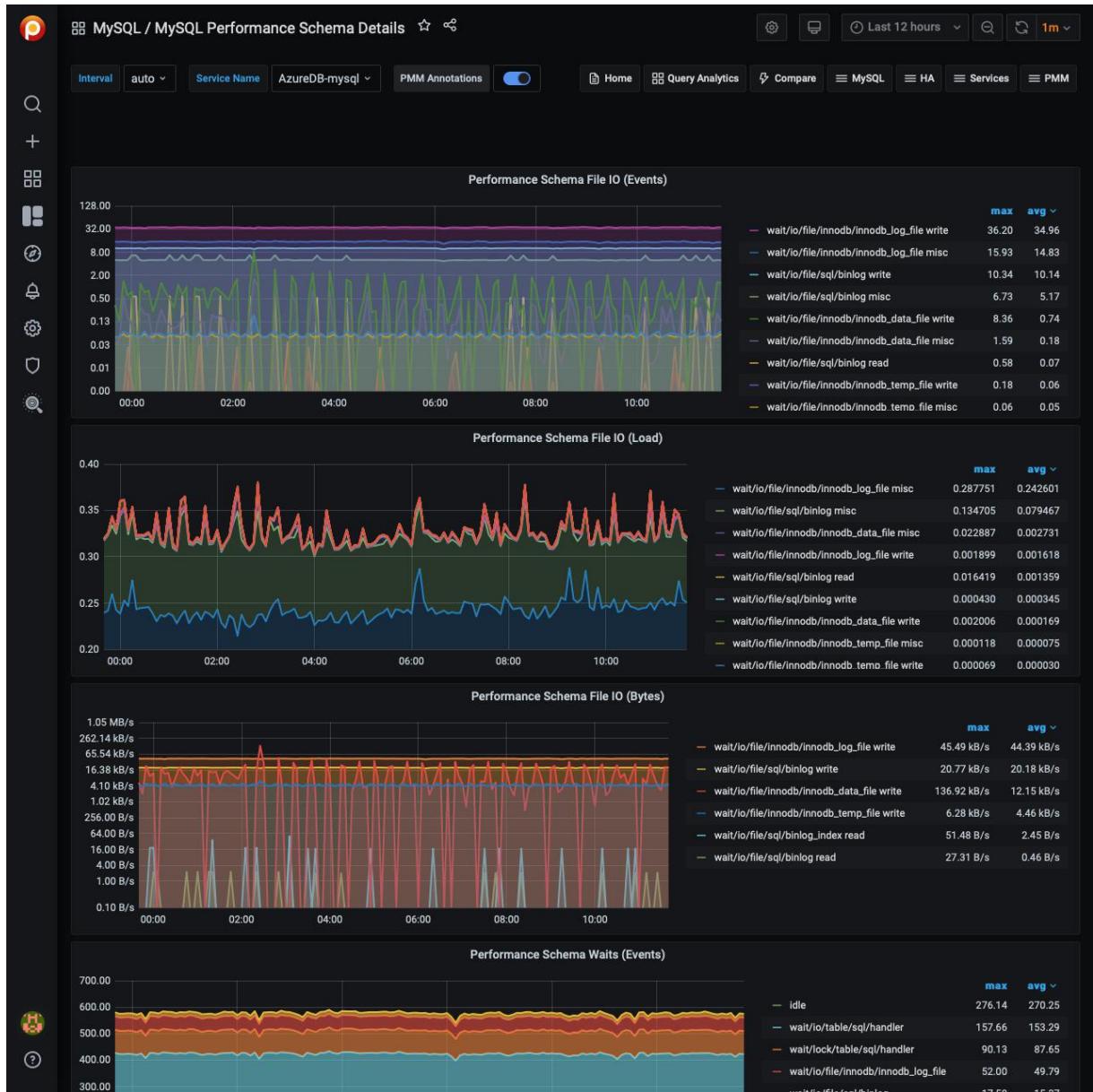


This dashboard helps to analyze *Performance Schema* wait events. It plots the following metrics for the chosen (one or more) wait events:

- Count - Performance Schema Waits
- Load - Performance Schema Waits
- Avg Wait Time - Performance Schema Waits

Last update: 2021-06-25

## MySQL Performance Schema Details

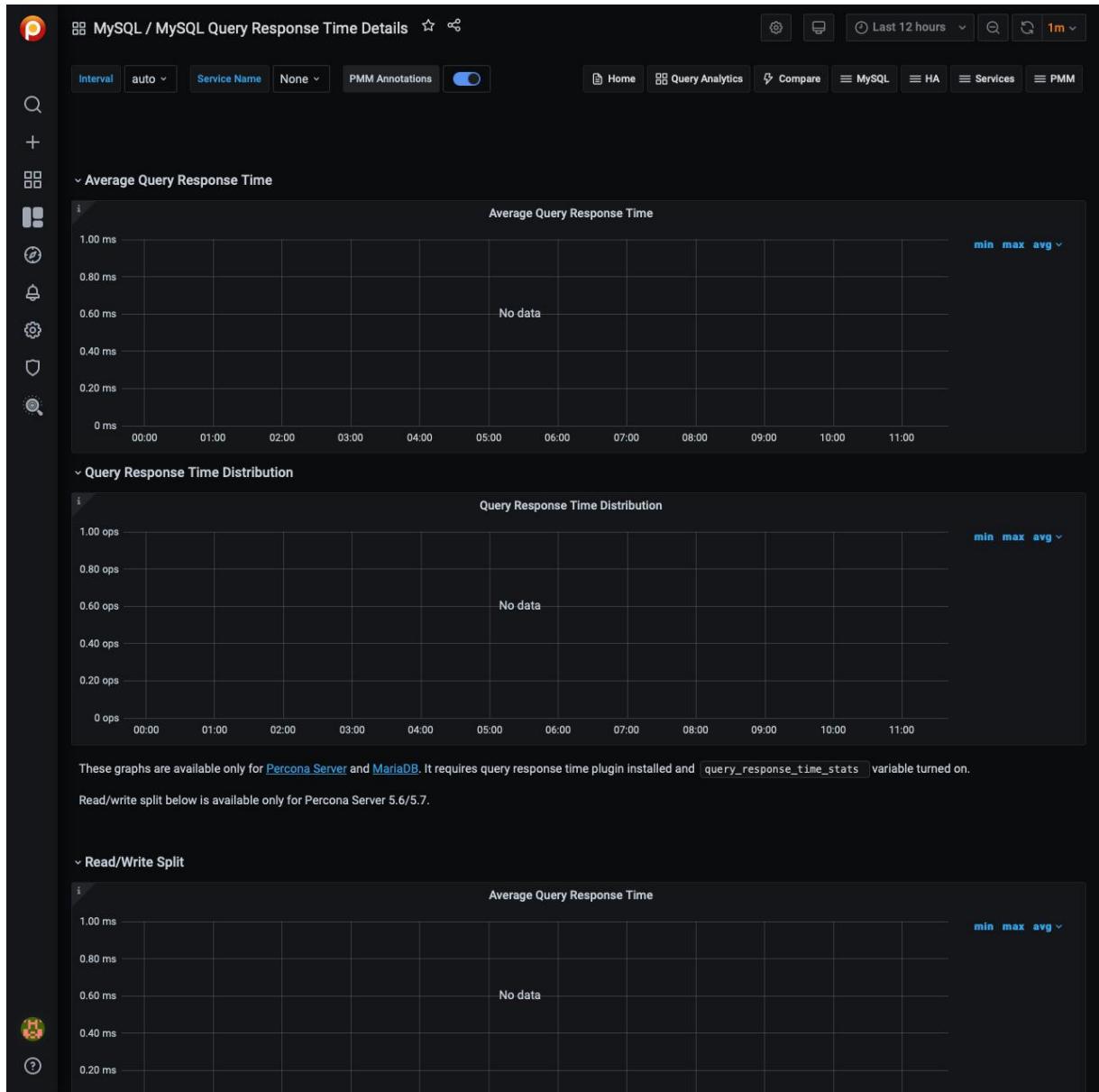


The MySQL Performance Schema dashboard helps determine the efficiency of communicating with Performance Schema. This dashboard contains the following metrics:

- Performance Schema file IO (events)
- Performance Schema file IO (load)
- Performance Schema file IO (Bytes)
- Performance Schema waits (events)
- Performance Schema waits (load)
- Index access operations (load)
- Table access operations (load)
- Performance Schema SQL and external locks (events)
- Performance Schema SQL and external locks (seconds)

Last update: 2021-06-25

## MySQL Query Response Time Details



### AVERAGE QUERY RESPONSE TIME

The Average Query Response Time graph shows information collected using the Response Time Distribution plugin sourced from table `INFORMATION_SCHEMA.QUERY_RESPONSE_TIME`. It computes this value across all queries by taking the sum of seconds divided by the count of queries.

### QUERY RESPONSE TIME DISTRIBUTION

Query response time counts (operations) are grouped into three buckets:

- 100 ms - 1 s
- 1 s - 10 s
- > 10 s

### AVERAGE QUERY RESPONSE TIME

Available only in Percona Server for MySQL, provides visibility of the split of **READ** vs **WRITE** query response time.

**READ QUERY RESPONSE TIME DISTRIBUTION**

Available only in Percona Server for MySQL, illustrates READ query response time counts (operations) grouped into three buckets:

- 100 ms - 1 s
- 1 s - 10 s
- > 10 s

**WRITE QUERY RESPONSE TIME DISTRIBUTION**

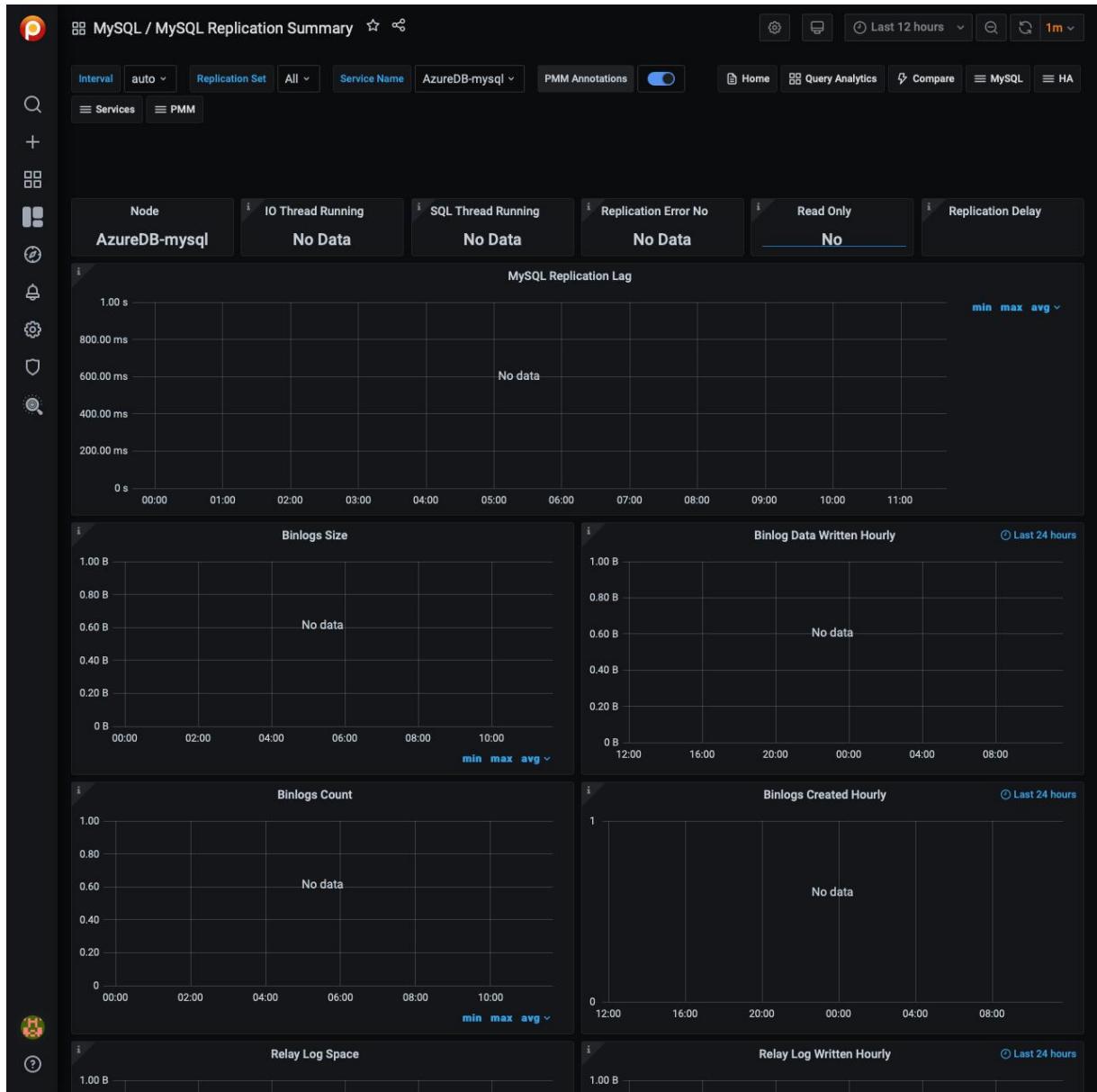
Available only in Percona Server for MySQL, illustrates WRITE query response time counts (operations) grouped into three buckets:

- 100 ms - 1 s
- 1 s - 10 s
- > 10 s

---

Last update: 2021-06-25

## MySQL Replication Summary



### IO THREAD RUNNING

This metric shows if the IO Thread is running or not. It only applies to a secondary host.

SQL Thread is a process that runs on a secondary host in the replication environment. It reads the events from the local relay log file and applies them to the secondary server.

Depending on the format of the binary log it can read query statements in plain text and re-execute them or it can read raw data and apply them to the local host.

#### Possible values

Yes

The thread is running and is connected to a replication primary

No

The thread is not running because it is not launched yet or because an error has occurred connecting to the primary host

Connecting

The thread is running but is not connected to a replication primary

No value

The host is not configured to be a replication secondary

IO Thread Running is one of the parameters that the command `SHOW SLAVE STATUS` returns.

#### **SQL THREAD RUNNING**

This metric shows if the SQL thread is running or not. It only applies to a secondary host.

#### **Possible values**

Yes

SQL Thread is running and is applying events from the relay log to the local secondary host

No

SQL Thread is not running because it is not launched yet or because of an error occurred while applying an event to the local secondary host

#### **REPLICATION ERROR NO**

This metric shows the number of the last error in the SQL Thread encountered which caused replication to stop.

One of the more common errors is *Error: 1022 Duplicate Key Entry*. In such a case replication is attempting to update a row that already exists on the secondary. The SQL Thread will stop replication to avoid data corruption.

#### **READ ONLY**

This metric indicates whether the host is configured to be in *Read Only* mode or not.

#### **Possible values**

Yes

The secondary host permits no client updates except from users who have the SUPER privilege or the REPLICATION SLAVE privilege.

This kind of configuration is typically used for secondary hosts in a replication environment to avoid a user can inadvertently or voluntarily modify data causing inconsistencies and stopping the replication process.

No

The secondary host is not configured in *Read Only* mode.

#### **MYSQL REPLICATION DELAY**

This metric shows the number of seconds the secondary host is delayed in replication applying events compared to when the primary host applied them, denoted by the `Seconds_Behind_Master` value, and only applies to a secondary host.

Since the replication process applies the data modifications on the secondary asynchronously, it could happen that the secondary replicates events after some time. The main reasons are:

- **Network round trip time** - high latency links will lead to non-zero replication lag values.
- **Single threaded nature of replication channels** - primary servers have the advantage of applying changes in parallel, whereas secondary ones are only able to apply changes in serial, thus limiting their throughput. In some cases Group Commit can help but is not always applicable.
- **High number of changed rows or computationally expensive SQL** - depending on the replication format (`ROW` vs `STATEMENT`), significant changes to the database through high volume of rows modified, or expensive CPU will all contribute to secondary servers lagging behind the primary.

Generally adding more CPU or Disk resources can alleviate replication lag issues, up to a point.

#### BINLOG SIZE

This metric shows the overall size of the binary log files, which can exist on both primary and secondary servers.

The binary log (also known as the *binlog*) contains events that describe database changes: `CREATE TABLE`, `ALTER TABLE`, , updates, inserts, deletes and other statements or database changes.

The binlog file is read by secondaries via their IO Thread process to replicate database changes modification on the data and on the table structures. There can be more than one binlog file depending on the binlog rotation policy (for example using the configuration variables `max_binlog_size` and `expire_logs_days`) or because of server reboots.

When planning the disk space, take care of the overall dimension of binlog files and adopt a good rotation policy or think about having a separate mount point or disk to store the binlog data.

#### BINLOG DATA WRITTEN HOURLY

This metric shows the amount of data written hourly to the binlog files during the last 24 hours. This metric can give you an idea of how big is your application in terms of data writes (creation, modification, deletion).

#### BINLOG COUNT

This metric shows the overall count of binary log files, on both primary and secondary servers.

#### BINLOGS CREATED HOURLY

This metric shows the number of binlog files created hourly during the last 24 hours.

#### RELAY LOG SPACE

This metric shows the overall size of the relay log files. It only applies to a secondary host.

The relay log consists of a set of numbered files containing the events to be executed on the secondary host to replicate database changes.

The relay log has the same format as the binlog.

There can be multiple relay log files depending on the rotation policy adopted (using the configuration variable `max_relay_log_size`).

As soon as the SQL thread completes to execute all events in the relay log file, the file is deleted.

If this metric contains a high value, the variable `max_relay_log_file` is high too. Generally, this not a serious issue. If the value of this metric is constantly increased, the secondary is delaying too much in applying the events.

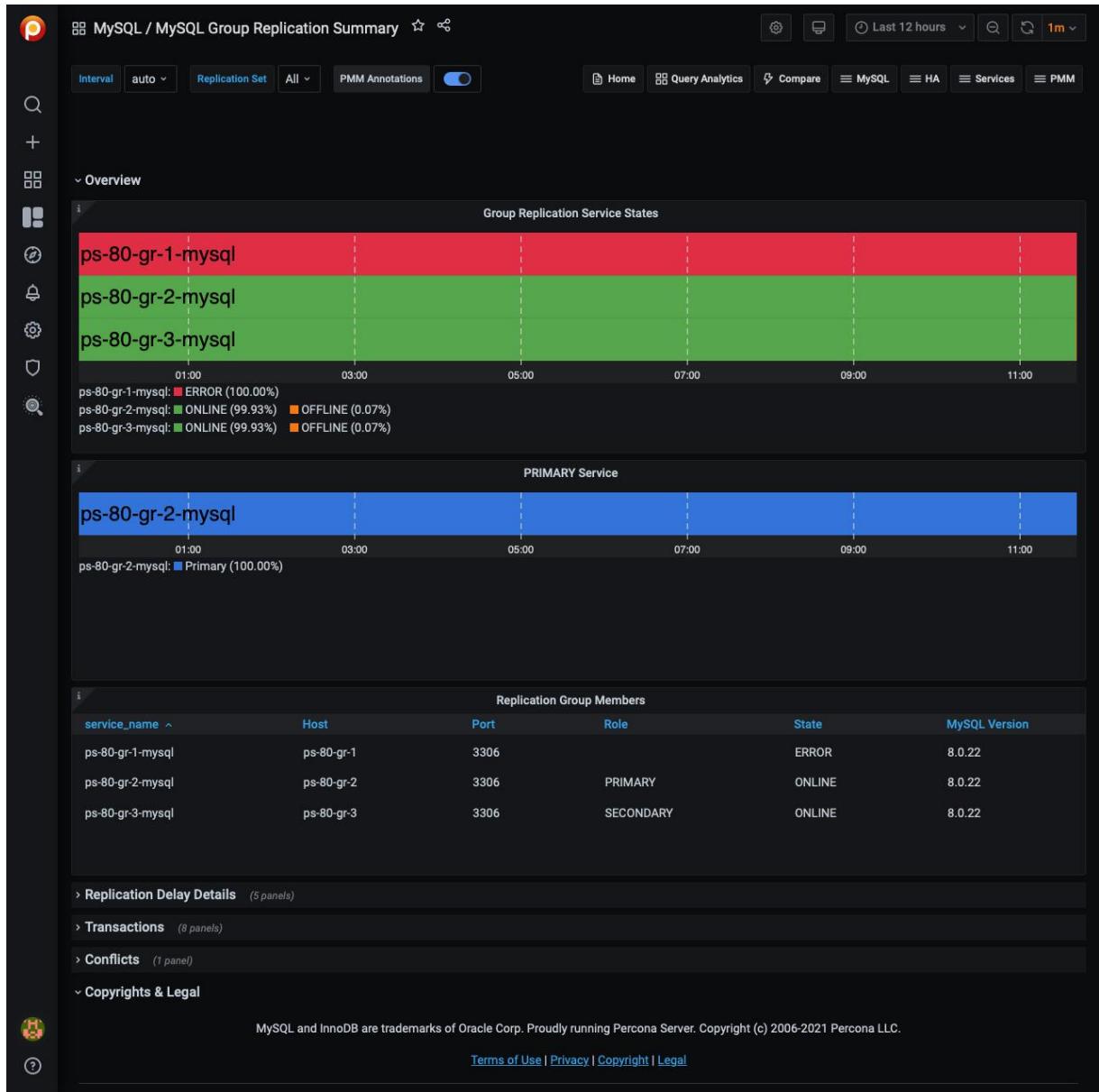
Treat this metric in the same way as the [MySQL Replication Delay](#) metric.

#### RELAY LOG WRITTEN HOURLY

This metric shows the amount of data written hourly into relay log files during the last 24 hours.

Last update: 2021-06-25

## MySQL Group Replication Summary



### OVERVIEW

- PRIMARY Service
- Group Replication Service States
- Replication Group Members
- Replication Lag
- Replication Delay
- Transport Time

### TRANSACTIONS

- Transaction Details
- Applied Transactions
- Sent Transactions

- Checked Transactions
- Rolled Back Transactions
- Transactions Row Validating
- Transactions in the Queue for Checking
- Received Transactions Queue

**CONFLICTS**

- Detected Conflicts
- 

Last update: 2021-06-03

## MySQL Table Details

The dashboard provides detailed information about MySQL databases, specifically focusing on table sizes and database growth.

**Largest Tables by Row Count:**

| schema  | service_name        | table      | Value |
|---------|---------------------|------------|-------|
| sbtest1 | ps-80-replica-mysql | sbtest1    | 34 K  |
| sbtest  | mariadb-105-mysql   | sbtest1    | 10 K  |
| sbtest  | ps-80-gr-3-mysql    | sbtest1    | 10 K  |
| sbtest  | ps-80-gr-2-mysql    | sbtest1    | 10 K  |
| sys     | ps-80-replica-mysql | sys_config | 6     |
| sys     | ps-80-nr-3-mysql    | sys_config | 6     |

**Largest Tables by Size:**

| schema    | service_name        | table        | Value  |
|-----------|---------------------|--------------|--------|
| sbtest1   | ps-80-replica-mysql | sbtest1      | 9 MiB  |
| sbtest    | mariadb-105-mysql   | sbtest1      | 3 MiB  |
| sbtest    | ps-80-gr-2-mysql    | sbtest1      | 3 MiB  |
| sbtest    | ps-80-gr-3-mysql    | sbtest1      | 3 MiB  |
| employees | ps-80-replica-mysql | dept_manager | 32 KiB |
| employees | ps-80-replica-mysql | departments  | 32 KiB |

**Total Databases Size - mariadb-105-mysql:**

**Most Fragmented Tables by Freeable Size - mariadb-105-mysql:**

| schema | service_name      | table   | Value |
|--------|-------------------|---------|-------|
| sbtest | mariadb-105-mysql | sbtest1 | 4 MiB |

**Total Databases Size - ps-80-gr-2-mysql:**

**Most Fragmented Tables by Freeable Size - ps-80-gr-2-mysql:**

| schema | service_name     | table                                 | Value |
|--------|------------------|---------------------------------------|-------|
| sbtest | ps-80-gr-2-mysql | sbtest1                               | 4 MiB |
| sys    | ps-80-gr-2-mysql | x\$ps_digest_avg_latency_distribution | 0 B   |
| sys    | ps-80-gr-2-mysql | waits_global_by_latency               | 0 B   |
| sys    | ps-80-gr-2-mysql | user_summary_by_file_io               | 0 B   |

**Total Databases Size - ps-80-gr-3-mysql:**

**Most Fragmented Tables by Freeable Size - ps-80-gr-3-mysql:**

| schema | service_name | table | Value |
|--------|--------------|-------|-------|
|--------|--------------|-------|-------|

### LARGEST TABLES

#### Largest Tables by Row Count

The estimated number of rows in the table from `information_schema.tables`.

#### Largest Tables by Size

The size of the table components from `information_schema.tables`.

### PIE

#### Total Database Size

The total size of the database: as data + index size, so freeable one.

## Most Fragmented Tables by Freeable Size

The list of 5 most fragmented tables ordered by their freeable size

### TABLE ACTIVITY

The next two graphs are available only for [Percona Server](#) and [MariaDB](#) and require `userstat` variable turned on.

#### ROWS READ

The number of rows read from the table, shown for the top 5 tables.

#### ROWS CHANGED

The number of rows changed in the table, shown for the top 5 tables.

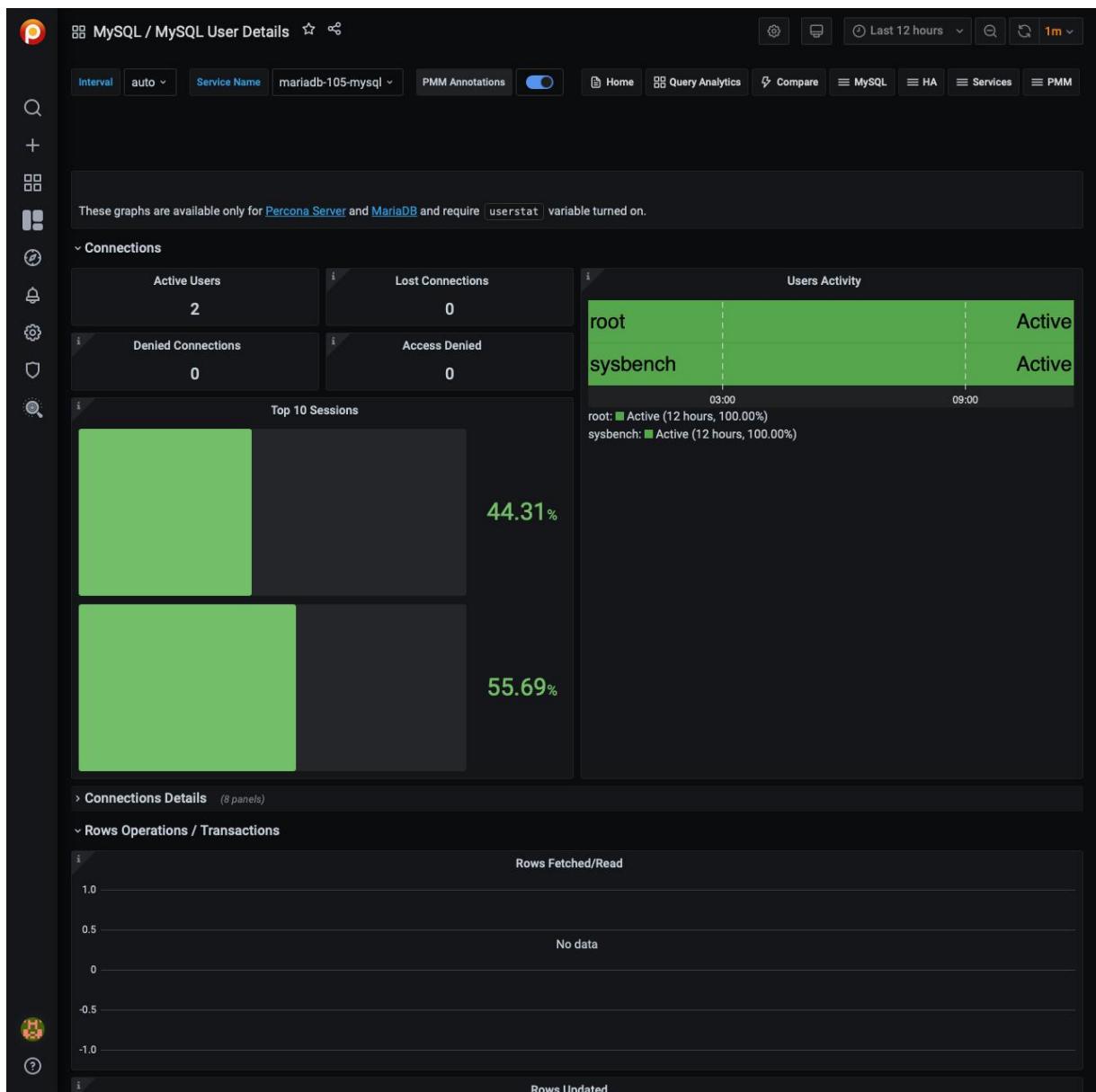
#### AUTO INCREMENT USAGE

The current value of an `auto_increment` column from `information_schema`, shown for the top 10 tables.

---

Last update: 2021-05-11

## MySQL User Details



This dashboard requires Percona Server for MySQL 5.1+ or MariaDB 10.1/10.2 with XtraDB. Also `userstat` should be enabled, for example with the `SET GLOBAL userstat=1` statement. See [Setting up MySQL](#).

Data is displayed for the 5 top users.

### Top Users by Connections Created

The number of times user's connections connected using SSL to the server.

### Top Users by Traffic

The number of bytes sent to the user's connections.

### Top Users by Rows Fetched/Read

The number of rows fetched by the user's connections.

### Top Users by Rows Updated

The number of rows updated by the user's connections.

### Top Users by Busy Time

The cumulative number of seconds there was activity on connections from the user.

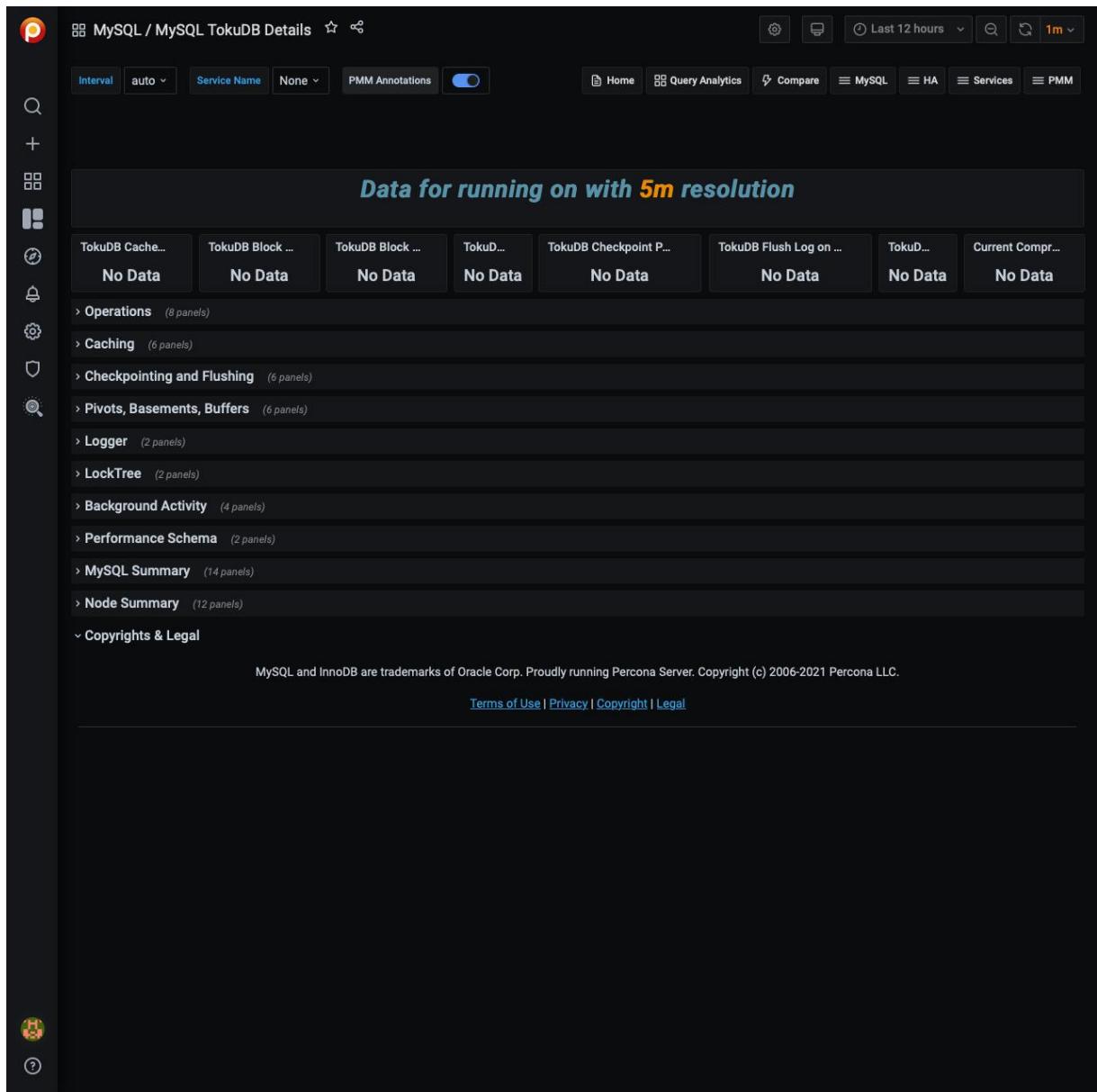
### Top Users by CPU Time

The cumulative CPU time elapsed, in seconds, while servicing connections of the user.

---

Last update: 2021-06-10

## MySQL TokuDB Details

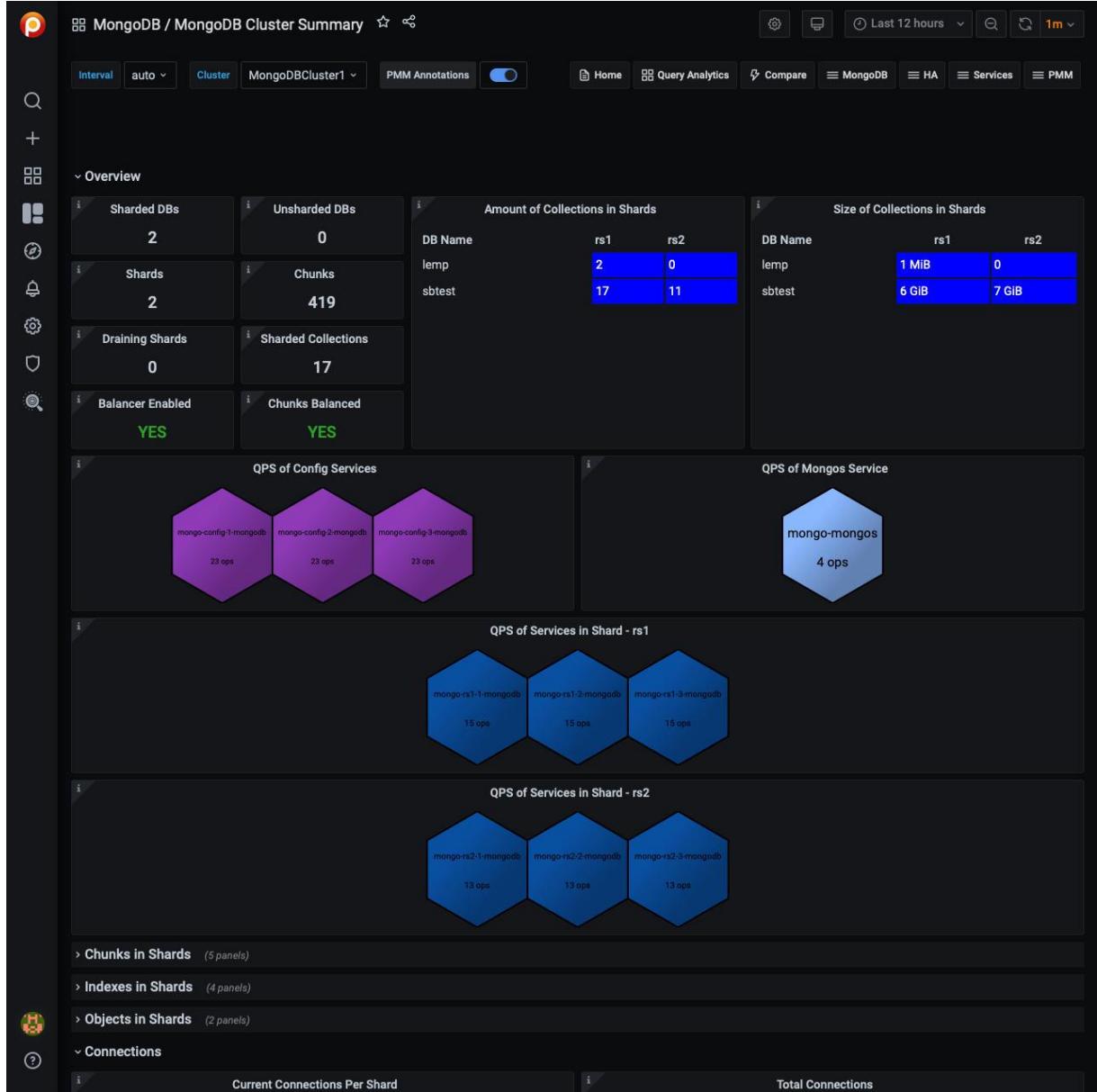


No description

Last update: 2021-05-11

## 5.4.8 MongoDB Dashboards

### MongoDB Cluster Summary



### CURRENT CONNECTIONS PER SHARD

TCP connections (Incoming) in mongod processes.

### TOTAL CONNECTIONS

Incoming connections to mongos nodes.

### CURSORS PER SHARD

The Cursor is a MongoDB Collection of the document which is returned upon the find method execution.

### MONGOS CURSORS

The Cursor is a MongoDB Collection of the document which is returned upon the find method execution.

**OPERATIONS PER SHARD**

Ops/sec, classified by legacy wire protocol type (`query`, `insert`, `update`, `delete`, `getmore`).

**TOTAL MONGOS OPERATIONS**

Ops/sec, classified by legacy wire protocol type (`query`, `insert`, `update`, `delete`, `getmore`).

**CHANGE LOG EVENTS**

Count, over last 10 minutes, of all types of configuration db changelog events.

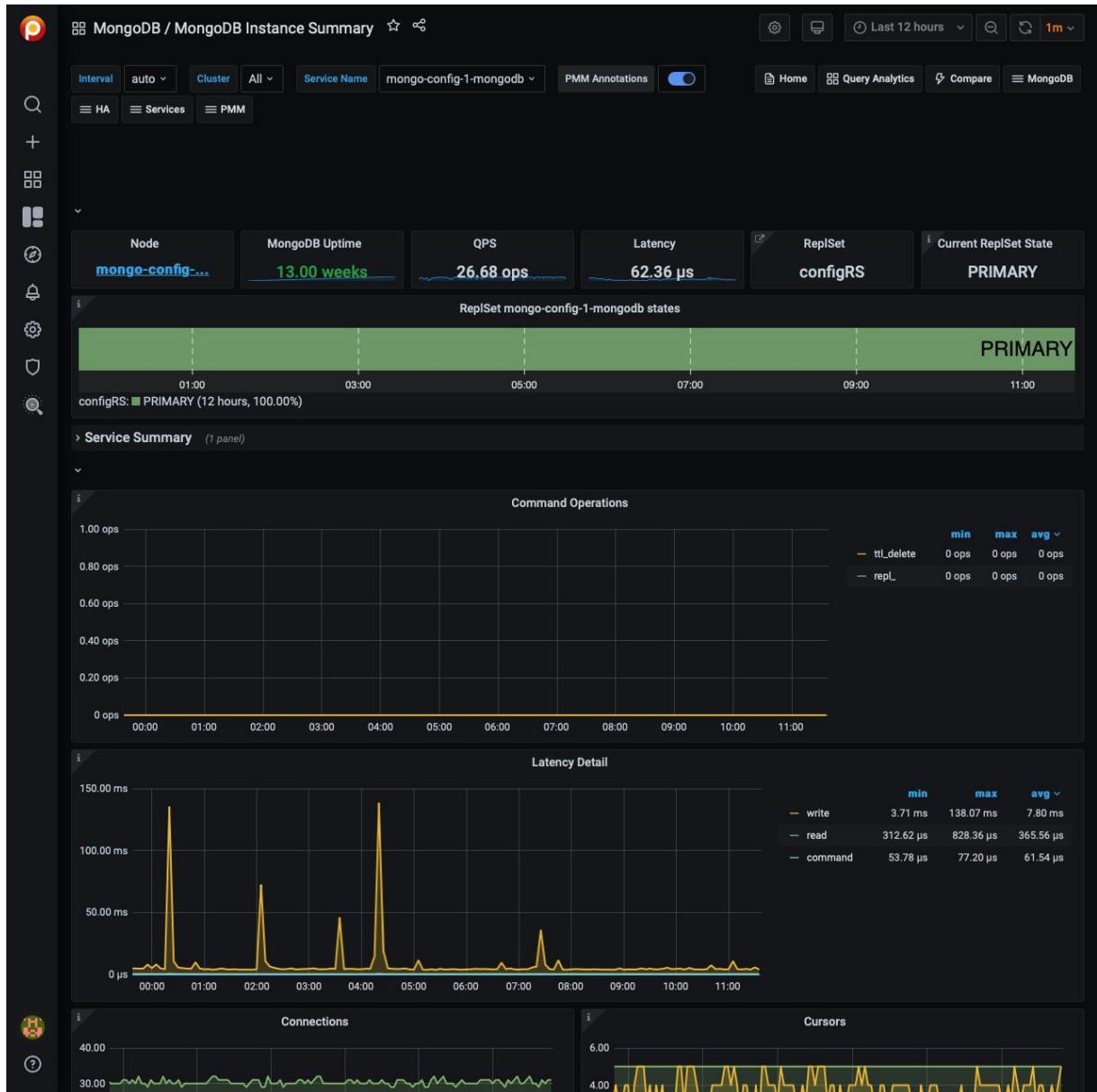
**OPLOG RANGE BY SET**

Timespan 'window' between oldest and newest ops in the Oplog collection.

---

Last update: 2021-05-11

## MongoDB Instance Summary



### COMMAND OPERATIONS

Ops or Replicated Ops/sec classified by legacy wire protocol type (query, insert, update, delete, getmore). And (from the internal TTL threads) the docs deletes/sec with TTL indexes.

### LATENCY DETAIL

Average latency of operations (classified by read, write, or (other) command)

### CONNECTIONS

TCP connections (Incoming)

### CURSORS

Open cursors. Includes idle cursors.

### DOCUMENT OPERATIONS

Docs per second inserted, updated, deleted or returned. (not 1-to-1 with operation counts.)

**QUEUED OPERATIONS**

Operations queued due to a lock.

**QUERY EFFICIENCY**

Ratio of Documents returned or Index entries scanned / full documents scanned

**SCANNED AND MOVED OBJECTS**

This panel shows the number of objects (both data (`scanned_objects`) and index (`scanned`)) as well as the number of documents that were moved to a new location due to the size of the document growing. Moved documents only apply to the MMAPv1 storage engine.

**GETLASTERROR WRITE TIME**

Legacy driver operation: Number of, and Sum of time spent, per second executing `getLastError` commands to confirm write concern.

**GETLASTERROR WRITE OPERATIONS**

Legacy driver operation: Number of `getLastError` commands that timed out trying to confirm write concern.

**ASSERT EVENTS**

This panel shows the number of assert events per second on average over the given time period. In most cases assertions are trivial, but you would want to check your log files if this counter spikes or is consistently high.

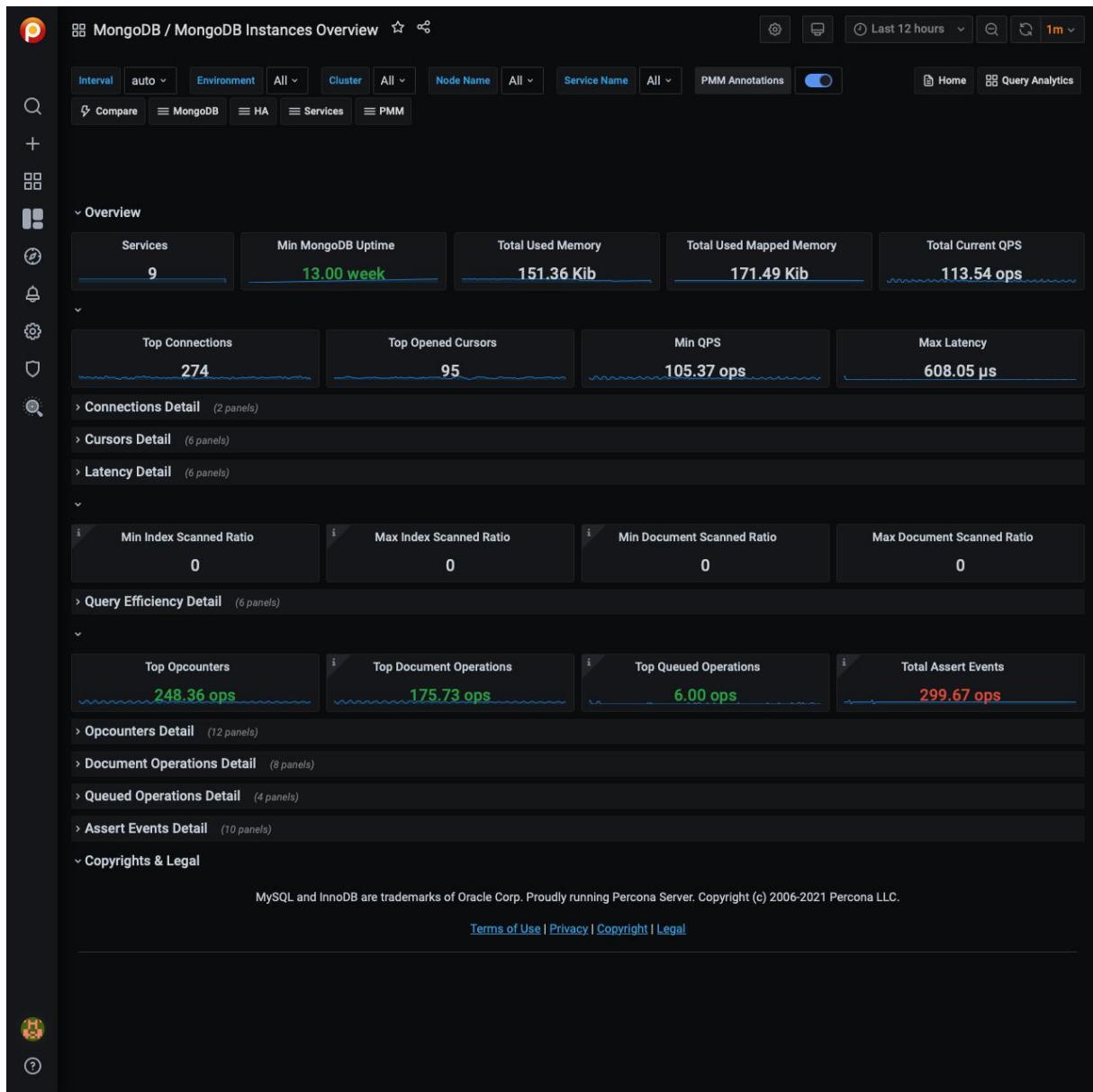
**PAGE FAULTS**

Unix or Window memory page faults. Not necessarily from MongoDB.

---

Last update: 2021-05-11

## MongoDB Instances Overview



This dashboard provides basic information about MongoDB instances.

### COMMAND OPERATIONS

Shows how many times a command is executed per second on average during the selected interval.

Look for peaks and drops and correlate them with other graphs.

### CONNECTIONS

Keep in mind the hard limit on the maximum number of connections set by your distribution.

Anything over 5,000 should be a concern, because the application may not close connections correctly.

### CURSORS

Helps identify why connections are increasing. Shows active cursors compared to cursors being automatically killed after 10 minutes due to an application not closing the connection.

#### DOCUMENT OPERATIONS

When used in combination with *Command Operations*, this graph can help identify *write amplification*. For example, when one `insert` or `update` command actually inserts or updates hundreds, thousands, or even millions of documents.

#### QUEUED OPERATIONS

Any number of queued operations for long periods of time is an indication of possible issues. Find the cause and fix it before requests get stuck in the queue.

#### GETLASTERROR WRITE TIME, GETLASTERROR WRITE OPERATIONS

This is useful for write-heavy workloads to understand how long it takes to verify writes and how many concurrent writes are occurring.

#### ASSERTS

Asserts are not important by themselves, but you can correlate spikes with other graphs.

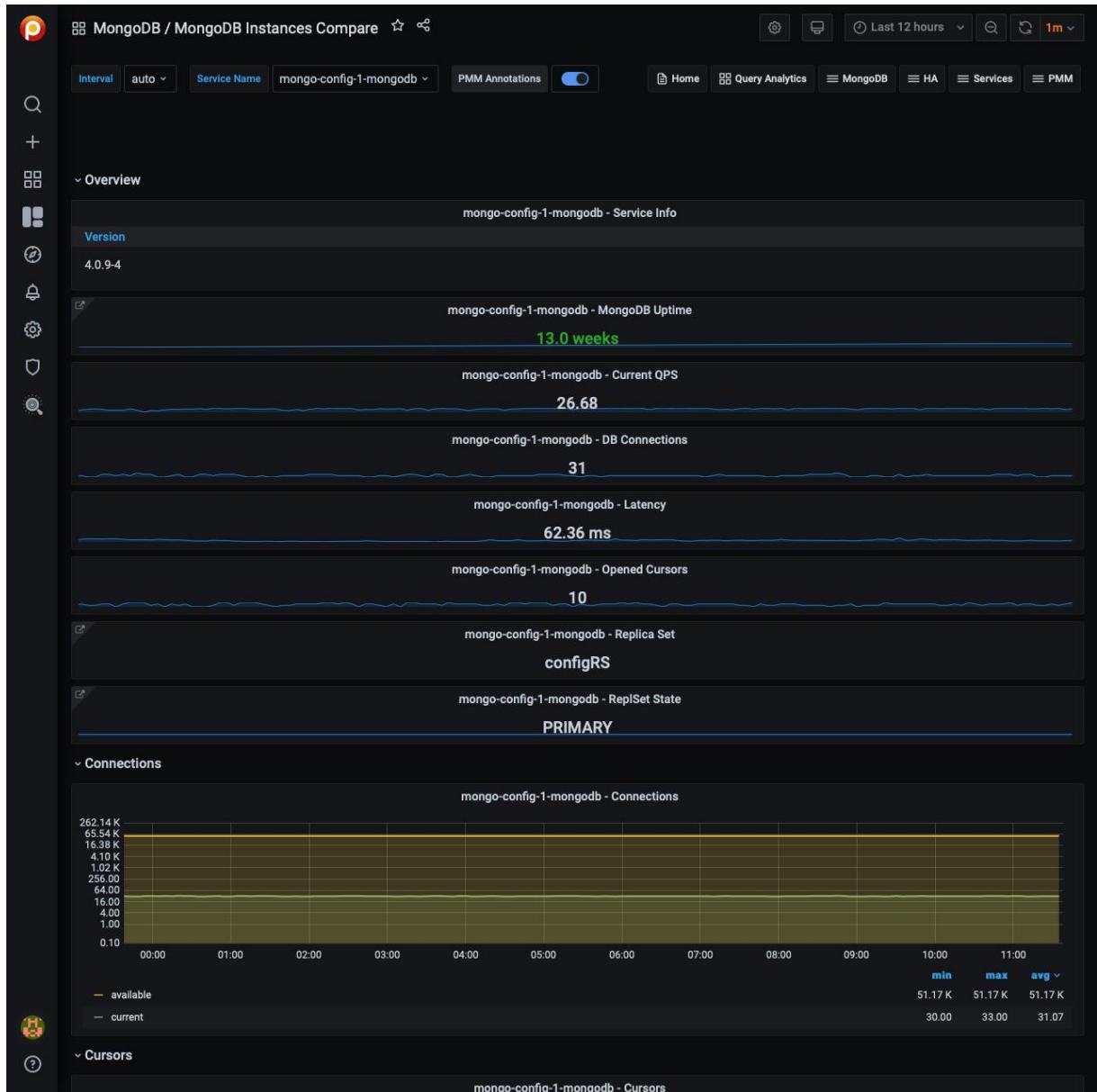
#### MEMORY FAULTS

Memory faults indicate that requests are processed from disk either because an index is missing or there is not enough memory for the data set. Consider increasing memory or sharding out.

---

Last update: 2021-06-03

## MongoDB Instances Compare



### CONNECTIONS

No description

### CURSORS

No description

### LATENCY

Average latency of operations (classified by read, write, or (other) command)

### SCAN RATIOS

Ratio of index entries scanned or whole docs scanned / number of documents returned

### INDEX FILTERING EFFECTIVENESS

No description

**REQUESTS**

Ops/sec (classified by (legacy) wire protocol request type)

**DOCUMENT OPERATIONS**

Documents inserted/updated/deleted or returned per sec

**QUEUED OPERATIONS**

The number of operations that are currently queued and waiting for a lock

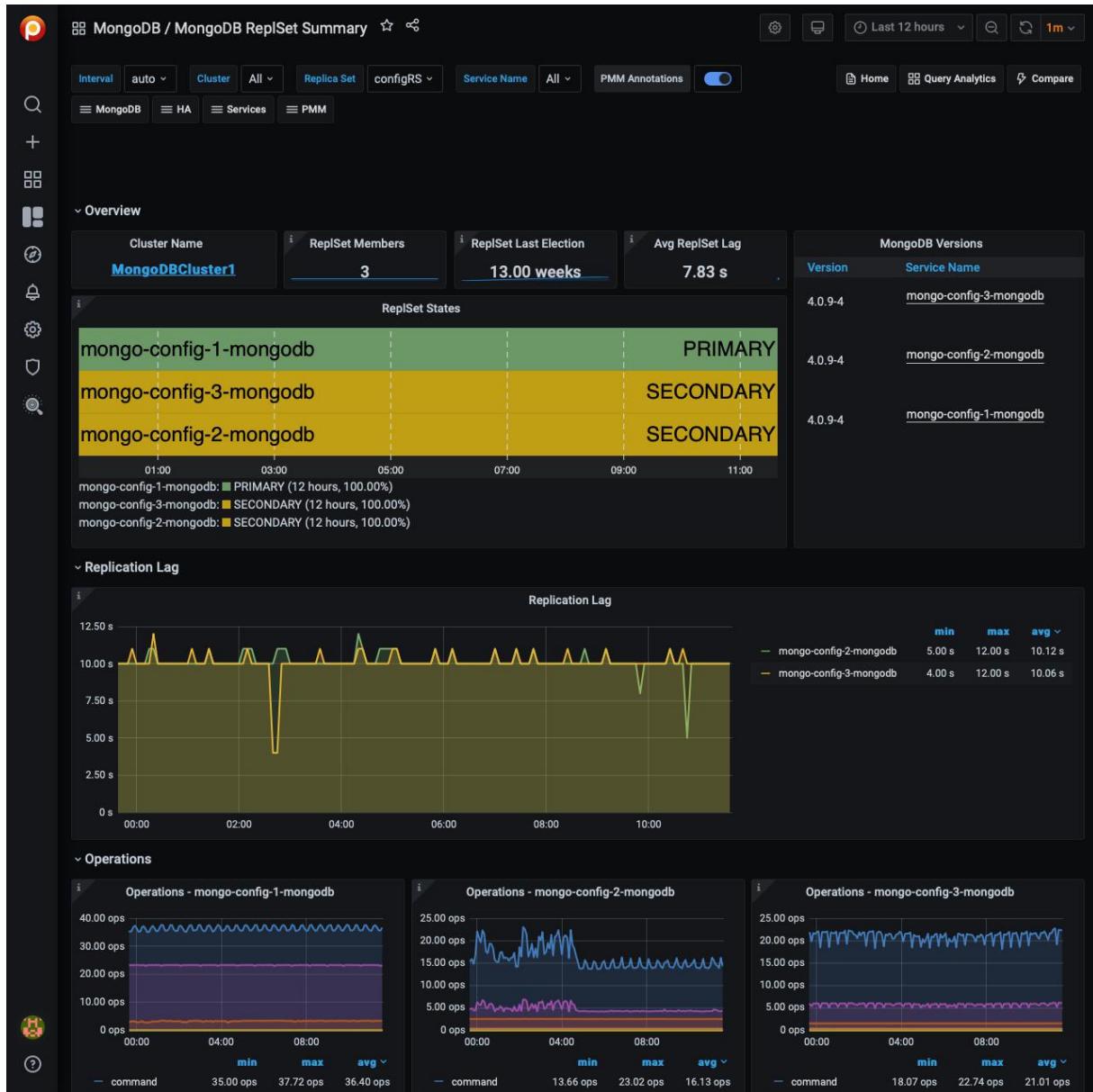
**USED MEMORY**

No description

---

Last update: 2021-05-11

## MongoDB RepSet Summary



### REPLICATION LAG

MongoDB replication lag occurs when the secondary node cannot replicate data fast enough to keep up with the rate that data is being written to the primary node. It could be caused by something as simple as network latency, packet loss within your network, or a routing issue.

### OPERATIONS - BY SERVICE NAME

Operations are classified by legacy wire protocol type (insert, update, and delete only).

### MAX MEMBER PING TIME - BY SERVICE NAME

This metric can show a correlation with the replication lag value.

### MAX HEARTBEAT TIME

Time span between now and last heartbeat from replicaset members.

**ELECTIONS**

Count of elections. Usually zero; 1 count by each healthy node will appear in each election. Happens when the primary role changes due to either normal maintenance or trouble events.

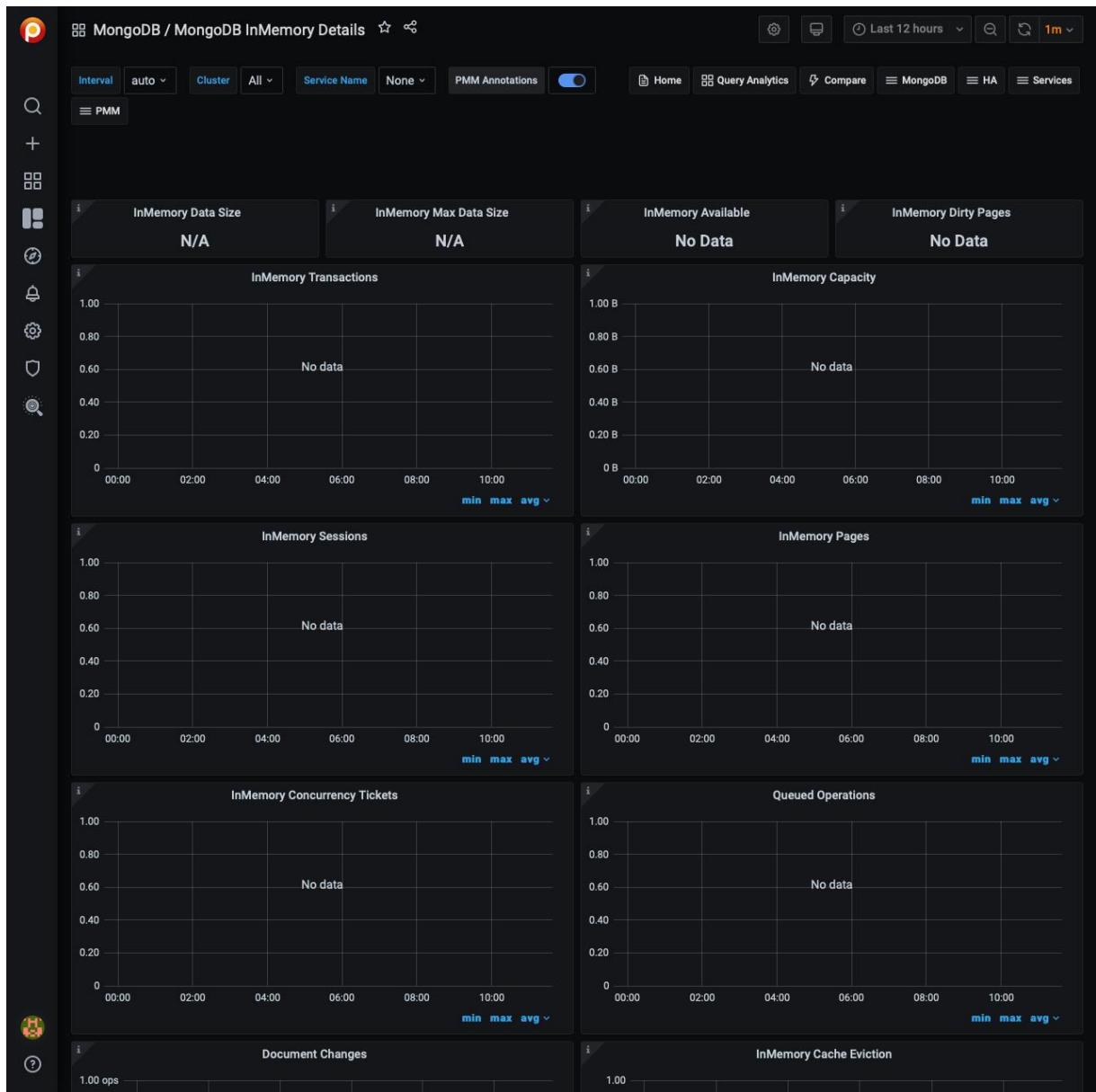
**OPLOG RECOVERY WINDOW - BY SERVICE NAME**

Timespan 'window' between newest and the oldest op in the Oplog collection.

---

Last update: 2021-05-11

## MongoDB InMemory Details



### INMEMORY TRANSACTIONS

WiredTiger internal transactions

### INMEMORY CAPACITY

Configured max and current size of the WiredTiger cache.

### INMEMORY SESSIONS

Internal WiredTiger storage engine cursors and sessions currently open.

### INMEMORY PAGES

Pages in the WiredTiger cache

### INMEMORY CONCURRENCY TICKETS

A WT 'ticket' is assigned out for every operation running simultaneously in the WT storage engine. "Tickets available" = hard coded high value - "Tickets Out".

**QUEUED OPERATIONS**

Operations queued due to a lock

**DOCUMENT CHANGES**

Mixed metrics: Docs per second inserted, updated, deleted or returned on any type of node (primary or secondary); + replicated write Ops/sec; + TTL deletes per second.

**INMEMORY CACHE EVICTION**

This panel shows the number of pages that have been evicted from the WiredTiger cache for the given time period. The InMemory storage engine only evicts modified pages which signals a compaction of the data and removal of the dirty pages.

**SCANNED AND MOVED OBJECTS**

This panel shows the number of objects (both data (`scanned_objects`) and index (`scanned`)) as well as the number of documents that were moved to a new location due to the size of the document growing. Moved documents only apply to the MMAPv1 storage engine.

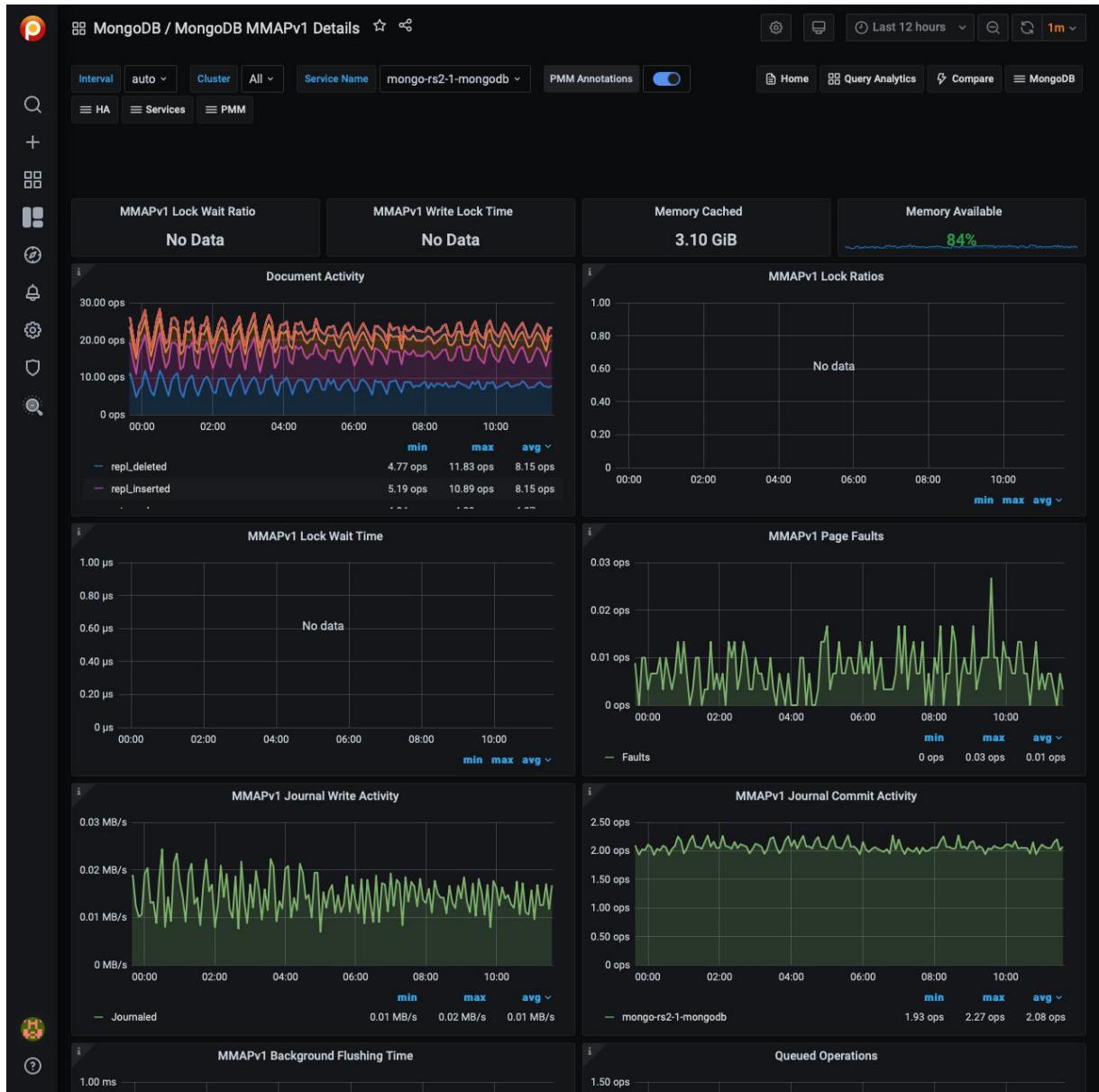
**PAGE FAULTS**

Unix or Window memory page faults. Not necessarily from MongoDB.

---

Last update: 2021-05-11

## MongoDB MMAPv1 Details



### DOCUMENT ACTIVITY

Docs per second inserted, updated, deleted or returned. Also showing replicated write ops and internal TTL index deletes.

### MMAPV1 LOCK WAIT TIME

Time spent per second waiting to acquire locks.

### MMAPV1 PAGE FAULTS

Unix or Window memory page faults. Not necessarily from MongoDB.

### MMAPV1 JOURNAL WRITE ACTIVITY

MB processed through the journal in memory.

### MMAPV1 JOURNAL COMMIT ACTIVITY

MB committed to disk for the journal.

**MMAPV1 BACKGROUND FLUSHING TIME**

Average time in ms, over full uptime of `mongod` process, the MMAP background flushes have taken.

**QUEUED OPERATIONS**

Queue size of ops waiting to be submitted to storage engine layer. (see WiredTiger concurrency tickets for number of ops being processed simultaneously in storage engine layer.)

**CLIENT OPERATIONS**

Ops and Replicated Ops/sec, classified by legacy wire protocol type (`query`, `insert`, `update`, `delete`, `getmore`).

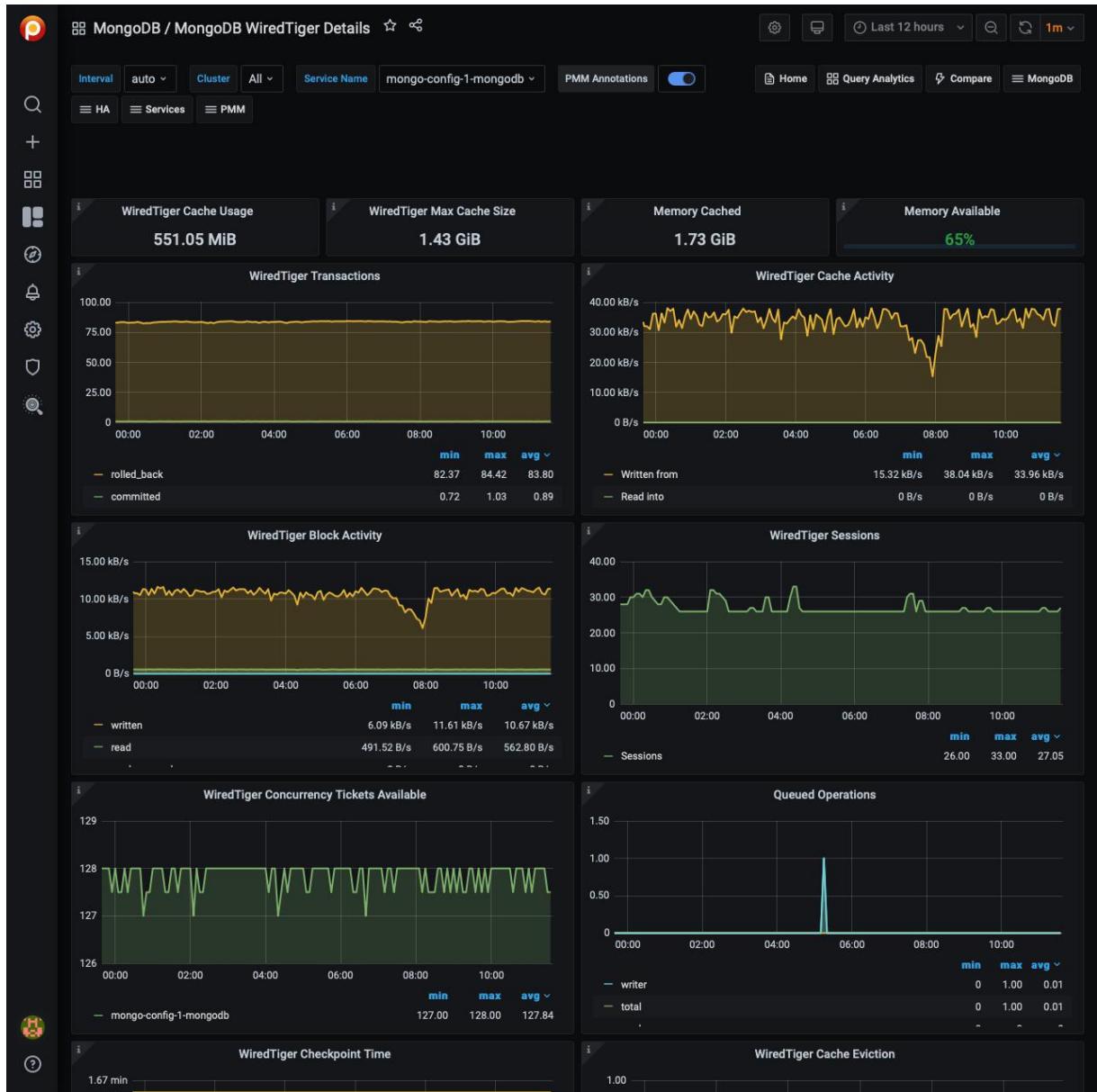
**SCANNED AND MOVED OBJECTS**

This panel shows the number of objects (both data (`scanned_objects`) and index (`scanned`)) as well as the number of documents that were moved to a new location due to the size of the document growing. Moved documents only apply to the MMAPv1 storage engine.

---

Last update: 2021-05-11

## MongoDB WiredTiger Details



### WIREDTIGER TRANSACTIONS

WiredTiger internal transactions

### WIREDTIGER CACHE ACTIVITY

Data volume transferred per second between the WT cache and data files. Writes out always imply disk; Reads are often from OS file buffer cache already in RAM, but disk if not.

### WIREDTIGER BLOCK ACTIVITY

Data volume handled by the WT block manager per second

### WIREDTIGER SESSIONS

Internal WT storage engine cursors and sessions currently open

**WIREDTIGER CONCURRENCY TICKETS AVAILABLE**

A WT ‘ticket’ is assigned out for every operation running simultaneously in the WT storage engine. “Available” = hard-coded high value - “Out”.

**QUEUED OPERATIONS**

Operations queued due to a lock.

**WIREDTIGER CHECKPOINT TIME**

The time spent in WT checkpoint phase. Warning: This calculation averages the cyclical event (default: 1 min) execution to a per-second value.

**WIREDTIGER CACHE EVICTION**

Least-recently used pages being evicted due to WT cache becoming full.

**WIREDTIGER CACHE CAPACITY**

Configured max and current size of the WT cache.

**WIREDTIGER CACHE PAGES****WIREDTIGER LOG OPERATIONS**

WT internal write-ahead log operations.

**WIREDTIGER LOG ACTIVITY**

Data volume moved per second in WT internal write-ahead log.

**WIREDTIGER LOG RECORDS**

Number of records appended per second in WT internal log.

**DOCUMENT CHANGES**

Mixed metrics: Docs per second inserted, updated, deleted or returned on any type of node (primary or secondary); + replicated write Ops/sec; + TTL deletes per second.

**SCANNED AND MOVED OBJECTS**

This panel shows the number of objects (both data ( `scanned_objects` ) and index ( `scanned` )) as well as the number of documents that were moved to a new location due to the size of the document growing. Moved documents only apply to the MMAPv1 storage engine.

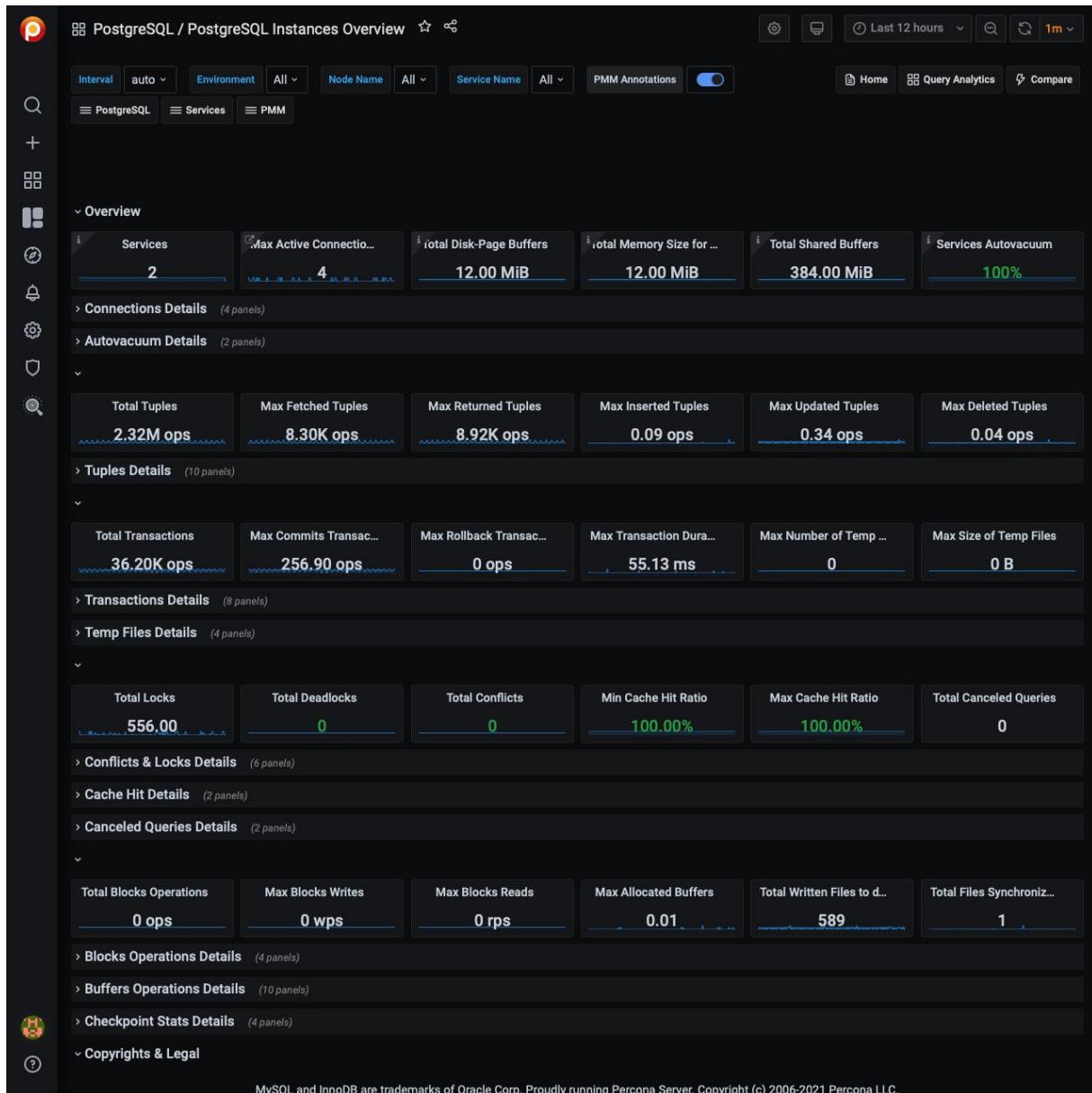
**PAGE FAULTS**

Unix or Window memory page faults. Not necessarily from MongoDB.

Last update: 2021-05-11

## 5.4.9 PostgreSQL Dashboards

### PostgreSQL Instances Overview



#### CONNECTED

Reports whether PMM Server can connect to the PostgreSQL instance.

#### VERSION

The version of the PostgreSQL instance.

#### SHARED BUFFERS

Defines the amount of memory the database server uses for shared memory buffers. Default is 128MB. Guidance on tuning is 25% of RAM, but generally doesn't exceed 40%.

**DISK-PAGE BUFFERS**

The setting `wal_buffers` defines how much memory is used for caching the write-ahead log entries. Generally this value is small (3% of `shared_buffers` value), but it may need to be modified for heavily loaded servers.

**MEMORY SIZE FOR EACH SORT**

The parameter `work_mem` defines the amount of memory assigned for internal sort operations and hash tables before writing to temporary disk files. The default is `4MB`.

**DISK CACHE SIZE**

PostgreSQL's `effective_cache_size` variable tunes how much RAM you expect to be available for disk caching. Generally adding Linux free+cached will give you a good idea. This value is used by the query planner whether plans will fit in memory, and when defined too low, can lead to some plans rejecting certain indexes.

**AUTOVACUUM**

Whether autovacuum process is enabled or not. Generally the solution is to vacuum more often, not less.

**POSTGRESQL CONNECTIONS****Max Connections**

The maximum number of client connections allowed. Change this value with care as there are some memory resources that are allocated on a per-client basis, so setting `max_connections` higher will generally increase overall PostgreSQL memory usage.

**Connections**

The number of connection attempts (successful or not) to the PostgreSQL server.

**Active Connections**

The number of open connections to the PostgreSQL server.

**POSTGRESQL TUPLES****Tuples**

The total number of rows processed by PostgreSQL server: fetched, returned, inserted, updated, and deleted.

**Read Tuple Activity**

The number of rows read from the database: as returned so fetched ones.

**Tuples Changed per 5 min**

The number of rows changed in the last 5 minutes: inserted, updated, and deleted ones.

**POSTGRESQL TRANSACTIONS****Transactions**

The total number of transactions that have been either been committed or rolled back.

**Duration of Transactions**

Maximum duration in seconds any active transaction has been running.

**TEMP FILES****Number of Temp Files**

The number of temporary files created by queries.

**Size of Temp files**

The total amount of data written to temporary files by queries in bytes.

All temporary files are taken into account by these two gauges, regardless of why the temporary file was created (e.g., sorting or hashing), and regardless of the `log_temp_files` setting.

**CONFLICTS AND LOCKS****Conflicts/Deadlocks**

The number of queries canceled due to conflicts with recovery in the database (due to dropped tablespaces, lock timeouts, old snapshots, pinned buffers, or deadlocks).

**Number of Locks**

The number of deadlocks detected by PostgreSQL.

**BUFFERS AND BLOCKS OPERATIONS****Operations with Blocks**

The time spent reading and writing data file blocks by back ends, in milliseconds.

**Tip**

Capturing read and write time statistics is possible only if `track_io_timing` setting is enabled. This can be done either in configuration file or with the following query executed on the running system:

```
ALTER SYSTEM SET track_io_timing=ON;
SELECT pg_reload_conf();
```

**Buffers**

The number of buffers allocated by PostgreSQL.

**CANCELED QUERIES**

The number of queries that have been canceled due to dropped tablespaces, lock timeouts, old snapshots, pinned buffers, and deadlocks.

Data shown by this gauge are based on the `pg_stat_database_conflicts` view.

**CACHE HIT RATIO**

The number of times disk blocks were found already in the buffer cache, so that a read was not necessary.

This only includes hits in the PostgreSQL buffer cache, not the operating system's file system cache.

**CHECKPOINT STATS**

The total amount of time that has been spent in the portion of checkpoint processing where files are either written or synchronized to disk, in milliseconds.

**POSTGRESQL SETTINGS**

The list of all settings of the PostgreSQL server.

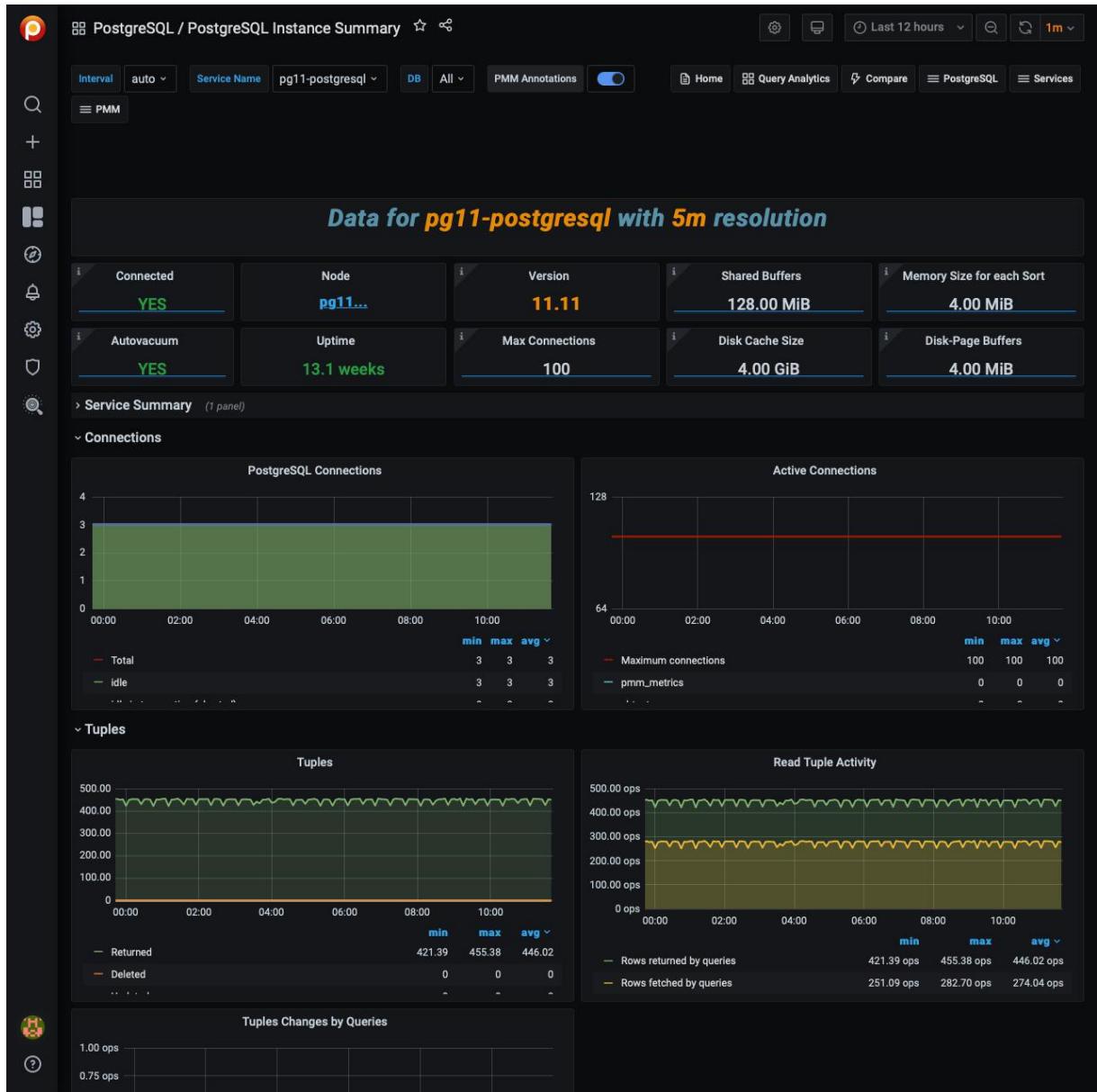
**SYSTEM SUMMARY**

This section contains the following system parameters of the PostgreSQL server: CPU Usage, CPU Saturation and Max Core Usage, Disk I/O Activity, and Network Traffic.

---

Last update: 2021-06-15

## PostgreSQL Instance Summary



### NUMBER OF TEMP FILES

Cumulative number of temporary files created by queries in this database since service start. All temporary files are counted, regardless of why the temporary file was created (e.g., sorting or hashing), and regardless of the `log_temp_files` setting.

### SIZE OF TEMP FILES

Cumulative amount of data written to temporary files by queries in this database since service start. All temporary files are counted, regardless of why the temporary file was created, and regardless of the `log_temp_files` setting.

### TEMP FILES ACTIVITY

Number of temporary files created by queries in this database. All temporary files are counted, regardless of why the temporary file was created (e.g., sorting or hashing), and regardless of the `log_temp_files` setting.

**TEMP FILES UTILIZATION**

Total amount of data written to temporary files by queries in this database. All temporary files are counted, regardless of why the temporary file was created, and regardless of the `log_temp_files` setting.

**CANCELED QUERIES**

Based on `pg_stat_database_conflicts` view

---

Last update: 2021-05-11

## PostgreSQL Instances Compare

**PostgreSQL / PostgreSQL Instances Compare**

Interval: auto | Node Name: All | Service Name: All | PMM Annotations:

Last 12 hours | 1m

Home | Query Analytics | PostgreSQL | Services | PMM

**Overview**

| pg11-postgresql - Service Info                                                                                             | pg12-postgresql - Service Info                                                                                           | pmm-server-postgresql - Service Info                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>Version</b><br>PostgreSQL 11.11 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-44), 64-bit | <b>Version</b><br>PostgreSQL 12.7 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 7.3.1 20180303 (Red Hat 7.3.1-5), 64-bit | <b>Version</b><br>PostgreSQL 11.12 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 7.3.1 20180303 (Red Hat 7.3.1-5), 64-bit |
| <b>pg11-postgresql - Shared Buffers</b><br><b>128.00 MiB</b>                                                               | <b>pg12-postgresql - Shared Buffers</b><br><b>128.00 MiB</b>                                                             | <b>pmm-server-postgresql - Shared Buffers</b><br><b>128.00 MiB</b>                                                        |
| <b>pg11-postgresql - Disk-Page Buffers</b><br><b>4.00 MiB</b>                                                              | <b>pg12-postgresql - Disk-Page Buffers</b><br><b>4.00 MiB</b>                                                            | <b>pmm-server-postgresql - Disk-Page Buffers</b><br><b>4.00 MiB</b>                                                       |
| <b>pg11-postgresql - Memory Size for each Sort</b><br><b>4.00 MiB</b>                                                      | <b>pg12-postgresql - Memory Size for each Sort</b><br><b>4.00 MiB</b>                                                    | <b>pmm-server-postgresql - Memory Size for each Sort</b><br><b>4.00 MiB</b>                                               |
| <b>pg11-postgresql - Disk Cache Size</b><br><b>4.00 GiB</b>                                                                | <b>pg12-postgresql - Disk Cache Size</b><br><b>4.00 GiB</b>                                                              | <b>pmm-server-postgresql - Disk Cache Size</b><br><b>4.00 GiB</b>                                                         |
| <b>pg11-postgresql - Autovacuum</b><br><b>YES</b>                                                                          | <b>pg12-postgresql - Autovacuum</b><br><b>YES</b>                                                                        | <b>pmm-server-postgresql - Autovacuum</b><br><b>YES</b>                                                                   |

> **Connections** (4 panels)

> **Tuples** (6 panels)

> **Transactions** (4 panels)

> **Temp Files** (4 panels)

> **Conflicts & Locks** (4 panels)

> **Buffers & Blocks Operations** (4 panels)

> **Others** (6 panels)

> **Copyrights & Legal**

MySQL and InnoDB are trademarks of Oracle Corp. Proudly running Percona Server. Copyright (c) 2006-2021 Percona LLC.

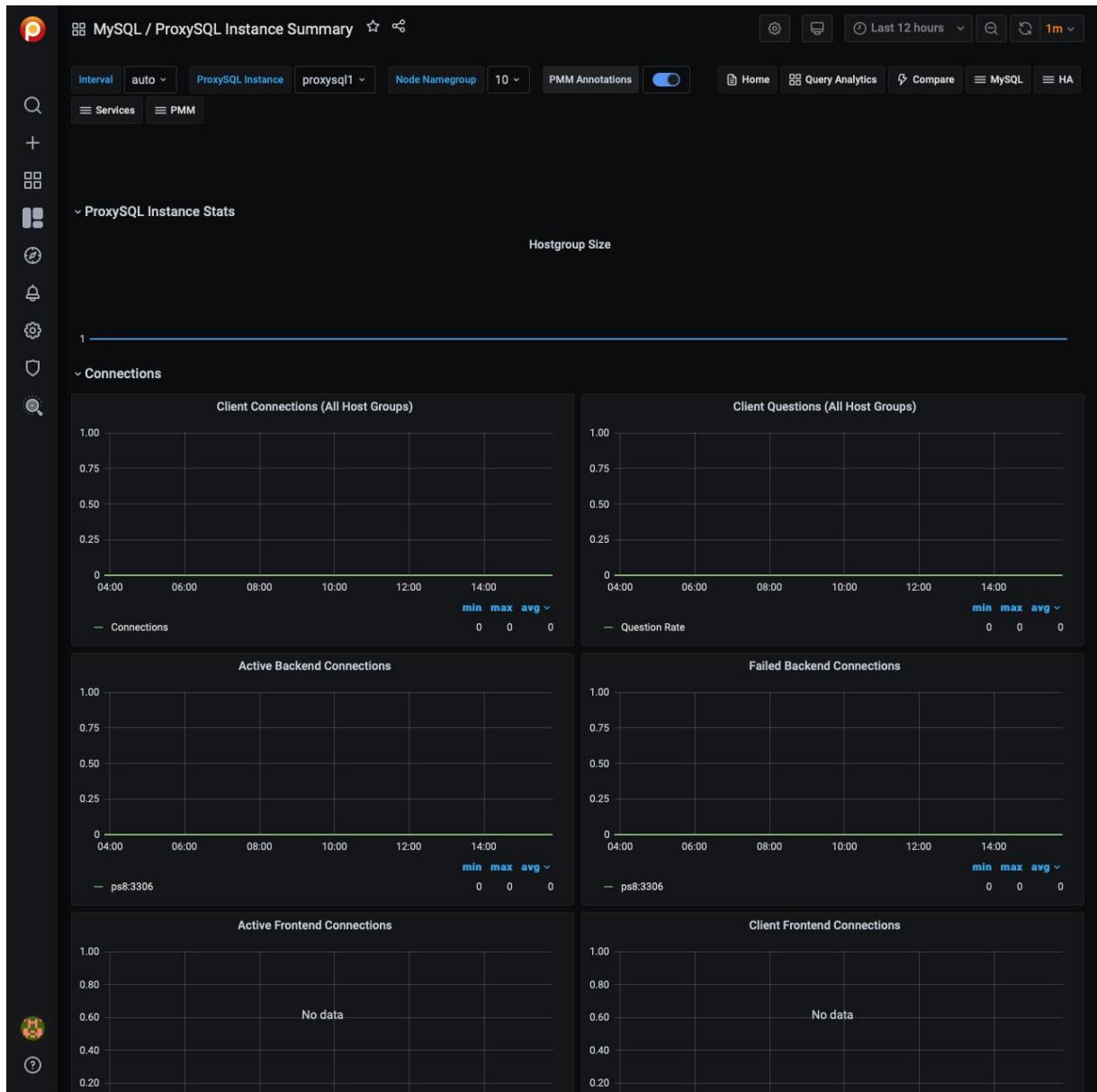
[Terms of Use](#) | [Privacy](#) | [Copyright](#) | [Legal](#)

No description

Last update: 2021-06-03

## 5.4.10 ProxySQL Dashboards

### ProxySQL Instance Summary



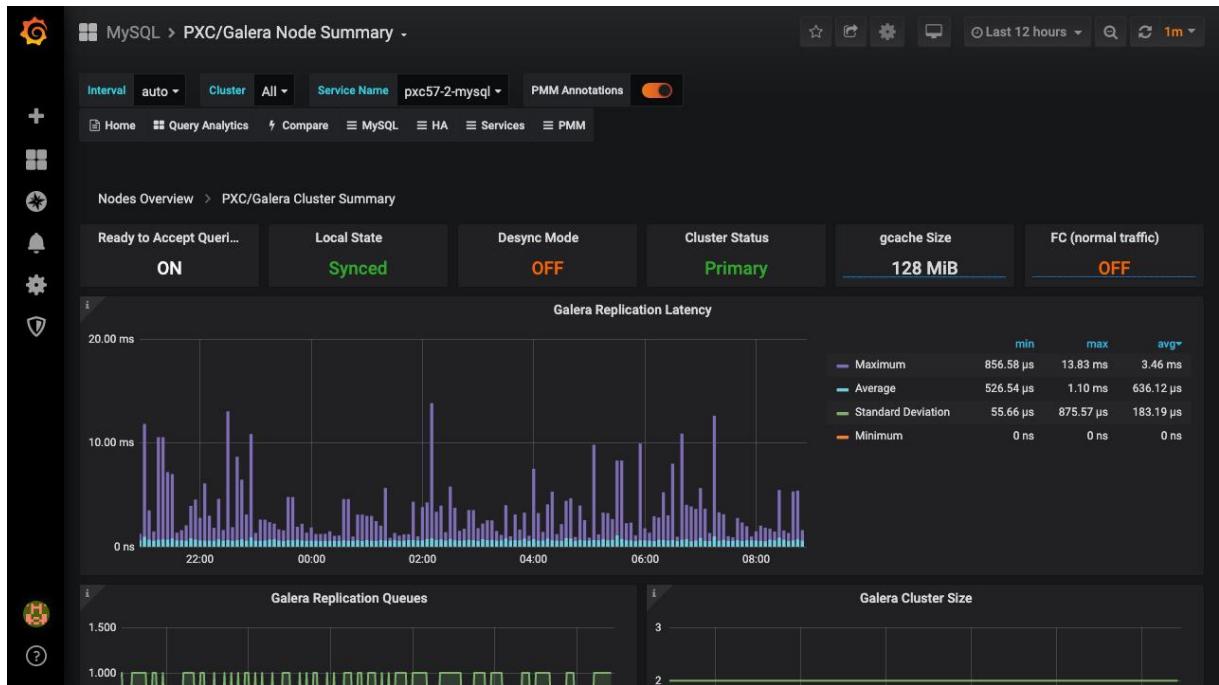
### NETWORK TRAFFIC

Network traffic refers to the amount of data moving across a network at a given point in time.

Last update: 2021-05-11

## 5.4.11 HA Dashboards

### PXC/Galera Node Summary



#### GALERA REPPLICATION LATENCY

Shows figures for the replication latency on group communication. It measures latency from the time point when a message is sent out to the time point when a message is received. As replication is a group operation, this essentially gives you the slowest ACK and longest RTT in the cluster.

#### GALERA REPPLICATION QUEUES

Shows the length of receive and send queues.

#### GALERA CLUSTER SIZE

Shows the number of members currently connected to the cluster.

#### GALERA FLOW CONTROL

Shows the number of `FC_PAUSE` events sent/received. They are sent by a node when its replication queue gets too full. If a node is sending out FC messages it indicates a problem.

#### GALERA PARALLELIZATION EFFICIENCY

Shows the average distances between highest and lowest seqno that are concurrently applied, committed and can be possibly applied in parallel (potential degree of parallelization).

#### GALERA WRITING CONFLICTS

Shows the number of local transactions being committed on this node that failed certification (some other node had a commit that conflicted with ours) – client received deadlock error on commit and also the number of local transactions in flight on this node that were aborted because they locked something an applier thread needed – deadlock error anywhere in an open transaction. Spikes in the graph may indicate writing to the same table potentially the same rows from 2 nodes.

**AVAILABLE DOWNTIME BEFORE SST REQUIRED**

Shows for how long the node can be taken out of the cluster before SST is required. SST is a full state transfer method.

**GALERA WRITESET COUNT**

Shows the count of transactions received from the cluster (any other node) and replicated to the cluster (from this node).

**GALERA WRITESET SIZE**

Shows the average transaction size received/replicated.

**GALERA WRITESET TRAFFIC**

Shows the bytes of data received from the cluster (any other node) and replicated to the cluster (from this node).

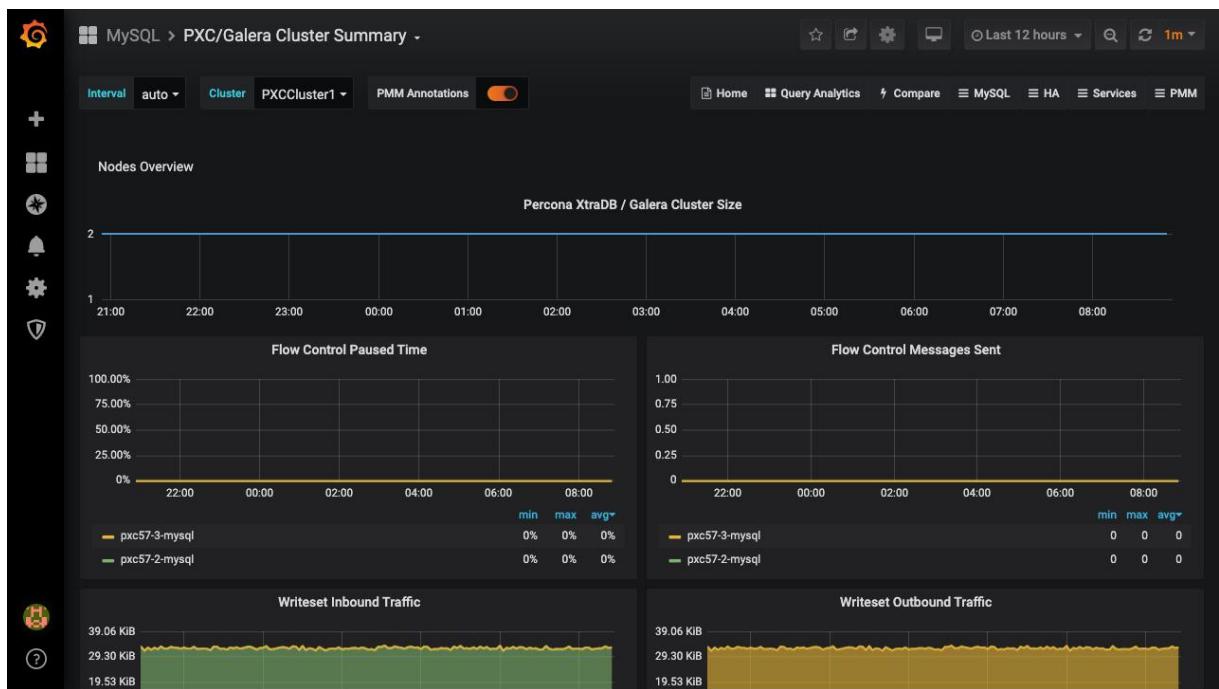
**GALERA NETWORK USAGE HOURLY**

Shows the bytes of data received from the cluster (any other node) and replicated to the cluster (from this node).

---

Last update: 2021-05-11

## PXC/Galera Cluster Summary

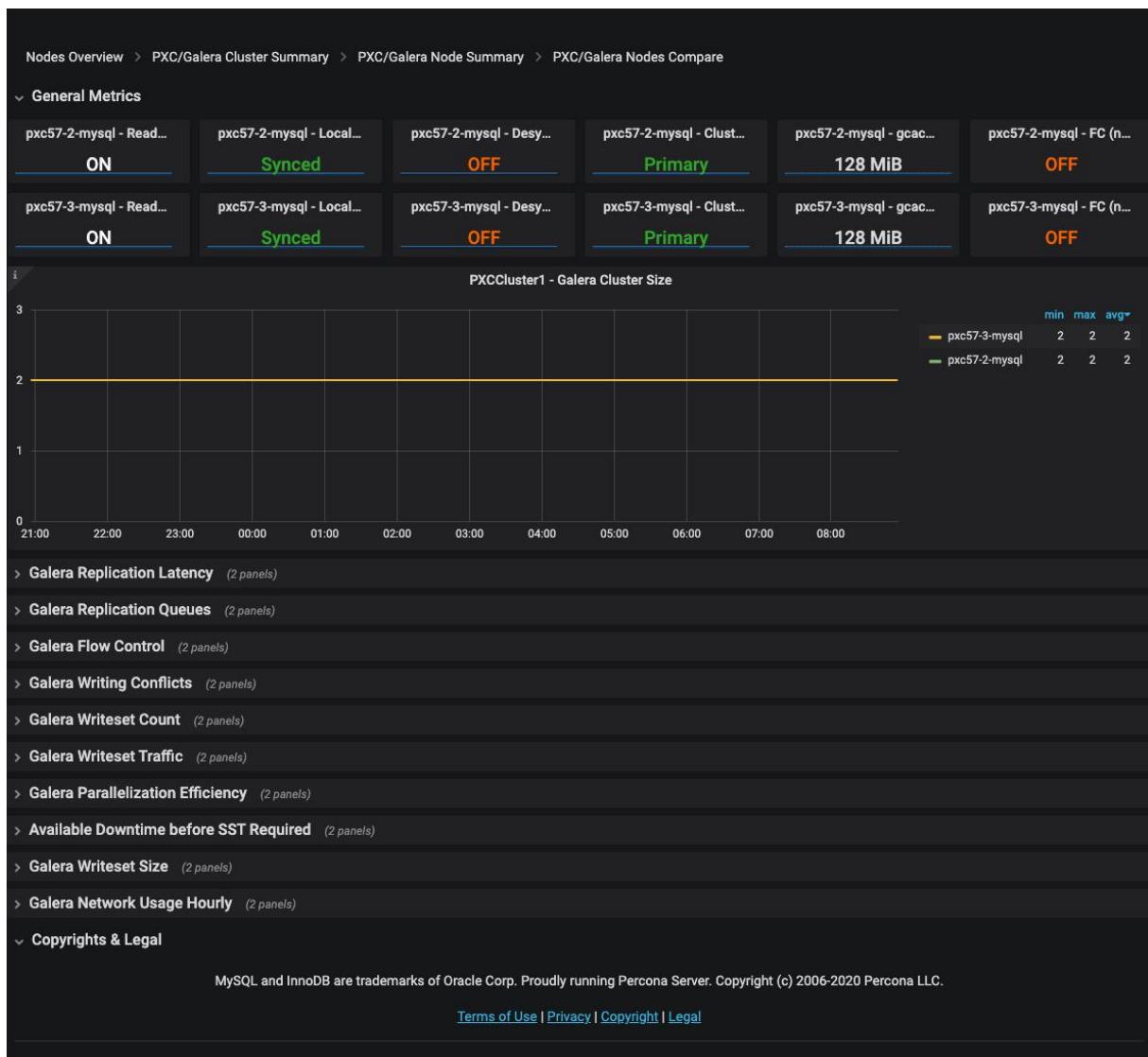


No description

---

Last update: 2021-05-11

## PXC/Galera Nodes Compare



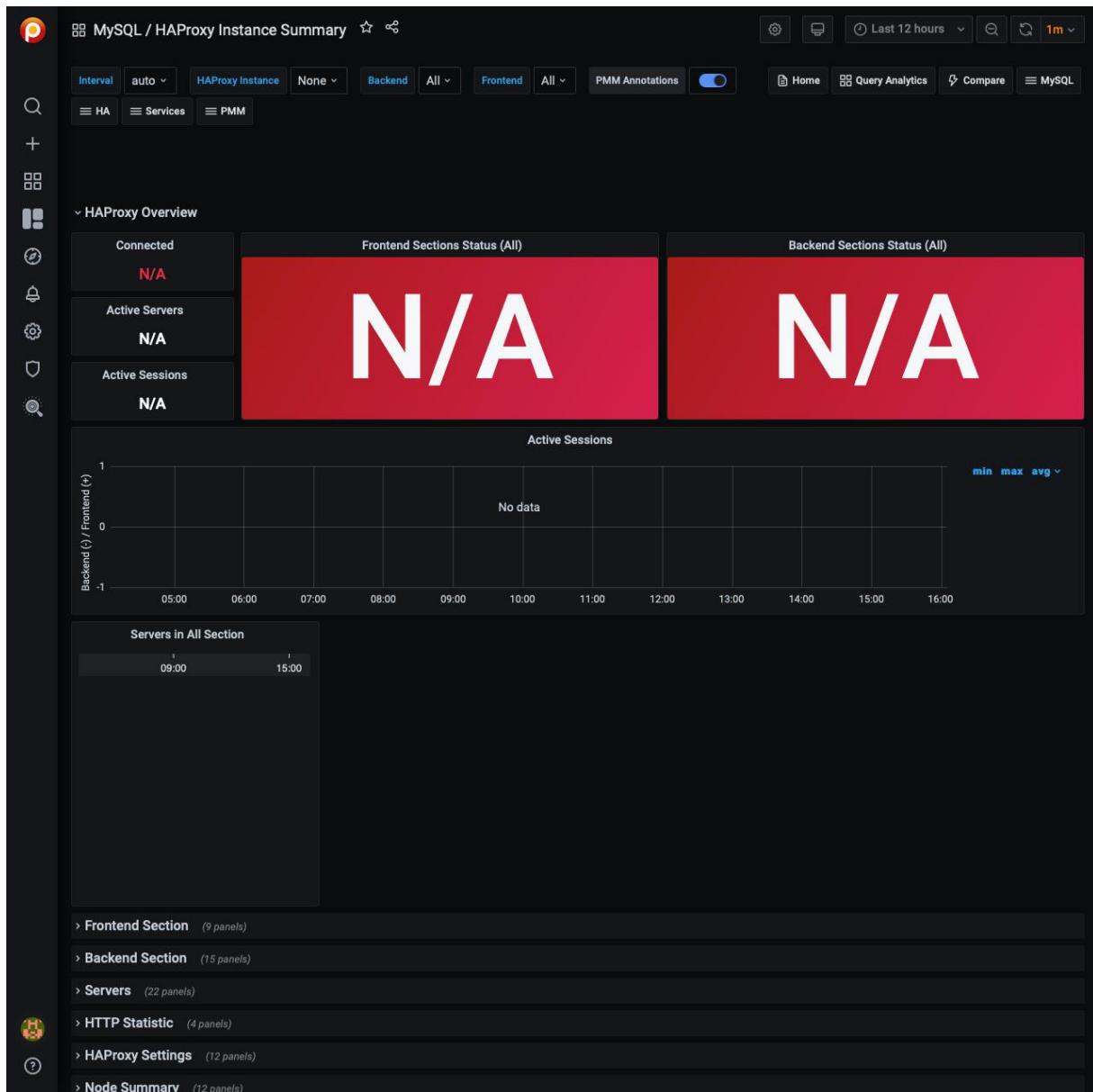
\$CLUSTER - GALERA CLUSTER SIZE

Shows the number of members currently connected to the cluster.

---

Last update: 2021-05-11

## HAPerxy Instance Summary



No description.

Last update: 2021-06-03

## 5.5 Commands

### 5.5.1 Commands

- **pmm-admin** – Command line tool for configuring and administering PMM
  - **pmm-agent** – Daemon process, communicating between PMM Client and PMM Server
- 

Last update: 2021-03-11

## 5.5.2 pmm-admin - PMM Administration Tool

### NAME

`pmm-admin` - Administer PMM

### SYNOPSIS

```
pmm-admin [FLAGS]

pmm-admin config [FLAGS] --server-url=server-url

pmm-admin add DATABASE [FLAGS] [NAME] [ADDRESS]
DATABASE:=[MongoDB | MySQL | PostgreSQL | ProxySQL]

pmm-admin add external [FLAGS] [NAME] [ADDRESS] (CAUTION: Technical preview feature)

pmm-admin add haproxy [FLAGS] [NAME]

pmm-admin add external [FLAGS] [NAME] [ADDRESS]

pmm-admin add external-serverless [FLAGS] [NAME] [ADDRESS]

pmm-admin remove [FLAGS] service-type [service-name]

pmm-admin register [FLAGS] [node-address] [node-type] [node-name]

pmm-admin list [FLAGS] [node-address]

pmm-admin status [FLAGS] [node-address]

pmm-admin summary [FLAGS] [node-address]

pmm-admin annotate [--node|--service] [--tags <tags>] [node-name|service-name]

pmm-admin help [COMMAND]
```

### DESCRIPTION

`pmm-admin` is a command-line tool for administering PMM using a set of COMMAND keywords and associated FLAGS.

PMM communicates with the PMM Server via a PMM agent process.

### COMMON FLAGS

`-h, --help`

Show help and exit.

`--help-long`

Show extended help and exit.

`--help-man`

Generate `man` page. (Use `pmm-admin --help-man | man -l -` to view.)

`--debug`

Enable debug logging.

`--trace`

Enable trace logging (implies debug).

`--json`

Enable JSON output.

`--version`

Show the application version and exit.

`--server-url=server-url`

PMM Server URL in `https://username:password@pmm-server-host/` format.

`--server-insecure-tls`

Skip PMM Server TLS certificate validation.

`--group=<group-name>`

Group name for external services. Default: `external`

## COMMANDS

### GENERAL COMMANDS

`pmm-admin help [COMMAND]`

Show help for `COMMAND`.

### INFORMATION COMMANDS

`pmm-admin list --server-url=server-url [FLAGS]`

Show Services and Agents running on this Node, and the agent mode (push/pull).

`pmm-admin status --server-url=server-url [FLAGS]`

Show the following information about a local pmm-agent, and its connected server and clients:

- Agent: Agent ID, Node ID.
- PMM Server: URL and version.
- PMM Client: connection status, time drift, latency, `vagent status`, `pmm-admin version`.
- Agents: Agent ID path and client name.

FLAGS:

`--wait=<period><unit>`

Time to wait for a successful response from pmm-agent. `period` is an integer. `unit` is one of `ms` for milliseconds, `s` for seconds, `m` for minutes, `h` for hours.

```
pmm-admin summary --server-url=server-url [FLAGS]
```

Creates an archive file in the current directory with default file name `summary_<hostname>_<year>_<month>_<date>_<hour>_<minute>_<second>.zip`. The contents are two directories, `client` and `server` containing diagnostic text files.

FLAGS:

```
--filename="filename"
```

The Summary Archive filename.

```
--skip-server
```

Skip fetching `logs.zip` from PMM Server.

```
--pprof
```

Include performance profiling data in the summary.

#### CONFIGURATION COMMANDS

```
pmm-admin config
```

```
pmm-admin config [FLAGS] [node-address] [node-type] [node-name]
```

Configure a local `pmm-agent`.

FLAGS:

```
--node-id=node-id
```

Node ID (default is auto-detected).

```
--node-model=node-model
```

Node model.

```
--region=region
```

Node region.

```
--az=availability-zone
```

Node availability zone.

```
--metrics-mode=mode
```

Metrics flow mode for agents node-exporter. Allowed values: - `auto`: chosen by server (default). - `push`: agent will push metrics. - `pull`: server scrapes metrics from agent.

```
--paths-base=dir
```

Base path where all binaries, tools and collectors of PMM client are located

```
pmm-admin register
```

```
pmm-admin register [FLAGS] [node-address] [node-type] [node-name]
```

Register the current Node with the PMM Server.

```
--server-url=server-url
PMM Server URL in https://username:password@pmm-server-host/ format.

--machine-id="/machine_id/9812826a1c45454a98ba45c56cc4f5b0"
Node machine-id (default is auto-detected).

--distro="linux"
Node OS distribution (default is auto-detected).

--container-id=container-id
Container ID.

--container-name=container-name
Container name.

--node-model=node-model
Node model.

--region=region
Node region.

--az=availability-zone
Node availability zone.

--custom-labels=labels
Custom user-assigned labels.
```

**pmm-admin remove****pmm-admin remove [FLAGS] service-type [service-name]**

Remove Service from monitoring.

**--service-id=service-id**

Service ID.

**--force**

Remove service with that name or ID and all dependent services and agents.

When you remove a service, collected data remains on PMM Server for the specified **retention period**.**pmm-admin annotate****pmm-admin annotate [--node|--service] <annotation> [--tags <tags>] [--node-name=<node>] [--service-name=<service>]**Annotate an event. ([Read more](#))

**<annotation>**

The annotation string. If it contains spaces, it should be quoted.

**--node**

Annotate the current node or that specified by `--node-name`.

**--service**

Annotate all services running on the current node, or that specified by `--service-name`.

**--tags**

A quoted string that defines one or more comma-separated tags for the annotation. Example: "tag 1,tag 2".

**--node-name**

The node name being annotated.

**--service-name**

The service name being annotated.

## Combining flags

Flags may be combined as shown in the following examples.

**--node**

Current node.

**--node-name**

Node with name.

**--node --node-name=NODE\_NAME**

Node with name.

**--node --service-name**

Current node and service with name.

**--node --node-name --service-name**

Node with name and service with name.

**--node --service**

Current node and all services of current node.

**-node --node-name --service --service-name**

Service with name and node with name.

**--service**

All services of the current node.

```
--service-name
```

Service with name.

```
--service --service-name
```

Service with name.

```
--service --node-name
```

All services of current node and node with name.

```
--service-name --node-name
```

Service with name and node with name.

```
--service --service-name -node-name
```

Service with name and node with name.

 **Tip**

If node or service name is specified, they are used instead of other parameters.

## DATABASE COMMANDS

### MongoDB

```
pmm-admin add mongodb [FLAGS] [node-name] [node-address]
```

Add MongoDB to monitoring.

FLAGS:

```
--node-id=node-id
```

Node ID (default is auto-detected).

```
--pmm-agent-id=pmm-agent-id
```

The pmm-agent identifier which runs this instance (default is auto-detected).

```
--username=username
```

MongoDB username.

```
--password=password
```

MongoDB password.

```
--agent-password=password
```

Override the default password for accessing the `/metrics` endpoint. (Username is `pmm` and default password is the agent ID.)

Avoid using special characters like " , ' and '\$' in the custom password.

```
--query-source=profiler
Source of queries, one of: profiler, none (default: profiler).

--environment=environment
Environment name.

--cluster=cluster
Cluster name.

--replication-set=replication-set
Replication set name.

--custom-labels=custom-labels
Custom user-assigned labels.

--skip-connection-check
Skip connection check.

--tls
Use TLS to connect to the database.

--tls-skip-verify
Skip TLS certificates validation.

--tls-certificate-key-file=PATHTOCERT
Path to TLS certificate file.

--tls-certificate-key-file-password=IFPASSWORDTOCERTISSET
Password for TLS certificate file.

--tls-ca-file=PATHTOCACERT
Path to certificate authority file.

--metrics-mode=mode
Metrics flow mode for agents node-exporter. Allowed values: - auto: chosen by server (default). - push: agent will push metrics. - pull: server scrapes metrics from agent.
```

## Advanced Options

PMM starts the MongoDB exporter by default only with `diagnosticdata` and `replicasetstatus` collectors enabled.

```
FLAGS:
`--enable-all.Collectors`
: Enable all collectors

`--disable.Collectors`
: Comma-separated list of collector names to exclude from exporter
```

```

`--max-collections-limit=-1`
: Disable collstats, dbstats, topmetrics and indexstats if there are more than <n>
collections. 0: No limit. Default is -1, PMM automatically sets this value.

!!! caution ""
A very high limit of `max-collections-limit` could impact the CPU and Memory usage.
Check `--stats-collections` to limit the scope of collections and DB's metrics to be fetched.

`--stats-collections=db1,db2.col1`
: Collections for collstats & indexstats

```

## Enable all collectors

To enable all collectors, pass the parameter `--enable-all.Collectors` in the `pmm-admin add mongodb` command. This will enable `collstats`, `dbstats`, `indexstats`, and `topmetrics` collectors.

## Disable some collectors

To enable only some collectors, pass the parameter `--enable-all.Collectors` along with the parameter `--disable.Collectors`.

For example, if you want all collectors except `topmetrics`, specify:

```
--enable-all.Collectors --disable.Collectors=topmetrics
```

## Limit dbStats, collStats and indexStats

By default, PMM decides the limit for the number of collections to monitor the collStats and indexStats collectors.

You can also set an additional limit for the collStats, indexStats, dbStats, and topmetrics collectors with the `-max-collections-limit` parameter.

Set the value of the parameter `--max-collections-limit` to:

- 0: which indicates that collStats and indexStats can handle unlimited collections.
- n, which indicates that collStats and indexStats can handle <=n collections. If the limit is crossed - exporter stops collecting monitoring data for the collStats and indexStats collectors.
- -1 (default) doesn't need to be explicitly set. It indicates that PMM decides how many collections it would monitor, currently <=200 (subject to change).

To further limit collections to monitor, enable collStats and indexStats for some databases or collections:

- Specify the databases and collections that collStats and indexStats will use to collect data using the parameter `--stats-collections`. This parameter receives a comma-separated list of namespaces in the form `database[.collection]`.

## Examples

To add MongoDB with all collectors (`diagnosticdata`, `replicasetstatus`, `collstats`, `dbstats`, `indexstats`, and `topmetrics`) with default limit detected by PMM (currently <=200 collections, but subject to change):

```
pmm-admin add mongodb --username=admin --password=admin_pass --enable-all.Collectors mongodb_srv_1
127.0.0.1:27017
```

To add MongoDB with all collectors (`diagnosticdata`, `replicasetstatus`, `collstats`, `dbstats`, `indexstats`, and `topmetrics`) with `max-collections-limit` set to 1000:

```
pmm-admin add mongodb --username=admin --password=admin_pass --enable-all.Collectors --max-collections-
limit=1000 mongodb_srv_1 127.0.0.1:27017
```

To enable all the collectors with an unlimited number of collections monitored:

```
pmm-admin add mongodb --username=admin --password=admin_pass --enable-all.Collectors --max.collections-limit=0 mongodb_srv_1 127.0.0.1:27017
```

To add MongoDB with default collectors (`diagnosticdata` and `replicasetstatus`):

```
pmm-admin add mongodb --username=admin --password=admin_pass mongodb_srv_1 127.0.0.1:27017
```

Disable `collstats` collector and enable all the others without limiting `max.collections-limit`:

```
pmm-admin add mongodb --username=admin --password=admin_pass --enable-all.Collectors --max.collections-limit=0 --disable.Collectors=collstats mongodb_srv_1 127.0.0.1:27017
```

If `--stats.collections=db1,db2.col1` then the collectors are run as follows:

| Database | Collector is run on             |
|----------|---------------------------------|
| db1      | All the collections             |
| db2      | <b>Only</b> for collection col1 |

Enable all collectors and limit monitoring for `dbstats`, `indexstats`, `collstats` and `topmetrics` for all collections in `db1` and `col1` collection in `db2`, without limiting `max.collections-limit` for a number of collections in `db1`:

```
pmm-admin add mongodb --username=admin --password=admin_pass --enable-all.Collectors --max.collections-limit=0 --stats.collections=db1,db2.col1 mongodb_srv_1 127.0.0.1:27017
```

## Resolutions

PMM collects metrics in two resolutions to decrease CPU and Memory usage: high and low resolutions.

In high resolution we collect metrics from collectors which work fast: - `diagnosticdata` - `replicasetstatus` - `topmetrics`

In low resolution we collect metrics from collectors which could take some time: - `dbstats` - `indexstats` - `collstats`

## MySQL

```
pmm-admin add mysql [FLAGS] node-name node-address | [--name=service-name] --address=address[:port] | --socket
```

Add MySQL to monitoring.

FLAGS:

`--address`

MySQL address and port (default: 127.0.0.1:3306).

`--socket=socket`

Path to MySQL socket. (Find the socket path with `mysql -u root -p -e "select @@socket"`.)

`--node.id=node.id`

Node ID (default is auto-detected).

`--pmm.agent.id=pmm.agent.id`

The pmm-agent identifier which runs this instance (default is auto-detected).

```
--username=username
```

MySQL username.

```
--password=password
```

MySQL password.

```
--agent-password=password
```

Override the default password for accessing the `/metrics` endpoint. (Username is `pmm` and default password is the agent ID.)

Avoid using special characters like ", ; and '\$' in the custom password.

```
--query-source=slowlog
```

Source of SQL queries, one of: `slowlog`, `perfschema`, `none` (default: `slowlog`). For `slowlog` query source you need change permissions for specific files. Root permissions are needed for this.

```
--size-slow-logs=N
```

Rotate slow log file at this size. If `0`, use server-defined default. Negative values disable log rotation. A unit suffix must be appended to the number and can be one of:

- `KiB`, `MiB`, `GiB`, `TiB` for base 2 units (1024, 1048576, etc).

```
--disable-queryexamples
```

Disable collection of query examples.

```
--disable-tablestats
```

Disable table statistics collection.

Excluded collectors for low-resolution time intervals:

- `--collect.auto_increment.columns`
- `--collect.info_schema.tables`
- `--collect.info_schema.tablestats`
- `--collect.perf_schema.indexiowaits`
- `--collect.perf_schema.tableiowaits`
- `--collect.perf_schema.file_instances`

Excluded collectors for medium-resolution time intervals:

- `--collect.perf_schema.tablelocks`

```
--disable-tablestats-limit=disable-tablestats-limit
```

Table statistics collection will be disabled if there are more than specified number of tables (default: server-defined). 0=no limit. Negative value disables collection.

```
--environment=environment
```

Environment name.

`--cluster=cluster`

Cluster name.

`--replication-set=replication-set`

Replication set name.

`--custom-labels=custom-labels`

Custom user-assigned labels.

`--skip-connection-check`

Skip connection check.

`--tls`

Use TLS to connect to the database.

`--tls-skip-verify`

Skip TLS certificates validation.

`--tls-cert-file=PATHTOCERT`

Path to TLS client certificate file.

`--tls-key=PATHTOCERTKEY`

Key for TLS client certificate file.

`--tls-ca-file=PATHTOCACERT`

Path to certificate authority file.

`--ssl-ca=PATHTOCACERT`

The path name of the Certificate Authority (CA) certificate file. If used must specify the same certificate used by the server. (-ssl-capath is similar but specifies the path name of a directory of CA certificate files.)

`--ssl-cert=PATHTOCERTKEY`

The path name of the client public key certificate file.

`--ssl-key`

The path name of the client private key file.

`--ssl-skip-verify`

Skip SSL certificate verification.

`--metrics-mode=mode`

Metrics flow mode for agents node-exporter. Allowed values: - `auto` : chosen by server (default). - `push` : agent will push metrics. - `pull` : server scrapes metrics from agent.

## PostgreSQL

`pmm-admin add postgresql [FLAGS] [node-name] [node-address]`

Add PostgreSQL to monitoring.

FLAGS:

`--node-id=<node id>`

Node ID (default is auto-detected).

`--pmm-agent-id=<pmm agent id>`

The pmm-agent identifier which runs this instance (default is auto-detected).

`--username=<username>`

PostgreSQL username.

`--password=<password>`

PostgreSQL password.

`--database=<database>`

PostgreSQL database (default: `postgres`).

`--agent-password=password`

Override the default password for accessing the `/metrics` endpoint. (Username is `pmm` and default password is the agent ID.)

Avoid using special characters like " ", ';' and '\$' in the custom password.

`--query-source=<query source>`

Source of SQL queries, one of: `pgstatements`, `pgstatmonitor`, `none` (default: `pgstatements`).

`--environment=<environment>`

Environment name.

`--cluster=<cluster>`

Cluster name.

`--replication-set=<replication set>`

Replication set name.

`--custom-labels=<custom labels>`

Custom user-assigned labels.

`--skip-connection-check`

Skip connection check.

`--tls`

Use TLS to connect to the database.

`--tls-skip-verify`

Skip TLS certificates validation.

`--tls-ca-file`

TLS CA certificate file.

`--tls-cert-file`

TLS certificate file.

`--tls-key-file`

TLS certificate key file.

`--metrics-mode=mode`

Metrics flow mode for agents node-exporter. Allowed values: - `auto`: chosen by server (default). - `push`: agent will push metrics. - `pull`: server scrapes metrics from agent.

## ProxySQL

```
pmm-admin add proxysql [FLAGS] [node-name] [node-address]
```

Add ProxySQL to monitoring.

FLAGS:

`--node-id=node-id`

Node ID (default is auto-detected).

`--pmm-agent-id=pmm-agent-id`

The pmm-agent identifier which runs this instance (default is auto-detected).

`--username=username`

ProxySQL username.

`--password=password`

ProxySQL password.

`--agent-password=password`

Override the default password for accessing the `/metrics` endpoint. (Username is `pmm` and default password is the agent ID.)

Avoid using special characters like ", ; and '\$' in the custom password.

`--environment=environment`

Environment name.

```
--cluster=cluster
Cluster name.

--replication-set=replication-set
Replication set name.

--custom-labels=custom-labels
Custom user-assigned labels.

--skip-connection-check
Skip connection check.

--tls
Use TLS to connect to the database.

--tls-skip-verify
Skip TLS certificates validation.

--metrics-mode=mode
Metrics flow mode for agents node-exporter. Allowed values: - auto : chosen by server (default). - push : agent will push metrics. - pull : server scrapes metrics from agent.

--disable.Collectors
Comma-separated list of collector names to exclude from exporter.
```

**HAProxy**

```
pmm-admin add haproxy [FLAGS] [NAME]

Add HAProxy to monitoring.

FLAGS:

--server-url=SERVER-URL
PMM Server URL in https://username:password@pmm-server-host/ format.

--server-insecure-tls
Skip PMM Server TLS certificate validation.

--username=USERNAME
HAProxy username.

--password=PASSWORD
HAProxy password.

--scheme=SCHEME
Scheme to generate URI to exporter metrics endpoints (http or https).
```

**--metrics-path=METRICS-PATH**

Path under which metrics are exposed, used to generate URI (default: /metrics).

**--listen-port=LISTEN-PORT**

Listen port of haproxy exposing the metrics for scraping metrics (Required).

**--service-node-id=SERVICE-NODE-ID**

Node ID where service runs (default is auto-detected).

**--environment=ENVIRONMENT**

Environment name like 'production' or 'qa'.

**--cluster=CLUSTER**

Cluster name.

**--replication-set=REPLICATION-SET**

Replication set name.

**--custom-labels=CUSTOM-LABELS**

Custom user-assigned labels. Example: region=east,app=app1.

**--metrics-mode=MODE**

Metrics flow mode for agents node-exporter. Allowed values: - `auto`: chosen by server (default). - `push`: agent will push metrics. - `pull`: server scrapes metrics from agent.

**--skip-connection-check**

Skip connection check.

## OTHER COMMANDS

**pmm-admin add external [FLAGS]**

Add External source of data (like a custom exporter running on a port) to be monitored.

FLAGS:

**--service-name="current-hostname"**

Service name (autodetected defaults to the hostname where `pmm-admin` is running).

**--agent-node-id=AGENT-NODE-ID**

Node ID where agent runs (default is autodetected).

**--username=USERNAME**

External username.

**--password=PASSWORD**

External password.

`--scheme=http or https`

Scheme to generate URI to exporter metrics endpoints.

`--metrics-path=/metrics`

Path under which metrics are exposed, used to generate URI.

`--listen-port=LISTEN-PORT`

Listen port of external exporter for scraping metrics. (Required.)

`--service-node-id=SERVICE-NODE-ID`

Node ID where service runs (default is autodetected).

`--environment=prod`

Environment name like 'production' or 'qa'.

`--cluster=east-cluster`

Cluster name.

`--replication-set=rs1`

Replication set name.

`--custom-labels=CUSTOM-LABELS`

Custom user-assigned labels. Example: `region=east,app=app1`.

`--metrics-mode=auto`

Metrics flow mode, can be `push`: agent will push metrics, `pull`: server scrape metrics from agent or `auto`: chosen by server.

`--group="external"`

Group name of external service. (Default: `external`.)

`pmm-admin add external-serverless [FLAGS]`

Add External Service on Remote node to monitoring.

Usage example: `pmm-admin add external-serverless --url=http://1.2.3.4:9093/metrics`.

Also, individual parameters can be set instead of `--url` like: `pmm-admin add external-serverless --scheme=http --host=1.2.3.4 --listen-port=9093 --metrics-path=/metrics --container-name=ddd --external-name=e125`.

Note that some parameters are mandatory depending on the context. For example, if you specify `--url`, `--schema` and other related parameters are not mandatory. But if you specify `--host` you must provide all other parameters needed to build the destination URL, or you can specify `--address` instead of host and port as individual parameters.

FLAGS:

`--url=URL`

Full URL to exporter metrics endpoints.

```
--scheme=https  
Scheme to generate URL to exporter metrics endpoints.  
  
--username=USERNAME  
External username.  
  
--password=PASSWORD  
External password.  
  
--address=1.2.3.4:9000  
External exporter address and port.  
  
--host=1.2.3.4  
External exporters hostname or IP address.  
  
--listen-port=9999  
Listen port of external exporter for scraping metrics.  
  
--metrics-path=/metrics  
Path under which metrics are exposed, used to generate URL.  
  
--environment=testing  
Environment name.  
  
--cluster=CLUSTER  
Cluster name.  
  
--replication-set=rs1  
Replication set name.  
  
--custom-labels='app=myapp,region=s1'  
Custom user-assigned labels.  
  
--group="external"  
Group name of external service. (Default: external.)  
  
--machine-id=MACHINE-ID  
Node machine-id.  
  
--distro=DISTRO  
Node OS distribution.  
  
--container-id=CONTAINER-ID  
Container ID.
```

```
--container-name=CONTAINER-NAME
```

Container name.

```
--node-model=NODE-MODEL
```

Node model.

```
--region=REGION
```

Node region.

```
--az=AZ
```

Node availability zone.

## EXAMPLES

```
pmm-admin add mysql --query-source=slowlog --username=pmm --password=pmm sl-mysql 127.0.0.1:3306
```

```
MySQL Service added.  
Service ID : /service_id/a89191d4-7d75-44a9-b37f-a528e2c4550f  
Service name: sl-mysql
```

```
pmm-admin add mysql --username=pmm --password=pmm --service-name=ps-mysql --host=127.0.0.1 --port=3306
```

```
pmm-admin status  
pmm-admin status --wait=30s
```

```
Agent ID: /agent_id/c2a55ac6-a12f-4172-8850-4101237a4236  
Node ID : /node_id/29b2cc24-3b90-4892-8d7e-4b44258d9309  
PMM Server:  
  URL : https://x.x.x.x:443/  
  Version: 2.5.0  
PMM Client:  
  Connected : true  
  Time drift: 2.152715ms  
  Latency : 465.658µs  
  pmm-admin version: 2.5.0  
  pmm-agent version: 2.5.0  
Agents:  
  /agent_id/aeb42475-486c-4f48-a906-9546fc7859e8 mysql_slowlog_agent Running
```

## DISABLE COLLECTORS

```
pmm-admin add mysql --disable.Collectors='heartbeat,global_status,info_schema.innodb_cmp' --username=pmm --password=pmm --service-name=db1-mysql --host=127.0.0.1 --port=3306
```

For other collectors that you can disable with the `--disable.Collectors` option, please visit the official repositories for each exporter:

- [node\\_exporter](#)
  - [mysqld\\_exporter](#)
  - [mongodb\\_exporter](#)
  - [postgres\\_exporter](#)
  - [proxysql\\_exporter](#)
- 

Last update: 2022-02-08

### 5.5.3 pmm-agent - PMM Client agent

#### NAME

`pmm-agent` - The PMM Client daemon program.

#### SYNOPSIS

```
pmm-agent [command] [options]
```

#### DESCRIPTION

`pmm-agent`, part of the PMM Client package, runs as a daemon process on all monitored hosts.

#### COMMANDS

```
pmm-agent run
```

Run pmm-agent (default).

```
pmm-agent setup [node-address] [node-type] [node-name]
```

Configure local pmm-agent (requires root permissions)

```
pmm-agent help [command]
```

Show help (for command) and exit.

## OPTIONS AND ENVIRONMENT

Most options can be set via environment variables (shown in parentheses).

| Option                                    | Environment variable           | Description                                                                                                                                                                                    |
|-------------------------------------------|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| --server-<br>password=SERVER-<br>PASSWORD | PMM_AGENT_SERVER_PASSWORD      | Password to connect to PMM Server.                                                                                                                                                             |
| --server-<br>username=SERVER-<br>USERNAME | PMM_AGENT_SERVER_USERNAME      | Username to connect to PMM Server.                                                                                                                                                             |
| --server-<br>address=host:port            | PMM_AGENT_SERVER_ADDRESS       | PMM Server address and port number.                                                                                                                                                            |
| --server-insecure-tls                     | PMM_AGENT_SERVER_INSECURE_TLS  | Skip PMM Server TLS certificate validation.                                                                                                                                                    |
| --az=AZ                                   | PMM_AGENT_SETUP_AZ             | Node availability zone.                                                                                                                                                                        |
| --config-file=path_to/<br>pmm-agent.yaml  | PMM_AGENT_CONFIG_FILE          | Configuration file path and name.                                                                                                                                                              |
| --container-<br>id=CONTAINER-ID           | PMM_AGENT_SETUP_CONTAINER_ID   | Container ID.                                                                                                                                                                                  |
| --container-<br>name=CONTAINER-NAME       | PMM_AGENT_SETUP_CONTAINER_NAME | Container name.                                                                                                                                                                                |
| --debug                                   | PMM_AGENT_DEBUG                | Enable debug output.                                                                                                                                                                           |
| --distro=distro                           | PMM_AGENT_SETUP_DISTRO         | Node OS distribution (default is auto-detected).                                                                                                                                               |
| --force                                   | PMM_AGENT_SETUP_FORCE          | Remove Node with that name and all dependent Services and Agents (if existing).                                                                                                                |
| --id=/agent_id/...                        | PMM_AGENT_ID                   | ID of this pmm-agent.                                                                                                                                                                          |
| --listen-address=LISTEN-<br>ADDRESS       | PMM_AGENT_LISTEN_ADDRESS       | Agent local API address.                                                                                                                                                                       |
| --listen-port=LISTEN-<br>PORT             | PMM_AGENT_LISTEN_PORT          | Agent local API port.                                                                                                                                                                          |
| --machine-id=machine-id                   | PMM_AGENT_SETUP_MACHINE_ID     | Node machine ID (default is auto-detected).                                                                                                                                                    |
| --metrics-mode=auto                       | PMM_AGENT_SETUP_METRICS_MODE   | Metrics flow mode for agents node-exporter. Can be <code>push</code> (agent will push metrics), <code>pull</code> (server scrapes metrics from agent) or <code>auto</code> (chosen by server). |
| --node-model=NODE-MODEL                   | PMM_AGENT_SETUP_NODE_MODEL     | Node model.                                                                                                                                                                                    |
| --paths-base=PATH                         | PMM_AGENT_PATHS_BASE           | Base path for PMM client, where all binaries, tools and collectors are located. If not set, default is <code>/usr/local/percona/pmm2</code> .                                                  |

| Option                             | Environment variable              | Description                                                                                                                     |
|------------------------------------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| --paths-<br>exporters_base=PATH    | PMM_AGENT_PATHS_EXPORTERS_BASE    | Base path for exporters to use. If not set, or set to a relative path, uses value of <code>--paths-base</code> prepended to it. |
| --paths-<br>mongodb_exporter=PATH  | PMM_AGENT_PATHS_MONGODB_EXPORTER  | Path to <code>mongodb_exporter</code> .                                                                                         |
| --paths-<br>mysqld_exporter=PATH   | PMM_AGENT_PATHS_MYSQLD_EXPORTER   | Path to <code>mysqld_exporter</code> .                                                                                          |
| --paths-<br>node_exporter=PATH     | PMM_AGENT_PATHS_NODE_EXPORTER     | Path to <code>node_exporter</code> .                                                                                            |
| --paths-<br>postgres_exporter=PATH | PMM_AGENT_PATHS_POSTGRES_EXPORTER | Path to <code>postgres_exporter</code> .                                                                                        |
| --paths-<br>proxysql_exporter=PATH | PMM_AGENT_PATHS_PROXYSQL_EXPORTER | Path to <code>proxysql_exporter</code> .                                                                                        |
| --paths-pt-summary=PATH            | PMM_AGENT_PATHS_PT_SUMMARY        | Path to <code>pt-summary</code> .                                                                                               |
| --paths-pt-mysql-summary=PATH      | PMM_AGENT_PATHS_PT_MYSQL_SUMMARY  | Path to <code>pt-mysql-summary</code> .                                                                                         |
| --paths-pt-pg-summary=PATH         | PMM_AGENT_PATHS_PT_PG_SUMMARY     | Path to <code>pt-pg-summary</code> .                                                                                            |
| --paths-tempdir=PATH               | PMM_AGENT_PATHS_TEMPDIR           | Temporary directory for exporters.                                                                                              |
| --ports-max=PORTS-MAX              | PMM_AGENT_PORTS_MAX               | Highest allowed port number for listening sockets.                                                                              |
| --ports-min=PORTS-MIN              | PMM_AGENT_PORTS_MIN               | Lowest allowed port number for listening sockets.                                                                               |
| --region=REGION                    | PMM_AGENT_SETUP_REGION            | Node region.                                                                                                                    |
| --skip-registration                | PMM_AGENT_SETUP_SKIP_REGISTRATION | Skip registration on PMM Server.                                                                                                |
| --trace                            | PMM_AGENT_TRACE                   | Enable trace output (implies <code>--debug</code> ).                                                                            |
| -h, --help                         |                                   | Show help (synonym for <code>pmm-agent help</code> ).                                                                           |
| --version                          |                                   | Show application version, PMM version, time-stamp, git commit hash and branch.                                                  |

#### USAGE AND EXAMPLES OF `paths-base` FLAG

Since 2.23.0 this flag could be used for easier setup of pmm agent. With this flag the root permissions for PMM client aren't needed anymore and it will be fully working.

**Examples:**

- **Case 1:** There are no root permissions for `/usr/local/percona/pmm2` folder or there is a need to change default folder for PMM files. Command:

```
pmm-agent setup --paths-base=/home/user/custom/pmm2 --config-file=pmm-agent-dev.yaml --server-insecure-tls --server-address=127.0.0.1:443 --server-username=admin --server-password=admin
```

Config output:

```
# Updated by `pmm-agent setup`.
---
id: /agent_id/be568008-b1b4-4bd9-98c7-392d1f4b724e
listen-address: 127.0.0.1
listen-port: 7777
server:
  address: 127.0.0.1:443
  username: admin
  password: admin
  insecure-tls: true
paths:
  paths_base: /home/user/custom/pmm2
  exporters_base: /home/user/custom/pmm2/exporters
  node_exporter: /home/user/custom/pmm2/exporters/node_exporter
  mysqld_exporter: /home/user/custom/pmm2/exporters/mysqld_exporter
  mongodb_exporter: /home/user/custom/pmm2/exporters/mongodb_exporter
  postgres_exporter: /home/user/custom/pmm2/exporters/postgres_exporter
  proxysql_exporter: /home/user/custom/pmm2/exporters/proxysql_exporter
  rds_exporter: /home/user/custom/pmm2/exporters/rds_exporter
  azure_exporter: /home/user/custom/pmm2/exporters/azure_exporter
  vmagent: /home/user/custom/pmm2/exporters/vmagent
  tempdir: /tmp
  pt_summary: /home/user/custom/pmm2/tools/pt-summary
  pt_pg_summary: /home/user/custom/pmm2/tools/pt-pg-summary
  pt_mysql_summary: /home/user/custom/pmm2/tools/pt-mysql-summary
  pt_mongodb_summary: /home/user/custom/pmm2/tools/pt-mongodb-summary
ports:
  min: 42000
  max: 51999
debug: false
trace: false
```

As could be seen above, base for all exporters and tools was changed only by setting `--paths-base`. With this tag the folder for PMM that doesn't require root access could be specified.

- **Case 2:** The older `--paths-exporters_base` flag could be passed along with the `--paths-base` Command:

```
pmm-agent setup --paths-base=/home/user/custom/pmm2 --paths-exporters_base=/home/user/exporters --config-file=pmm-agent-dev.yaml --server-insecure-tls --server-address=127.0.0.1:443 --server-username=admin --server-password=admin
```

Config output:

```
# Updated by `pmm-agent setup`.
---
id: /agent_id/afce1917-8836-4857-b3e5-ad372c2ddbe5
listen-address: 127.0.0.1
listen-port: 7777
server:
  address: 127.0.0.1:443
  username: admin
  password: admin
```

```

    insecure-tls: true
paths:
  paths_base: /home/user/custom/pmm2
  exporters_base: /home/user/exporters
  node_exporter: /home/user/exporters/node_exporter
  mysqld_exporter: /home/user/exporters/mysqld_exporter
  mongodb_exporter: /home/user/exporters/mongodb_exporter
  postgres_exporter: /home/user/exporters/postgres_exporter
  proxysql_exporter: /home/user/exporters/proxysql_exporter
  rds_exporter: /home/user/exporters/rds_exporter
  azure_exporter: /home/user/exporters/azure_exporter
  vagent: /home/user/exporters/vagent
  tempdir: /tmp
  pt_summary: /home/user/custom/pmm2/tools/pt-summary
  pt_pg_summary: /home/user/custom/pmm2/tools/pt-pg-summary
  pt_mysql_summary: /home/user/custom/pmm2/tools/pt-mysql-summary
  pt_mongodb_summary: /home/user/custom/pmm2/tools/pt-mongodb-summary
ports:
  min: 42000
  max: 51999
debug: false
trace: false

```

As could be seen above the behavior for the `--paths-base` was the same, but paths for all exporters were overwritten by the `--paths-exporter_base` flag.

**Summary:** Flag `--paths-base` will set path for all exporters and tools, but each one could be overridden by specific flag (like `--paths-mongodb_exporter`, `--paths-pt-mysql-summary` and etc).

## LOGGING

By default, pmm-agent sends messages to stderr and to the system log (`syslogd` or `journald` on Linux).

To get a separate log file, edit the `pmm-agent` start-up script.

### systemd-based systems

- Script file: `/usr/lib/systemd/system/pmm-agent.service`
- Parameter: `StandardError`
- Default value: `file:/var/log/pmm-agent.log`

Example:

```
StandardError=file:/var/log/pmm-agent.log
```

### initd-based systems

- Script file: `/etc/init.d/pmm-agent`
- Parameter: `pmm_log`
- Default value: `/var/log/pmm-agent.log`

Example:

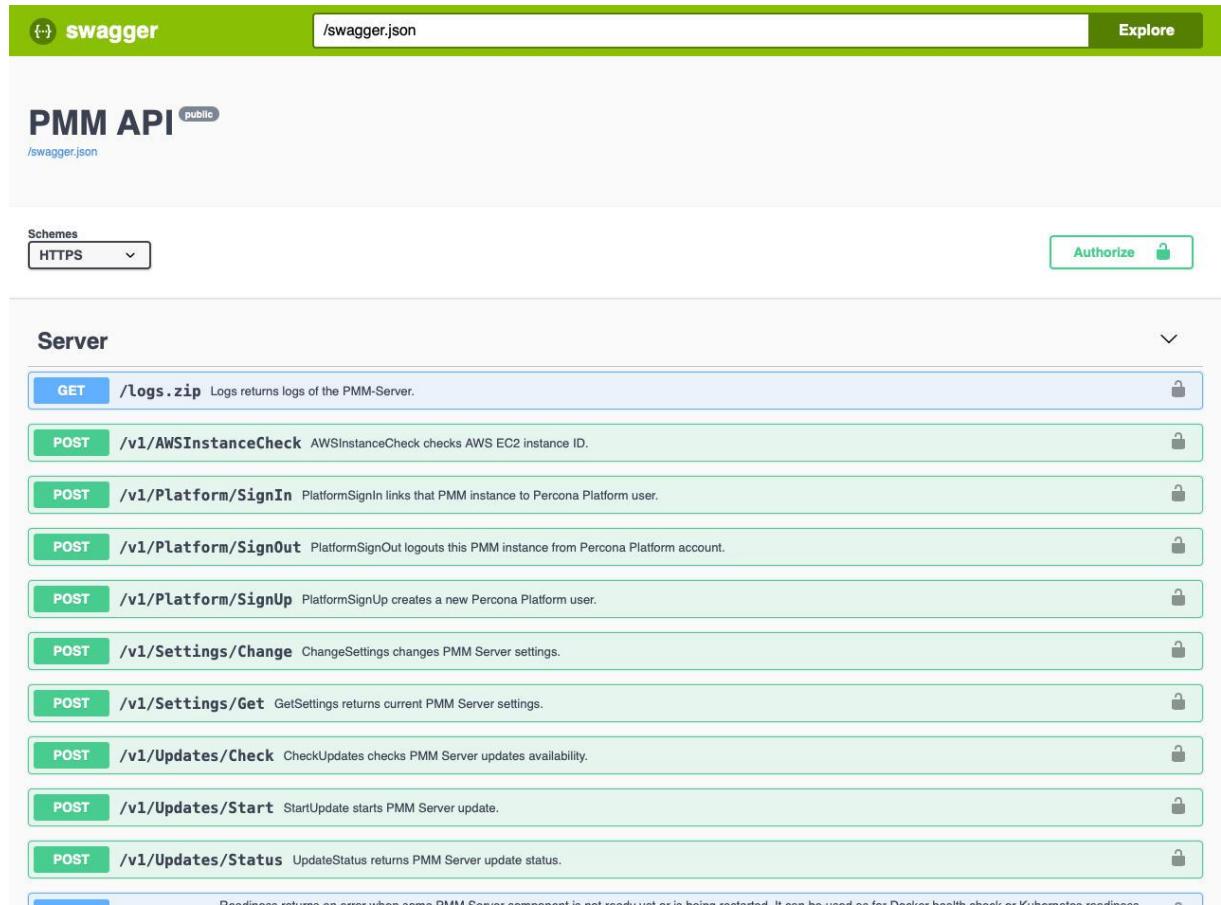
```
pmm_log="/var/log/pmm-agent.log"
```

If you change the default log file name, reflect the change in the log rotation rules file `/etc/logrotate.d/pmm-agent-logrotate`.

Last update: 2021-10-21

## 5.6 API

PMM Server lets you visually interact with API resources representing all objects within PMM. You can browse the API using the [Swagger UI](#), accessible at the `/swagger/` endpoint URL:



The screenshot shows the PMM API Swagger UI interface. At the top, there's a navigation bar with a 'swagger' icon, the URL '/swagger.json', and a 'Explore' button. Below the header, the title 'PMM API' is displayed with a 'public' badge. A 'Schemes' dropdown is set to 'HTTPS'. On the right, there's an 'Authorize' button with a lock icon.

The main content area is titled 'Server' and lists several API endpoints:

- GET /logs.zip** Logs returns logs of the PMM-Server.
- POST /v1/AWSInstanceCheck** AWSInstanceCheck checks AWS EC2 instance ID.
- POST /v1/Platform/SignIn** PlatformSignIn links that PMM instance to Percona Platform user.
- POST /v1/Platform/SignOut** PlatformSignOut logsouts this PMM instance from Percona Platform account.
- POST /v1/Platform/SignUp** PlatformSignUp creates a new Percona Platform user.
- POST /v1/Settings/Change** ChangeSettings changes PMM Server settings.
- POST /v1/Settings/Get** GetSettings returns current PMM Server settings.
- POST /v1/Updates/Check** CheckUpdates checks PMM Server updates availability.
- POST /v1/Updates/Start** StartUpdate starts PMM Server update.
- POST /v1/Updates>Status** UpdateStatus returns PMM Server update status.

A note at the bottom states: 'Readiness returns an error when some PMM Server component is not ready yet or is being contacted. It can be used as for Docker health checks or Kubernetes readiness'.

Clicking an object lets you examine objects and execute requests on them:

**Server**

**GET** `/logs.zip` Logs returns logs of the PMM-Server.

**Parameters**

No parameters

**Responses**

Response content type `application/zip`

| Code    | Description            |
|---------|------------------------|
| 200     | A successful response. |
| default | An error response.     |

Example Value | Model

```
{
  "code": 0,
  "error": "string",
  "message": "string"
}
```

**POST** `/v1/AWSInstanceCheck` AWSInstanceCheck checks AWS EC2 instance ID.

**POST** `/v1/Platform/SignIn` PlatformSignIn links that PMM instance to Percona Platform user.

**POST** `/v1/Platform/SignOut` PlatformSignOut logsouts this PMM instance from Percona Platform account.

**POST** `/v1/Platform/SignUp` PlatformSignUp creates a new Percona Platform user.

**POST** `/v1/Settings/Change` ChangeSettings changes PMM Server settings.

**POST** `/v1/Settings/Get` GetSettings returns current PMM Server settings.

The objects visible are nodes, services, and agents:

- A **Node** represents a bare metal server, a virtual machine, a Docker container, or a more specific type such as an Amazon RDS Node. A node runs zero or more Services and Agents, and has zero or more Agents providing insights for it.
- A **Service** represents something useful running on the Node: Amazon Aurora MySQL, MySQL, MongoDB, etc. It runs on zero (Amazon Aurora Serverless), single (MySQL), or several (Percona XtraDB Cluster) Nodes. It also has zero or more Agents providing insights for it.
- An **Agent** represents something that runs on the Node which is not useful in itself but instead provides insights (metrics, query performance data, etc) about Nodes and/or Services. An agent always runs on the single Node (except External Exporters), and provides insights for zero or more Services and Nodes.

Nodes, Services, and Agents have **Types** which define specific their properties, and their specific logic.

Nodes and Services are external by nature – we do not manage them (create, destroy), but merely maintain a list of them (add to inventory, remove from inventory) in `pmm-managed`. Most Agents are started and stopped by `pmm-agent`. One exception is the External Exporter Type which is started externally.

### 5.6.1 API Keys and authentication

API keys are used to control access to the PMM server components and resources. With an API key, you are authenticated to the PMM server, have access to PMM server components and resources, and perform various actions on them. You can use API keys as a replacement for basic authentication.

#### Generate API keys

PMM uses the Grafana API keys for authentication. Following are the steps to generate the API keys:

1. Login to PMM.
2. From the side menu, click *Configuration* → *API keys*.
3. On the Configuration page, click *Add API Key*.
4. *Add API key* dialog box opens.
5. Enter the following to generate an API key:
  - key name (you can give any desired name)
  - Select the Role from the dropdown
  - Enter a value in the Time to live text box (hover on the tooltip for more information).
  - Click Add.
6. *API Key Created* window displays your newly created key. Make sure to copy your key and keep it secure.

#### Authenticate

You can authenticate your request using the HTTPS header.

 **Important**

Use the `-k` or `--insecure` parameter to force cURL to ignore invalid and self-signed SSL certificate errors. The option will skip the SSL verification process, and you can bypass any SSL errors while still having SSL-encrypted communication. However, using the `--insecure` parameter is not recommended. Although the data transfer is encrypted, it is not entirely secure. For enhanced security of your PMM installation, you need valid SSL certificates. For information on validating SSL certificates, refer to: [SSL certificates](#).

```
curl -H "Authorization: Bearer <api_key>" https://127.0.0.1/v1/version
```

#### Use an API key in basic auth

You can pass the API key into a REST API call as a query parameter in the following format. Replace `API_KEY` with your API key.

#### Example

```
curl -X GET https://api_key:API_KEY@localhost/v1/version
```

---

Last update: 2021-12-30

## 5.7 VictoriaMetrics

[VictoriaMetrics](#) is a third-party monitoring solution and time-series database that replaced Prometheus in [PMM 2.12.0](#).

### 5.7.1 Push/Pull modes

VictoriaMetrics metrics data can be both ‘pushed’ to the server and ‘pulled’ by the server. When setting up services, you can decide which mode to use.

The ‘push’ mode is now default for newly-added services. (In PMM 2.12.0 the default mode was ‘pull’.)

The mode (push/pull) is controlled by the `--metrics-mode` flag for the `pmm-admin config` and `pmm-admin add` commands.

If you need to change the metrics mode for an existing Service, you must remove it and re-add it with the same name and the required flags. (You cannot update a service.)

### 5.7.2 Remapped targets for direct Prometheus paths

Direct Prometheus paths return structured information directly from Prometheus, bypassing the PMM application.

They are accessed by requesting a URL of the form `<PMM SERVER URL>/prometheus/<PATH>`.

As a result of the move to VictoriaMetrics some direct Prometheus paths are no longer available.

| Prometheus path                            | VictoriaMetrics equivalent                                                                                                                   |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <code>/prometheus/alerts</code>            | No change.                                                                                                                                   |
| <code>/prometheus/config</code>            | No equivalent, but there is some information at <code>/prometheus/targets</code> .                                                           |
| <code>/prometheus	flags</code>             | The <code>flag</code> metrics at <code>/prometheus/metrics</code> .                                                                          |
| <code>/prometheus/graph</code>             | <code>/graph/explore</code> (Grafana) or <code>graph/d/prometheus-advanced/advanced-data-exploration</code> (PMM dashboard).                 |
| <code>/prometheus/rules</code>             | No change.                                                                                                                                   |
| <code>/prometheus/service-discovery</code> | No equivalent.                                                                                                                               |
| <code>/prometheus/status</code>            | Some information at <code>/prometheus/metrics</code> . High cardinality metrics information at <code>/prometheus/api/v1/status/tsdb</code> . |
| <code>/prometheus/targets</code>           | <code>/victoriameetrics/targets</code>                                                                                                       |

### 5.7.3 Troubleshooting

To troubleshoot issues, see the VictoriaMetrics [troubleshooting documentation](#).

You can also contact the VictoriaMetrics team via:

- [Google Groups](#)
  - [Slack](#)
  - [Reddit](#)
  - [Telegram](#)
- 

Last update: 2021-07-29

## 5.8 Glossary

### 5.8.1 Annotation

A way of showing a mark on dashboards signifying an important point in time.

### 5.8.2 Dimension

In the Query Analytics dashboard, to help focus on the possible source of performance issues, you can group queries by *dimension*, one of: Query, Service Name, Database, Schema, User Name, Client Host

### 5.8.3 EBS

Amazon's Elastic Block Store.

### 5.8.4 Fingerprint

A *normalized statement digest*—a query string with values removed that acts as a template or typical example for a query.

### 5.8.5 IAM

Identity and Access Management (for Amazon AWS).

### 5.8.6 MM

Metrics Monitor.

### 5.8.7 NUMA

Non-Uniform Memory Access.

### 5.8.8 PEM

Privacy Enhanced Mail.

### 5.8.9 QPS

Queries Per Second. A measure of the rate of queries being monitored.

### 5.8.10 Query Analytics

Component of PMM Server that enables you to analyze MySQL query performance over periods of time.

### 5.8.11 STT

Security Threat Tool.

### 5.8.12 Technical Preview

Releases intended for public preview and feedback but with no support or service level agreement (SLA). Should not be used on production or business-critical systems. May contain breaking changes to UI, API, CLI. ([Read more.](#))

### 5.8.13 VG

Volume Group.

---

Last update: 2021-06-29

## 6. FAQ

### 6.1 How can I contact the developers?

- [Community forum.](#)
- [Discord chat.](#)
- [PMM project in JIRA.](#)

### 6.2 What are the minimum system requirements?

- Server:
  - Disk: 1 GB per monitored database (1 week data retention)
  - Memory: 2 GB per monitored database
  - CPU: Supports [SSE4.2](#)
- Client:
  - Disk: 100 MB



#### See also

- [Setting up PMM Server](#)
- [Setting up PMM Client](#)

### 6.3 How can I upgrade from version 1?

There is no direct software upgrade path.

You must [set up](#) PMM 2 and connect your existing clients to it.

When all data is registered in PMM2 and expired in PMM1, decommission your PMM1 instance.



#### See also

- [Upgrade from PMM1](#)
- [Percona blog: Running PMM1 and PMM2 Clients on the Same Host](#)

### 6.4 How to control data retention?

Go to [Configuration](#) → [Settings](#) → [Advanced Settings](#) → [Data retention](#) to adjust the value in days.



#### See also

[Configure data retention](#)

## 6.5 How are PMM Server logs rotated?

PMM Server embeds multiple components, like Victoria Metrics, Query Analytics, Grafana, `managed`, PostgreSQL, ClickHouse, etc. (components). All PMM Server component logs are rotated by `supervisord`. The components' log rotation settings are stored in `*.ini` files within the `/etc/supervisord.d` directory. Those settings define both the maximum size of a log file and the number of log files to keep. The log rotation takes place once the log file reaches its maximum size.

## 6.6 What privileges are required to monitor a MySQL instance?

```
SELECT, PROCESS, SUPER, REPLICATION CLIENT, RELOAD
```

 See also

[Setting Up/Client/MySQL](#).

## 6.7 Can I monitor multiple service instances?

Yes.

You can add multiple instances of MySQL or any other service to be monitored from the same PMM Client.

To do this, you provide a unique port and IP address, or a socket for each instance, and specify a unique name for each. (If a name is not provided, PMM uses the name of the PMM Client host.)

For example, to add MySQL monitoring for two local MySQL servers:

```
pmm-admin add mysql --username root --password root instance-01 127.0.0.1:3001
pmm-admin add mysql --username root --password root instance-02 127.0.0.1:3002
```

 See also

[pmm-admin add mysql](#)

## 6.8 Can I rename instances?

Yes, by removing and re-adding with a different name.

When you remove a monitoring service, previously collected data remains available in Grafana. However, the metrics are tied to the instance name. So if you add the same instance back with a different name, it will be considered a new instance with a new set of metrics. So if you are re-adding an instance and want to keep its previous data, add it with the same name.

## 6.9 Can I add an AWS RDS MySQL or Aurora MySQL instance from a non-default AWS partition?

By default, the RDS discovery works with the default `aws` partition. But you can switch to special regions, like the `GovCloud` one, with the alternative `AWS partitions` (e.g. `aws-us-gov`) adding them to the `Settings` via the PMM Server API.

**POST** /v1/Settings/Change ChangeSettings changes PMM Server settings.

| Parameters                                                                                                                                                                                                                                                                                                                                     | Try it out |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>Name</b> <b>Description</b><br><b>body</b> * required<br><code>(body)</code> <pre>{   "enable_telemetry": true,   "disable_telemetry": true,   "metrics_resolutions": {     "hr": "string",     "mr": "string",     "lr": "string"   },   "data_retention": "string",   "ssh_key": "string",   "aws_partitions": [     "string"   ] }</pre> |            |
| <b>Parameter content type</b><br><input type="button" value="application/json"/>                                                                                                                                                                                                                                                               |            |

To specify other than the default value, or to use several, use the JSON Array syntax: `["aws", "aws-cn"]`.

## 6.10 What resolution is used for metrics?

The default values (in seconds):

| Preset            | Low | Medium | High |
|-------------------|-----|--------|------|
| Rare              | 300 | 180    | 60   |
| Standard          | 60  | 10     | 5    |
| Frequent          | 30  | 5      | 1    |
| Custom (defaults) | 60  | 10     | 5    |

### See also

[Metrics resolution](#)

## 6.11 How do I set up Alerting?

When a monitored service metric reaches a defined threshold, PMM Server can trigger alerts for it either using the Grafana Alerting feature or by using an external alert manager.

With these methods you must configure alerting rules that define conditions under which an alert should be triggered, and the channel used to send the alert (e.g. email).

Alerting in Grafana allows attaching rules to your dashboard panels. Grafana Alerts are already integrated into PMM Server and may be simpler to get set up.

Alertmanager allows the creation of more sophisticated alerting rules and can be easier to manage installations with a large number of hosts. This additional flexibility comes at the expense of simplicity.

We only offer support for creating custom rules to our customers, so you should already have a working Alertmanager instance prior to using this feature.



#### See also

[PMM Alerting with Grafana: Working with Templated Dashboards](#)

## 6.12 How do I use a custom Prometheus configuration file?

Normally, PMM Server fully manages the [Prometheus configuration file](#).

However, some users may want to change the generated configuration to add additional scrape jobs, configure remote storage, etc.

From version 2.4.0, when `pmm-managed` starts the Prometheus file generation process, it tries to load the `/srv/prometheus/prometheus.base.yml` file first, to use it as a base for the `prometheus.yml` file.

The `prometheus.yml` file can be regenerated by restarting the PMM Server container, or by using the `SetSettings` API call with an empty body.



#### See also

- [API](#)
- [Percona blog: Extending PMM's Prometheus Configuration](#)

## 6.13 How to troubleshoot an Update?

See [Troubleshoot update](#).

## 6.14 What are my login credentials when I try to connect to a Prometheus Exporter?

- User name: `pmm`
- Password: Agent ID

PMM protects an exporter's output from unauthorized access by adding an authorization layer. To access an exporter you can use `pmm` as a user name and the Agent ID as a password. You can find the Agent ID corresponding to a given exporter by running `pmm-admin list`.



#### See also

[pmm-admin list](#)

## 6.15 How to provision PMM Server with non-default admin password?

Currently there is no API available to change the `admin` password. If you're deploying through Docker you can use the following code snippet to change the password after starting the Docker container:

```
PMM_PASSWORD="mypassword"
echo "Waiting for PMM to initialize to set password..."
until [ "`docker inspect -f {{.State.Health.Status}} pmm-server`" = "healthy" ]; do sleep 1;
done
docker exec -t pmm-server bash -c "grafana-cli --homepath /usr/share/grafana admin reset-
admin-password $PMM_PASSWORD"
```

(This example assumes your Docker container is named `pmm-server`.)

---

Last update: 2021-08-27

## 7. Release Notes

### 7.1 Release Notes

- [Percona Monitoring and Management 2.26.0](#)
  - [Percona Monitoring and Management 2.25.0](#)
  - [Percona Monitoring and Management 2.24.0](#)
  - [Percona Monitoring and Management 2.23.0](#)
  - [Percona Monitoring and Management 2.22.0](#)
  - [Percona Monitoring and Management 2.21.0](#)
  - [Percona Monitoring and Management 2.20.0](#)
  - [Percona Monitoring and Management 2.19.0](#)
  - [Percona Monitoring and Management 2.18.0](#)
  - [Percona Monitoring and Management 2.17.0](#)
  - [Percona Monitoring and Management 2.16.0](#)
  - [Percona Monitoring and Management 2.15.1](#)
  - [Percona Monitoring and Management 2.15.0](#)
  - [Percona Monitoring and Management 2.14.0](#)
  - [Percona Monitoring and Management 2.13.0](#)
  - [Percona Monitoring and Management 2.12.0](#)
  - [Percona Monitoring and Management 2.11.1](#)
  - [Percona Monitoring and Management 2.11.0](#)
  - [Percona Monitoring and Management 2.10.1](#)
  - [Percona Monitoring and Management 2.10.0](#)
  - [Percona Monitoring and Management 2.9.1](#)
  - [Percona Monitoring and Management 2.9.0](#)
  - [Percona Monitoring and Management 2.8.0](#)
  - [Percona Monitoring and Management 2.7.0](#)
  - [Percona Monitoring and Management 2.6.1](#)
  - [Percona Monitoring and Management 2.6.0](#)
  - [Percona Monitoring and Management 2.5.0](#)
  - [Percona Monitoring and Management 2.4.0](#)
  - [Percona Monitoring and Management 2.3.0](#)
  - [Percona Monitoring and Management 2.2.2](#)
  - [Percona Monitoring and Management 2.2.1](#)
  - [Percona Monitoring and Management 2.2.0](#)
  - [Percona Monitoring and Management 2.1.0](#)
  - [Percona Monitoring and Management 2.0.1](#)
  - [Percona Monitoring and Management 2.0.0](#)
-

Last update: 2022-02-08

## 7.2 Percona Monitoring and Management 2.26.0

Date: February 8, 2022

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.2.1 Release Highlights

- **Experimental Dashboards**

New experimental dashboards are introduced that will be shipped to the PMM users. These dashboards will be uploaded to the **Experimental** folder to enable the users to test them in their environment and provide feedback.

The following Dashboards are being shipped as part of this release:

- K8s monitoring dashboard
- Environment Overview dashboard
- Environment Summary dashboard

 **Important**

These experimental dashboards are subject to change. It is recommended to use these dashboards for testing purposes only.

- **SMTP Configuration Verification**

When configuring an SMTP email server for sending out email notifications, you now have the option to test that your specified settings are correct. The **Email\* tab under Configuration > Settings > Communication** **now includes a Test\*\* button** to send a test alert through the specified server. For more information about setting up an SMTP email server, see [Configure](#).

- **Breaking change for Integrated Alerting (Technical Preview)**

This release introduces major changes to the core structure of rule templates. As a result, alert rules and templates created in previous PMM versions are not compatible with PMM 2.26.0 and cannot be migrated to this new version. After upgrading, make sure to manually recreate any custom alert rules and rule templates that you want to transfer to PMM 2.26.0.

 **Disclaimer**

Integrated Alerting is still a Preview functionality and, as such, subject to change. We recommend that early adopters use this feature for testing purposes only.

### 7.2.2 New Features

- **PMM-9059:** Following the recently introduced support for connecting PMM to Percona Platform, you now have the option to also unlink any servers that are no longer relevant to your Platform organization. To disconnect a PMM server, go to **Configuration > Settings > Percona Platform** and click **Disconnect**. You can check the list of servers connected to an organization in Percona Platform by clicking **View instances** on the Dashboard page. For more information, see [Configure](#) for more details.

- [PMM-9312](#): Tech Preview Feature: PMM now captures the MongoDB metrics such as dbStats, collStats, indexStats, and topmetrics. See [Documentation](#) for more details.

### 7.2.3 Improvements

- [PMM-9176](#): DBaaS - PMM now supports Percona Distribution for MongoDB Operator 1.10.0.
- [PMM-9159](#): DBaaS - PMM now supports Percona Kubernetes Operator for Percona XtraDB Cluster 1.10.0.
- [PMM-9180](#): Integrated Alerting > Add Alert Rule - Added the Template expression in a collapsible panel for an enhanced user experience (default view as collapsed) as the technical message could confuse the users.
- [PMM-7781](#): Integrated Alerting - Alert rules no longer depend on their source rule template after creation. This means that you can now update or delete rule templates without impacting existing rules that are based on that template. For more information, see [Integrated Alerting](#).
- [PMM-9356](#): Added new experimental Environment dashboards in PMM.
- [PMM-9296](#): Disclaimer about Technical Preview feature added to Percona Platform - Connect PMM to Percona portal page.

### 7.2.4 Bugs Fixed

- [PMM-9416](#): Upgrading to PMM 2.25.0 using docker (replacing the image) fails when upgrading from versions less than or equal to 2.23.0.

 **Caution**

It is recommended to upgrade directly to PMM 2.26.0 instead of 2.25.0 when updating from versions less than or equal to PMM 2.23.0 if your PMM doesn't have external access (access to `repo.percona.com`).

- [PMM-8867](#): Fixed an issue for PMM Client installation using the tarball script (without using RPMs) where the configuration was getting lost due to the configuration file `pmm-agent.yml` being recreated.
- [PMM-8094](#): DBaaS - Fixed an issue for paused clusters that froze with PSMDB v1.8 operators when all the pods were terminated, providing no cluster resumption option.
- [PMM-8535](#): DBaaS - Repeating error after force unregister
- [PMM-9144](#): Dashboard - Fixed the Add inventory page issue that indicated AWS RDS/Aurora supported only MySQL.
- [PMM-9289](#): **Get from Browser** on settings page does not fetch port in Public address field breaking the integration for Platform authentication.
- [PMM-9255](#): On connecting the PMM server to Percona Platform for an admin user insufficient access rights error message is thrown.
- [PMM-9049](#): Eliminated confusion around the current and available version date by adding a tooltip with an explanation for these dates.
- [PMM-9181](#): Integrated Alerting - Modified the label for the enable/disable button in order to avoid confusion.
- [PMM-5405](#): Fixed an issue where the `pmm-admin summary` command fails if a null value is passed for the `--filename` parameter.
- [PMM-8141](#): Fixed an issue where the metrics were not captured as the cleanup of the temporary folder on the client node deleted the requisite configuration file.

## 7.3 Percona Monitoring and Management 2.25.0

Date: December 14, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### **Important note for users considering docker way upgrade to PMM 2.25.0**

If you upgrade from a PMM version less than or equal to 2.23.0 using docker, it will fail if your PMM does not have external access (access to 'repo.percona.com'). Thus, it is recommended to upgrade to PMM 2.26.0 instead.

See [PMM-9416](#) for more details.

### 7.3.1 Release Highlights

- **Percona Platform (Technology Preview):**

- Connect Percona Monitoring and Management (PMM) to [Percona Platform](#) to boost the monitoring capabilities of your PMM installations and access all your Percona accounts and services from one single, centralized location. For more information, see the [Percona Platform Portal documentation](#)

- **Enhanced PostgreSQL monitoring**

- You can now [specify custom database names](#) when adding PostgreSQL Servers for monitoring. Previous PMM versions always used the default `postgres` name instead.
- Added support for the new version of `pg_stat_monitor` extension. [Release Candidate v1.0.0-rc.1](#) brings many new PostgreSQL metrics, Dashboards and Query Analytics! To find out about all the features available in the new `pg_stat_monitor` version, see the [pg\\_stat\\_monitor User guide](#)
- Added compatibility for the latest Percona Distributions for PostgreSQL 14, 13, 12, 11 updated on December 7, 2021, which includes the newest version of the `pg_stat_monitor` extension.

- **Grafana usability enhancements**

- PMM is now using the native Grafana provisioning mechanism for adding dashboards, plug-ins, and data sources. This ensures faster and more reliable upgrading to newer PMM versions. Your existing plug-ins and dashboard changes are preserved during upgrades, but always make sure to back them up before upgrading and check that everything transferred correctly afterward.
- Added option to change the time zone on dashboards. This selection is preserved while you navigate over Dashboards. If you need to change this setting permanently for your account, change it in your preferences by the URL: [https://YOUR\\_PMM\\_SERVER/graph/profile](https://YOUR_PMM_SERVER/graph/profile)

- **DBaaS (Technical Preview)**

- You can now deploy and update your DBaaS created PXC clusters to the latest version of Percona Distribution for MySQL Operator 1.9.0. This enables you to take advantage of the latest features and fixes.

### **. PMM environment enhancements**

- The pmm-client docker container can now be started as a sidecar. For users that use PMM client in Kubernetes or build automation around it, you can now start the client as a sidecar container simply by passing a flag. The client will also gracefully handle any instances where the connection to DB is not available. For more details, see the [PMM Client documentation](#)
- Removed support for Ubuntu 16.04. With the support of new products and new versions of already supported products, we also removed old, unsupported software. As of this release, we are no longer supporting Ubuntu 16.04 in PMM [according to recent announcements](#)

### **7.3.2 New Features**

- [PMM-9050](#): Connect PMM Server to Percona Platform for additional account info in PMM and value added content

### **7.3.3 Improvements**

- [PMM-8545](#): DBaaS: Support of Percona Distribution for MySQL Operator 1.9.0 in PMM
- [PMM-7677](#): Docker container for `pmm-client`: Option to change behavior and follow sidecar pattern The flags `PMM_AGENT_SIDECAR` and `PMM_AGENT_SIDECAR_SLEEP` does this. Read more in [documentation](#)
- [PMM-3516](#): Optimize provisioning mechanism of plugins, dashboards, and datasources in PMM
- [PMM-8674](#): Integrated Alerting: Add Tooltips to `Add Alert Rule` fields to make it easier to understand what information is needed
- [PMM-8505](#): Integrated Alerting: Clarify description of the 'Low memory' Alert Template
- [PMM-8503](#): Integrated Alerting: Field validation in Email and Slack tabs when updating settings
- [PMM-7527](#): Integrated Alerting: Improvements to overall user experience for action buttons in Alerting
- [PMM-7079](#): Integrated Alerting: New 'information' icon to give additional details about Alerts without cluttering screen
- [PMM-8259](#): Better clarification of error messages in `pmm-admin` when PMM server can't be unregistered
- [PMM-8972](#): Add ability to specify custom base path to exporters and tools using `pmm-admin` command
- [PMM-8282](#): Improved messaging for TLS option when adding Remote instances in PMM over UI

### **7.3.4 Bugs Fixed**

- [PMM-9169](#): Security Advisor Checks are not working for MongoDB instances
- [PMM-8982](#): Backup Management: User is not able to see MongoDB backup logs if backup was taken on older version of pmm-server
- [PMM-9157](#): Dashboards: Changing the timezone on dashboards does not persist navigation
- [PMM-7116](#): Dashboards: Incorrect STARTUP state on MongoDB ReplSet Summary dashboard
- [PMM-8993](#): Integrated Alerting: Sending email using Gmail fails
- [PMM-7802](#): PMM can't monitor MongoDB arbiter nodes (Thanks to Artem Meshcheryakov for reporting this issue)
- [PMM-6937](#): Can't add PostgreSQL instances to PMM without `postgres` DB in PostgreSQL server (Thanks to Daniel Kowalewski for reporting this issue)
- [PMM-7447](#): Can't add into PMM instances of PostgreSQL with SCRAM-SHA-256 authentication
- [PMM-9085](#): PMM Server crashes after upgrading to 2.22 every 4 hours
- [PMM-9156](#): `pmm-agent` paths-base option not working for pmm2-client binary installation in PMM 2.23.0
- [PMM-8461](#): DBaaS: Confusing error when accessing DBaaS pages when it's disabled

- [PMM-8110](#): DBaaS: Registering K8s cluster with operators already installed can cause error
- [PMM-8694](#): Query Analytics: URLs in Query Analytics with a selected query and a timestamp range does not select the query
- [PMM-9227](#): Pagination Reset on QAN after Time Range change doesn't work, results in wrong results
- [PMM-9298](#): PMM AMI image in 2.24.0 has only 8GB space for data and Volume Size Check fails while upgrading to 2.25.0

#### Important note for users of PMM 2.24.0

2.24.0 AMI image has only 8GB available for the data, it is a bug (see [PMM-9298](#)). To resize a disk to full size you need to login to AMI instance with SSH and use the following command:

```
curl https://raw.githubusercontent.com/percona/pmm-update/main/ansible/playbook/tasks/create-lvm.yml -o lvn-fix.yml && sudo ansible-playbook lvn-fix.yml
```

For instructions about how to access your instances by using an SSH client, see [Connecting to Your Linux Instance Using SSH](#). Make sure to replace the user name ec2-user used in this document with admin.

What this command does:

1. Downloads Ansible playbook and runs it
2. Copy your data from /srv to the temporary directory
3. Create lvm partition
4. Copy data from system disk to a new LVM partition

### 7.3.5 Known issues

- [PMM-9255](#): After connecting PMM to Percona Platform, PMM occasionally shows a false permission issue notification, incorrectly suggesting that the connection could not be established due to missing permissions. Reload the page to remove the incorrect notification and confirm the connection.
- [PMM-9312](#): It's not possible to enable collStats, indexStats and -max-collections-limit for MongoDB

---

Last update: 2022-02-08

## 7.4 Percona Monitoring and Management 2.24.0

Date: November 18, 2021

Installation: [Installing Percona Monitoring and Management](#)

### **⚠ Important note for users of PMM 2.24.0**

2.24.0 AMI image has only 8GB available for the data, it is a bug (see [PMM-9298](#)). To resize a disk to full size you need to login to AMI instance with SSH and use the following command:

```
curl https://raw.githubusercontent.com/percona/pmm-update/main/ansible/playbook/tasks/create-lvm.yml -o lvn-fix.yml && sudo ansible-playbook lvn-fix.yml
```

For instructions about how to access your instances by using an SSH client, see [Connecting to Your Linux Instance Using SSH](#) Make sure to replace the user name ec2-user used in this document with admin.

What this command does:

1. Downloads Ansible playbook and runs it
2. Copy your data from /srv to the temporary directory
3. Create lvm partition
4. Copy data from system disk to a new LVM partition

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.4.1 Release Highlights

- **Enhanced PostgreSQL monitoring**

- Beginning with this release, PMM now supports monitoring of PostgreSQL 14, both - Community edition and [Percona Distribution for PostgreSQL](#).
- We've made significant improvements in exposed data and added additional features if you monitor PostgreSQL together with the [pg\\_stat\\_monitor](#) extension (Part of Percona distribution for PostgreSQL). These features include:
  - The PostgreSQL queries will have complete Query Execution Plan information. This will help with future optimization of queries and give a clear understanding of query performance
  - Query execution histograms collection and presentation inside Query Analytics for a much more detailed understanding of query executions.
  - Query analytics will both show and let the user drill down to the Top Query of the particular query if it's a subquery and have this parent query. This feature will allow users to see the dependencies between queries better and understand the impact of subqueries.
  - Query Analytics can filter PostgreSQL queries by query commands like SELECT, UPDATE, etc., and by Application Name if it's set for PostgreSQL connection from the application.

- **Integrated Alerting (Technical preview):**

- Alerting in PMM now has an additional notification channel type - webhooks. So now, users can integrate Alerting with any tool they use for Incident management. Read more about new notification channels and how to set them up in [our documentation](#)

## 7.4.2 New Features

- [PMM-8027](#): Integrated Alerting: New notification channel added: Webhooks
- [PMM-8301](#): Add data collection support and visualization for newly added dimensions in pg\_stat\_monitor such as Application Name, Top Query, Plan in Query Analytics
- [PMM-8588](#): PostgreSQL Histograms added to QAN when using pg\_stat\_monitor extension
- [PMM-8632](#): New Filter: “Command Type” allows filtering queries based on type (SELECT, INSERT, UPDATE, DELETE, n/a) when pg\_stat\_monitor extension enabled

## 7.4.3 Improvements

- [PMM-8803](#): Backup Management: Improved error messages to indicate incompatible versions of software
- [PMM-8636](#): Integrated Alerting: Additional context to alerts to better convey issue detected
- [PMM-8644](#): Integrated Alerting: API should allow textual TLS configs for webhooks
- [PMM-8122](#): Integrated Alerting: UI does not indicate a port is needed in configuration for SMTP communication channel
- [PMM-8484](#): Added support for PostgreSQL 14 and Percona Distribution for PostgreSQL 14
- [PMM-7297](#): Updated plugin for Clickhouse data source from 2.1.0 to 2.3.1. This fixes some bugs and eliminates noise from warnings in logs as well as adding support of new types (DateTime64) and improved ARRAY JOIN parsing

## 7.4.4 Bugs Fixed

- [PMM-8975](#): Backup Management: long presentation of recurrent intervals in Backup scheduling
- [PMM-8541](#): Navigating through PMM Settings link at Failed security checks panel takes more than 30 seconds
- [PMM-8387](#): MySQL InnoDB Details dashboard is not in the left menu
- [PMM-8858](#): Dashboards: No Host uptime on Homepage for RDS instances
- [PMM-8611](#): Dashboards: PMM Agents status presented as DOWN while there is no recent data yet on the status
- [PMM-8393](#): Integrated Alerting: Alert rules not executed after upgrading PMM Server running as Docker container
- [PMM-8058](#): Integrated Alerting: Firing alerts disappear after PMM server restart
- [PMM-8089](#): PMM is not exposing data for memory used by MongoDB when it's mapped with the journal. This was inconsistent behavior compared to older versions of PMM.
- [PMM-9100](#): Dashboards: Binary Log related metrics on MySQL Replication dashboard are not prevented and not collected for MySQL8
- [PMM-8633](#): Unfinished queries are included in Query Analytics for PostgreSQL with pg\_stat\_monitor usage because of incorrect use of state\_code.
- [PMM-8859](#): Increased memory consumption on Client-side for PostgreSQL monitoring when executing either too many custom queries or some queries against too many tables
- [PMM-9046](#): Incorrect link to instructions about installing Image Rendering Plugin
- [PMM-8952](#): Query Analytics: No table/indexes information for Views when PostgreSQL server monitored with pg\_stat\_monitor

---

Last update: 2021-12-14

## 7.5 Percona Monitoring and Management 2.23.0

Date: October 21, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.5.1 Release Highlights

- **Grafana Upgrade** Embedded Grafana version was upgraded from 7.5.7 to 7.5.11 to improve some Grafana instrumentation metrics. Also, the upgrade is delivering a security fix that impacted Grafana. Please upgrade to the latest version of PMM ASAP. Read more about CVE issue [here](#)
- **Backup Management (Technical preview):**
  - An ability to view logs of the backup process for better visibility over the backup process
  - An ability to schedule Point-In-Time-Recoverable backup from MongoDB clusters with the correct configuration. Note: there is no UI to restore PITR for MongoDB at the moment. It will come with future releases, but it is possible to restore a PITR backup with [Percona Backup for MongoDB](#) manually
- **DBaaS (Technical preview):** From this release on, PMM users who are using the DBaaS feature will be able to update versions of their DBaaS controlled Databases by the push of a button according to each DB's compatibility matrix. Please note that we recently found a bug [PMM-8723](#) that was causing significant problems with DBaaS usage. This bug was fixed in this release, and no additional actions will be required.

### 7.5.2 New Features

- [PMM-8269](#): Backup Management: Ability to schedule Point-In-Time-Recoverable backups for MongoDB
- [PMM-8159](#): Backup Management: Ability to see a logs of backup process for MongoDB
- [PMM-7519](#): Backup Management: Version compatibility check prior to attempted MySQL data restoration
- [PMM-8200](#): DBaaS: Admin can now initiate a DB version upgrade with just a button click
- [PMM-8273](#): Integrated Alerting: Alert templates delivery from Percona.com for anonymous PMM servers

### 7.5.3 Improvements

- [PMM-8973](#): Grafana upgrade from 7.5.7 to 7.5.11 Includes better Grafana instrumentation metrics and fix for CVE-2021-39226 (read more on [Grafana blog](#))
- [PMM-8653](#): Added titles to Home Dashboard panels for better readability
- [PMM-8669](#): Integrated Alerting: Create a clearer distinction about using PMM Alerting as preferred method vs using an external Alertmanager
- [PMM-8539](#): Wrong Cluster Role presentation on MongoDB Cluster Summary
- [PMM-7559](#): Integrated Alerting: Improve error message when trying to delete a channel that is used by a rule
- [PMM-6763](#): Better color contrast in Time distribution in QAN details
- [PMM-5669](#): New flag `-paths-base` in pmm-agent to avoid problems with hardcoded paths. Please note: this is possible if you run pmm-agent separately from pmm-admin. The ability to specify base paths over pmm-admin is not yet implemented

## 7.5.4 Bugs Fixed

- [PMM-7985](#): Users were losing manually installed Grafana plugins after upgrade via Docker

### Caution

The issue is fixed automatically since 2.23.0 version forward. For the upgrades from versions before 2.23.0 please [backup](#) plugins first.

- [PMM-8767](#): Copied dashboards with tags were ending up in unexpected folder after upgrade
- [PMM-8635](#): MyRocks WAL panel from MySQL MyRocks Details Dashboard presented data in wrong units
- [PMM-8527](#): Dashboards: ProxySQL/HAProxy DB Conns, DB QPS, DB uptime metrics were missing on Home dashboard panels
- [PMM-8749](#): Adding more than 1 `mongos` was breaking MongoDB Cluster Summary dashboard
- [PMM-8004](#): Fixed broken metrics reporting in case of lost connection to MongoDB. (Thanks to Álvaro López López for reporting this issue)
- [PMM-8489](#): Failed to get topology labels when target server is `mongos`
- [PMM-6877](#): Fixed error flooding from when monitoring `mongos` (Thanks to Clyde Shillingford for reporting this issue)
- [PMM-8851](#): Can't monitor GCP Cloud SQL or other PostgreSQL with custom SSL certificates (Thanks to Jyoti Prakash for reporting this issue)
- [PMM-8646](#): PostgreSQL services monitoring was stalled after intermittent connection latency
- [PMM-8723](#): PMM wouldn't restart DBaaS functionality and would break it after upgrade via UI. Affecting versions starting from 2.17.0

---

Last update: 2021-10-21

## 7.6 Percona Monitoring and Management 2.22.0

Date: September 23, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.6.1 Release Highlights

- **DBaaS (Technical preview):** DBaaS users can now use the PMM UI to upgrade existing Clusters to the newer version of the operator without interacting directly with Kubernetes.

### 7.6.2 New Features

- [PMM-8396](#): Backup Management: Add an ability to specify the number of retries for Scheduled backups before operation fails
- [PMM-8079](#): DBaaS: Percona Operators inside Kubernetes cluster managed by PMM can now be upgraded
- [PMM-8077](#): DBaaS: Show the current version of Operators used in Kubernetes Cluster
- [PMM-7924](#): MySQL Performance Details dashboard: Add “Performance Schema Status Monitoring” chart

### 7.6.3 Improvements

- [PMM-8688](#): Backup Management: When adding a new Backup, the action button now reads “Create Backup” instead of “Edit” to reflect the action happening
- [PMM-8311](#): Integrated Alerting: Disable edit/delete buttons for Percona-sourced Templates
- [PMM-8509](#): Management of ability to update PMM Server in the same way as this implemented to other Settings for PMM. Users can use API, UI, or docker Environment Variables to change the setting responsible for the Update process. As with all PMM settings, environment variables have higher priority and can't be changed with the API or in the UI.
- [PMM-7392](#): DBaaS: Change Number of Nodes when editing Topology

### 7.6.4 Bugs Fixed

- [PMM-8613](#): Backup Management: Unable to restore MySQL backup
- [PMM-8463](#): Backup Management: State stuck on “Pending” when creating backup with already existing name
- [PMM-8408](#): DBaaS: Development version of PMM Client was used for monitoring DB Clusters created by DBaaS
- [PMM-8584](#): Wrong CPU metric labels in dashboards for RDS instances
- [PMM-8421](#): Listen-port ignored/removed for external services after server update to PMM 2.19 and higher (Thanks to Rainer Plischke for reporting this issue). **Please make sure to upgrade PMM Server to avoid loss of external exporter listen port ([PMM-8829](#)) and always upgrade PMM Server before PMM Client ([PMM-8854](#)).**
- [PMM-8703](#): Custom dashboard prevents PMM Server Docker update from 2.20 to 2.21 (Thanks to Hubertus Krogmann for reporting this issue)
- [PMM-7690](#): AWS discovery and monitoring based on IAM roles is not working

If you have problems upgrading your pmm-2-client packages, try clearing caches with:

```
sudo apt-get clean
```

and remove files manually with:

```
cd /var/cache/apt/archives && sudo rm -rf /*
```

---

Last update: 2021-09-27

## 7.7 Percona Monitoring and Management 2.21.0

Date: August 26, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.7.1 Release Highlights:

- **Custom certificates support:** We have added support for custom TLS certificates for remote monitoring of PostgreSQL and MongoDB services, configurable on the command line with `pmm-admin` or through the UI.
- **Backup scheduling with retention** (Technical Preview): When scheduling a backup you can now specify how many of the most recent backups to keep. Backups not in this range are automatically deleted to free space for new ones.
- New supported versions:
  - DBaaS functionality now supports Kubernetes Operator for MongoDB version 1.9.0.
  - PMM Client packages now support Debian 11 ("Bullseye").

### 7.7.2 New Features

- [PMM-8158](#): Backup Management: Delete option — When deleting a backup from Backup Management inventory a new option lets you also delete the data file from storage.
- [PMM-8156](#): Backup Management: Retention — You can now define how many of the most recent scheduled backups to keep.
- [PMM-8214](#): Ability to collect Kubernetes cluster metrics — Collection only, metrics are not currently presented on any PMM dashboard.
- [PMM-7477](#): Support custom TLS certificates when monitoring remote MongoDB instances
- [PMM-7888](#): Custom TLS certificates now allow SSL connections to PostgreSQL instances (Thanks to Jyoti Prakash for reporting this issue)

### 7.7.3 Improvements

- [PMM-8267](#): Backup Management: Active progress indicator
- [PMM-8549](#): Backup Management: Show loading status on delete window
- [PMM-8542](#): Backup Management: Inform that times should be entered in UTC timezone format
- [PMM-8316](#): DBaaS: PSMDB 1.9 operator support —For what's new see [release notes](#).
- [PMM-7612](#): Integrated Alerting: Validate communication settings 'From' email address format
- [PMM-7570](#): Specify Custom Basic Auth password for Agents when adding Services
- [PMM-8560](#): Add support for Debian 11 ("Bullseye") to `pmm-client` package
- [PMM-7087](#): Rename custom query file to `example-queries-postgres.yml` and include warning that the file will be overwritten on upgrade; user should create a copy with a new name to prevent losing metrics collection on future upgrades. (Thanks to Daniel Guzman Burgos for reporting this issue)
- [PMM-8568](#): Use latest CentOS patches for creating OVA, AMI and Azure images
- [PMM-5291](#): Update ClickHouse version from 19.7.5.27 to 21.3-lts
- [PMM-8091](#): Collect and present additional ProxySQL metrics taken from `runtime_mysql_servers` table

## 7.7.4 Bugs Fixed

- [PMM-8616](#): Backup Management: No ‘Delete from storage’ action on backup inventory
- [PMM-8543](#): Backups are not visible after PMM Server upgrade
- [PMM-8458](#): Backup Management: Inconsistent auto-fill of ‘Vendor’ field with on-demand backup
- [PMM-8404](#): Dashboard image rendering plugin renders image that includes error message
- [PMM-7286](#): Query Analytics can’t handle colon character (:) in service names (Thanks to Daniel Guzman Burgos for reporting this issue)
- [PMM-7278](#): `mongo_exporter` fails to authorize when MongoDB running with `authMechanism=MONGODB-X509` (Thanks to Lena D for reporting this issue)
- [PMM-8307](#): Default config limits for allowed connections prevents monitoring large numbers (500+) of DB servers
- [PMM-2168](#): `rds_exporter` not stopped when all RDS instances are removed or disabled
- [PMM-8219](#): PMM Server update panel “Current version” empty if no internet connectivity
- [PMM-8559](#): Unauthorized error appears while logging in

## 7.7.5 Known Issues

- Failure to upgrade when using a dashboard with custom tags.

### Important

In some cases users may not be able to complete the upgrade to 2.21.0 and we have linked this back to dashboards with custom tags. This is to be fixed in our upcoming 2.22.0 release but there are steps (more in the [ticket](#)) that you can take if you’re already impacted by this:

```
curl -LJ0s https://raw.githubusercontent.com/percona/pmm-server/c2e92bc3aec123affda5f1992c96c95ac74f4a2d/import-dashboards.py  
docker cp import-dashboards.py pmm-server:/usr/share/percona-dashboards/  
docker exec -it pmm-server chmod a+x /usr/share/percona-dashboards/import-dashboards.py
```

Last update: 2021-09-17

## 7.8 Percona Monitoring and Management 2.20.0

Date: August 3, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.8.1 Release Highlights

- The [Easy-install script](#) for PMM Server is checksum verified.
- PMM will use [Grafana 7.5](#) instead of 7.3. We are also preparing for Grafana 8.
- PostgreSQL monitoring with the `pg_stat_monitor` plugin enabled exposes new metrics such as Plan Times, WAL Fpi/Bytes/Records.
- For users who deploy PMM Server through the AWS Marketplace, AWS RDS service discovery will be executed without AWS credentials and tuning [IAM roles](#).
- For Backup Management (Technical Preview), we added the ability to schedule backups so you can schedule and see already scheduled [backups](#) in the UI.

### 7.8.2 Important note for users of PMM who started out using the Docker image of 2.16.0

If you installed PMM [version 2.16](#) as a new Docker image and have since used the home dashboard upgrade widget to upgrade to any of 2.17, 2.18, or 2.19, you might experience problems with monitoring the PMM server itself, Remote Monitoring, or RDS/Azure monitoring. If you experience any of these problems, you can simply run the following commands to get your instance working and it will be automatically resolved in the next release:

1. Enter the container: `docker exec -it pmm-server bash`
2. Roll back `pmm2-client` package to stable version: `yum downgrade -y pmm2-client`

Alternatively, you can replace the existing Docker container with a fresh install of the latest release by following the [official instructions for an upgrade](#). (This will guide you through taking a backup of your PMM Server and restoring it after installing a fresh docker image of PMM Server.)

### 7.8.3 New Features

- [PMM-8157](#): Backup Management: Show scheduled backups – A new view that shows the list of scheduled backups with quick actions to enable/disable, edit, copy, and delete.
- [PMM-8155](#): Backup Management: Scheduling of backups – Support for Backup Scheduling has been added so that users can define backup jobs to run automatically in the future with the option of making the schedules recurring.
- [PMM-7010](#): Option to unregister current node (`pmm-admin unregister`)

### 7.8.4 Improvements

- [PMM-7552](#): PMM Server Docker image: Add labels to PMM Server Docker image (Thanks to Nicolas for reporting this issue)
- [PMM-8266](#): PMM Server Docker image: Decommission and remove Prometheus binaries and configuration
- [PMM-8040](#): PMM Server Docker image: Remove yum cache to reduce size of image
- [PMM-7809](#): Grafana upgrade from 7.3.7 to 7.5.7 – [Read more at grafana.com](#)

- [PMM-8386](#): Overview text on the Home page is missing PostgreSQL as a supported technology
- [PMM-7722](#): DBaaS: Announce new supported version of operator – Shows that a new version of the operator is available.
- [PMM-6278](#): Modification of MySQL ‘Delete’ queries to provide ‘Explain’ information
- [PMM-8468](#): Forbid the use of outdated ciphers for HTTPS protocol on exporters
- [PMM-7649](#): Security Checks: Show “Insufficient access permissions” in UI for non admin users
- [PMM-8059](#): Update Query Analytics UI to clarify estimated results on MySQL ‘explain’ response where we modified original query
- [PMM-8043](#): Return Service Name in `GetCheckResults` API response
- [PMM-8000](#): Expose new numbered metrics available in `pg_stat_monitor` 0.9

### 7.8.5 Bugs Fixed

- [PMM-8299](#): Backup Management: Remove storage location shows wrong notification – When removing a storage location, PMM presents misleading information to the user in the form of two notification messages for both error and success.
- [PMM-8283](#): Backup Management: Error when removing location with ‘force delete’ option
- [PMM-8064](#): Dashboards: Size of Temp Files Report Metric value has wrong unit on PostgreSQL Instance Summary Dashboard
- [PMM-6981](#): Dashboards: Wrong version is shown for MariaDB services
- [PMM-7738](#): Integrated Alerting: Alerts for some built-in templates missing service name label
- [PMM-6877](#): `mongodb_exporter` doesn’t recognize being on a mongos host and fills the syslog with `replSetGetStatus` errors (Thanks to Clyde Shillingford for reporting this issue)
- [PMM-7627](#): Consistent PMM Server log management – Adds consistency to the log management of `nginx`, `postgresql` and `clickhouse-server`, which is now delegated to `supervisord`. Removes the `logrotate` daemon from the image.
- [PMM-8492](#): PMM Client version is 2.21.0 inside PMM Server after upgrade from 2.16.0

### 7.8.6 Known Issues (unfixed problems that you should be aware of)

- [PMM-8414](#): Backup Scheduler not working if user specifies explicit job start time

---

Last update: 2021-08-24

## 7.9 Percona Monitoring and Management 2.19.0

Date: June 30, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

Visit our [forum](#) to comment on this release.

### 7.9.1 Release Highlights

- Backup Management can now be enabled from the UI. Go to *Configuration* → *Settings* → *Advanced Settings*, and you will see it in the *Technical Preview* section. We also added support for MongoDB services on-demand backup and restore. For now, it only supports ReplicaSet on S3-compatible storage.
- Dashboards improvements
  - There are several community-driven improvements to ProxySQL data collection, with new dashboards to expose such metrics like: *Queries Latency histograms* and `SHUNNED_REPLICATION_LAG` state.
  - Fixes for Amazon Aurora service detection on [the dashboard](#), MongoDB ReplicaSet Summary, and other MongoDB memory-related panels.
- Improvements to DBaaS secrets by generating strong passwords for operators. This is an improvement to the Automated Operator Installation released in [PMM 2.18](#), which will greatly enhance security.

### 7.9.2 New Features

- [PMM-7639](#): Backup Management: Ability to remove Backup Location even if there are some backup artifacts on it
- [PMM-7567](#): Backup Management: Simple backup for MongoDB ReplicaSet
- [PMM-7568](#): Backup Management: Simple restore for MongoDB ReplicaSet

### 7.9.3 Improvements

- [PMM-8112](#): Dashboards: Collect and present histograms from ProxySQL on Queries Latency dashboard (Thanks to [foosinn](#) for helping with this improvement)
- [PMM-8081](#): Dashboards: Add collection and presentation for `SHUNNED_REPLICATION_LAG` state to `proxysql_connection_pool_status` (Thanks to [spaceform](#) for helping with this improvement)
- [PMM-7584](#): Components Upgrade: VictoriaMetrics 1.53.1 to [1.60.0](#)
- [PMM-8001](#): Better error handling when `pg_stat_monitor` is an unsupported version
- [PMM-7659](#): DBaaS: Ability to specify the type of connection for DBaaS cluster during DB Cluster creation
- [PMM-7828](#): DBaaS: Select Database Type by default if only one operator is installed
- [PMM-8153](#): Backup Management: Disable 'Restore' button for backups whose service has been deleted

### 7.9.4 Bugs Fixed

- [PMM-7194](#): 'Share with Percona' option doesn't export data from collapsed panels
- [PMM-8060](#): User cannot add remote instances after OVF/AMI upgrade (previously was Known Issue)

- [PMM-7104](#): Slowlog rotation by pmm-agent causing additional unexpected rotation of binary logs (Thanks to Arthur Tokarchuk for reporting this issue)
  - [PMM-8125](#): Error of monitoring GCP based PostgreSQL because of internal database `cloudsqladmin`
  - [PMM-6778](#): Can't specify custom labels during node addition/configuration
  - [PMM-8090](#): Multi-request protection breaks metrics gathering (Thanks to [Francisco Miguel Biete](#) for fixing this bug)
  - [PMM-5248](#): InnoDB TableSpace data is not collected for Percona Server 8
  - [PMM-4665](#): User's log file contains error messages about TokuDB and heartbeat despite not being used
  - [PMM-8014](#): Error when adding Amazon RDS MySQL with TLS over API
  - [PMM-3352](#): Low file descriptors limit (1024) with AMI or OVF images causes errors
  - [PMM-7948](#): `pmm-admin list` reports the wrong Node for External Services
  - [PMM-6295](#): Unclear/Incorrect statuses of the pmm-agent when Agent or PMM server went down (Thanks to Mikhail Solovyev for reporting this issue)
  - [PMM-5917](#): pmm-agent moves slow logs without checking privileges
  - [PMM-8021](#): "Query Analytics" misspelled on left side menu
  - [PMM-5283](#): Inconsistency with lengths of Example and Fingerprints in Query Analytics
  - [PMM-8121](#): Error message and help for "Remove Service" command is not helpful
  - [PMM-8196](#): Additional spaces in email/passwords fields on Sign up/Login pages causes Authentication problems
  - [PMM-8009](#): Long First/Last names causes errors when used in Register/Login form
  - [PMM-8220](#): Dashboards: Active Time Series Changes on Victoria Metrics dashboard report no data
  - [PMM-8202](#): Dashboards: No Amazon Aurora services are available on MySQL Amazon Aurora Details dashboard for selection
  - [PMM-8085](#): Dashboards: Wrong units are used in MongoDB dashboards on memory-related panels
  - [PMM-7154](#): Dashboards: No data on some panels from MongoDB ReplSet Summary dashboard
  - [PMM-8115](#): DBaaS: Delete PSMDB cluster action takes too long
  - [PMM-7737](#): DBaaS: Replace all default passwords in operator secrets during installation
  - [PMM-7970](#): DBaaS: Confusing message for Cluster name pattern on DB Cluster creation screen
  - [PMM-7755](#): DBaaS: Clusters with longer name not initializing
  - [PMM-7528](#): DBaaS: Error after Kubernetes cluster destroyed or removed externally to DBaaS
  - [PMM-8013](#): Backup Management: Unable to get Backup Artifact list after Service removal
- 

Last update: 2021-07-06

## 7.10 Percona Monitoring and Management 2.18.0

Date: June 1, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.10.1 Release Highlights

The goal for this small release was to eliminate a lot of bugs and complete some features.

- DBaaS

Added the ability for PMM to install the latest versions of the K8s operator into the K8s cluster. There is no longer any need to install the operator manually. Just connect PMM to your K8s cluster and let PMM do the rest.

- Backup Management

Backup functionality was released as a Technical Preview feature and will require specific prerequisites from the user side to be installed. Currently, PMM will allow you to:

- manage storage for backups (S3 only);
- execute a backup for MySQL and Mongo instances;
- restore a MySQL backup to the same instance from where it was taken (via the UI).

Restore in other cases is not yet implemented on the UI.

### 7.10.2 New Features

- [PMM-7509: Integrated Alerting: Pagination for Alerts list](#)

### 7.10.3 Improvements

- [PMM-8029: DBaaS: PSMDB 1.8 operator support](#)
- [PMM-7548: Integrated Alerting: Disable edit and delete buttons for templates manually staged by user directly on the file system](#)

### 7.10.4 Bugs Fixed

- [PMM-8053: Better error handling for non-admins who try to access the settings page](#)
- [PMM-7941: Wrong replication status for MongoDB Replica Set](#)
- [PMM-7302: Webhook usage with `max\_alerts` attribute for Alertmanager configuration causes errors for PMM](#)
- [PMM-7224: Instance Overview dashboards behave inconsistently](#)
- [PMM-6864: MongoDB Oplog Recovery Window dashboard is broken \(Thanks to Clyde Shillingford for reporting this issue\)](#)
- [PMM-7910: MongoDB Query metrics stops being collected if the cursor is failed once \(Thanks to Yann Rouillard for reporting this issue\)](#)
- [PMM-6451: Passing parameters between Query Analytics and Dashboards is broken](#)
- [PMM-5368: Unclear message “Failed to get PMM Server parameter” after configuration \(Thanks to Martin Wittwer for reporting this issue\)](#)
- [PMM-5135: Query Example is often empty for MySQL 8+ \(Thanks to Mikhail Solovyev for reporting this issue\)](#)

- [PMM-8083](#): Better configuration file checking during configuration
  - [PMM-7958](#): Databases cannot be deleted while PostgreSQL is being monitored
  - [PMM-6553](#): Slow log size units are not defined in help
  - [PMM-5931](#): Graph and values in Query Analytics are identical for TOTAL and case when data is Not Available
  - [PMM-5538](#): Heavy Load with Distinct Queries on Slowlog enabled could cause no data being reported
  - [PMM-8095](#): The link to the Community section in the PMM footer is broken
  - [PMM-7982](#): Query Analytics: Sorting element for the columns is hard to access
  - [PMM-6676](#): Terms and Privacy pages opened in the current tab complicates registration process
  - [PMM-6552](#): Do not register with server if configuration file fails to create on client
  - [PMM-6505](#): Inconsistent style for error messages on Add RDS instance page.
  - [PMM-8069](#): Integrated Alerting: Alert template now accepts `.yaml` extension in addition to `.yml` when manually staging on the file system
  - [PMM-7673](#): Integrated Alerting: Actions column is transparent
  - [PMM-7916](#): DBaaS: Wrong required resources when editing a cluster
  - [PMM-7753](#): DBaaS: Edit DB Cluster shows wrong values by default
  - [PMM-7184](#): DBaaS: Connection column showing different values after deleting DB cluster
  - [PMM-8088](#): DBaaS: In case of error, the kubeconfig file is left in the system
- 

Last update: 2021-06-25

## 7.11 Percona Monitoring and Management 2.17.0

Date: May 11, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.11.1 Release Highlights

- Custom certificates help define proper security levels for remotely monitored MySQL instances, including Google Cloud SQL.
- Usability improvements to the External Monitoring UI. When filling parameters, you can enter the parts of an endpoint (scheme, host, path) or let PMM automatically extract them from a URL.
- pg\_stat\_monitor 0.9.0 support. This change will give you compatibility with the latest version. Support for new features will be in an upcoming release.
- Single-line install of PMM Server on supported Linux distributions (this feature is in Technical Preview).
- DBaaS Changes: (this feature is in Technical Preview)
  - It is easier to experience DBaaS functionality; you can quickly turn it ON/OFF in *Advanced settings* on the *Settings* page. ([Read more](#))
  - Database components management will enable PMM administrators to limit users in your organization to specific (admin-approved) database versions in their DBaaS DB Clusters.
  - For PXC clusters created using DBaaS, HAProxy will now be used by default. Please note: Monitoring of the HAProxy in DBaaS will be enabled in an upcoming release.
- Changes to *Sign in to Percona Platform*. From this release, Registration of the Percona account will be more secure and require additional confirmation.

### 7.11.2 New Features

- [PMM-7863](#): DBaaS: Ability to specify in K8s configuration the version of HAProxy to be used for DB creation
- [PMM-7848](#), [PMM-7847](#), [PMM-7421](#): Add support for using SSL certificates between pmm-admin and monitored MySQL databases
- [PMM-7883](#): Single-line install of PMM Server on supported Linux distributions - [Technical Preview]
- [PMM-7013](#), [PMM-7819](#): DBaaS: Use HAProxy by default instead of ProxySQL for MySQL DB clusters
- [PMM-7356](#), [PMM-7581](#): DBaaS: Management of available versions of DB components
- [PMM-7358](#), [PMM-7576](#): DBaaS: Management of default versions of DB components

### 7.11.3 Improvements

- [PMM-7572](#): Add TLS options to mysqld\_exporter
- [PMM-7783](#): Support of pg\_stat\_monitor 0.9.0
- [PMM-7064](#): Integrated Alerting: Presenting severity of the Alert Rule using different colors
- [PMM-7946](#): Better error message on PMM client if server doesn't support HAProxy
- [PMM-7932](#): Usability improvements on UI for adding External Services
- [PMM-7641](#), [PMM-7820](#): Add DBaaS to Technical Preview section and allow user to Enable/Disable via UI
- [PMM-7966](#): Telemetry: Collect enabled/disabled status for Integrated Alerting and Security Threat Tool features

#### 7.11.4 Bugs Fixed

- [PMM-7911](#): DBaaS: Invalid Number of Nodes results in an annoying error message pop-up
  - [PMM-7884](#): DBaaS: Fix DB Cluster tab loading
  - [PMM-7917](#): PostgreSQL exporter has high CPU usage during Restart
  - [PMM-8037](#): User can create a Percona Platform account without proper confirmation
  - [PMM-7702](#): DBaaS: Cannot edit already-created PSMDB clusters
  - [PMM-7991](#): MySQL Summary panel doesn't exist on MySQL Summary dashboard
  - [PMM-7939](#): Inconsistent format of version reporting in pmm-admin
  - [PMM-7920](#): PostgreSQL Exporter has increased memory usage with pmm-client 2.15.1 & pmm-server 2.16.0
  - [PMM-7700](#): Integrated Alerting: Rule API crashing with more than two parameters or invalid values
  - [PMM-7616](#): Integrated Alerting: Incorrect title of the page in a browser
  - [PMM-7396](#): Integrated Alerting: Alerts tab error if user deletes Alert Rule which has Firing alerts
- 

Last update: 2021-06-25

## 7.12 Percona Monitoring and Management 2.16.0

Date: April 15, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.12.1 Important note for users of PMM 2.16.0

If you started using PMM from version 2.16 and have already upgraded to 2.17, 2.18, or 2.19, you might have some problems with PMM Server monitoring, Remove Monitoring, or RDS/Azure monitoring. If you experience a problem, we recommend you upgrade and replace the Docker container by following the official instructions for an upgrade here: <https://www.percona.com/doc/percona-monitoring-and-management/2.x/setting-up/server/docker.html#upgrade>.

If you can't do this, then you need to perform additional steps after upgrading to 2.20.

1. Enter the container: `docker exec -it pmm-server bash`
2. Roll back `pmm2-client` package to stable version: `yum downgrade -y pmm2-client`

### 7.12.2 Release Highlights

- **Amazon RDS PostgreSQL monitoring**

AWS monitoring in PMM now covers PostgreSQL RDS and PostgreSQL Aurora types. PMM will include them in a [Discovery UI](#) where they can be added which will result in node related metrics as well as PostgreSQL database performance metrics. Before this release, this was available only to MySQL-related instances from Amazon RDS.

- **Azure Discovery and Node metrics extraction**

**Technical Preview:** PMM will have the same level of support for Microsoft Azure Database as a Service (DBaaS) as we have for AWS's DBaaS (RDS/Aurora on MySQL or PostgreSQL). You will be able to easily discover and add Azure databases for monitoring by PMM complete with node-level monitoring. This feature is available only if you explicitly activate it on the [PMM Settings](#) page. Deactivating it will not remove added services from monitoring, but will just hide the ability to discover and add new Microsoft Azure Services.

(This is a feature technical preview because we want to release it as soon as possible to get feedback from users. We are expecting to do more work on this feature to make it more API and resource efficient.)

- **Security Threat Tool Scheduling - Manage execution and execution intervals**

Security Threat Tool users are now able to control the Security Check execution time intervals for groups of checks, move checks between groups, and disable individual checks if necessary.

- **Support for pg\_stat\_monitor 0.8**

Added compatibility with [pg\\_stat\\_monitor plugin v 0.8.0](#). This is not exposing the new features for the plugin in PMM yet, but ensures Query Analytics metrics are collected to the same degree it was with version 0.6.0 of the plugin.

- **Consistent support of Technical Preview Features**

Reworked the PMM Settings page to make it clear what features are in Technical Preview vs General Availability (GA) and to simplify activation/deactivation of technical preview features. We also provide a better definition of what a [Technical Preview](#) is.

- **Migration of Settings and other service pages in PMM from Grafana dashboards**

The *PMM Settings* page and several others (including *Add Instance* and *Inventory*) are being converted to Grafana pages and will no longer be presented as dashboards. Additionally, we're moving the menu to the sidebar navigation for consistency and more flexibility compared to the older menu structure.

- **Integrated Alerting improvements**

We released the next stage of improvements in Integrated Alerting functionality of PMM to simplify the usage of the feature. Together with improvements, we continue fixing known bugs in this feature.

- **[DBaaS] Resource planning and prediction (Resource calculator)**

**Technical preview:** While creating a DB cluster a user can see a prediction of the resources this cluster will consume with all components as well as the current total and available resources in the K8s. Users will be warned that if they attempt to create a DB cluster it may be unsuccessful because of available resources in the K8s.

- **[DBaaS] PSMDB 1.7.0 operator support**

DBaaS in PMM will be using the recently-released [Percona Kubernetes Operator for Percona Server for MongoDB 1.7.0](#) to create MongoDB clusters.

### 7.12.3 New Features

- [PMM-7313](#), [PMM-7610](#): Ability to discover and monitor Amazon RDS PostgreSQL instances with collecting PostgreSQL and RDS node metrics (Thanks to Daniel Guzman Burgos for reporting this issue).
- [PMM-7345](#): Expose metrics for all available databases on a PMM monitored PostgreSQL server.
- [PMM-7344](#): Update `postgres_exporter` version from 0.4.6 to 0.8.0. (See the full list of improvements in the [changelog](#).)
- [PMM-7767](#), [PMM-7696](#): Implement feature flag to enable Microsoft Azure monitoring. Users can [use the UI](#) or set an environment variable (`ENABLE_AZUREDISCOVER=1`) during container creation.
- [PMM-7684](#), [PMM-7498](#): Ability to discover running and supported Microsoft Azure Databases instances in a provided account.
- [PMM-7678](#), [PMM-7679](#), [PMM-7676](#), [PMM-7499](#), [PMM-7691](#): Prepare, modify and use `azure_exporter` to collect Node related metrics.
- [PMM-7681](#): Use Microsoft Azure metrics on Node/OS-related dashboards to show the metrics on panels.
- [PMM-7339](#): Security Threat Tool: Ability to execute security checks individually and on-demand.
- [PMM-7451](#), [PMM-7337](#): Security Threat Tool: Ability to change intervals for security checks on the *PMM Settings* page.
- [PMM-7722](#), [PMM-7338](#): Security Threat Tool: Ability to change default execution interval per check.
- [PMM-7336](#): Security Threat Tool: Execute checks based on execution interval they belong to.
- [PMM-7335](#): Security Threat Tool: Ship security check files with predefined execution interval.
- [PMM-7748](#): Add an additional experimental menu for Dashboards on the left side panel.
- [PMM-7688](#): Unify UX and layout of all PMM specific pages like Settings, Add Instance etc.
- [PMM-7687](#): Modify links in menus to ensure both menus are working as expected after dashboard URL change.
- [PMM-7705](#): Simplify display of features in technical preview to easily identify them and their current state.
- [PMM-7522](#), [PMM-7511](#): Integrated Alerting: Improve Notification Channels UX by Pagination for the Notification list.
- [PMM-7521](#), [PMM-7510](#): Integrated Alerting: Improve Alert Rule Templates UX by Pagination on Rule Templates list.
- [PMM-7652](#), [PMM-7674](#), [PMM-7503](#), [PMM-7486](#): DBaaS: While creating the DB cluster see all and available resources in the K8s cluster, such as Disk, CPU & Memory.
- [PMM-7508](#), [PMM-7488](#): DBaaS: See predicted resource usage for selected DB Cluster configuration.

- [PMM-7364](#): DBaaS: Show warning before starting creating the cluster if there are not enough resources in the K8s cluster to create DB Cluster with requested configuration.
- [PMM-7580](#), [PMM-7359](#): DBaaS: Users can select the database version to use during DB Cluster creation.

#### 7.12.4 Improvements

- [PMM-7506](#): Security Threat Tool: Reduce False Positives due to Roles automatically created in PXC with no password but cannot be used to login.
- [PMM-7569](#): Make PMM compatible with pg\_stat\_monitor 0.8 release.
- [PMM-7571](#): Modified Percona Platform Account registration flow from PMM server UI.
- [PMM-7513](#): Integrated Alerting: Ability to see default values and Threshold values during the Alert Rule creation.
- [PMM-7461](#): Integrated Alerting: Improve UX of tables presentation and loading on UI.
- [PMM-7375](#): Integrated Alerting: Inform users about the template that they are editing and warn them about the limitations.
- [PMM-7260](#): Integrated Alerting: Make it clearer what rule is being edited.

#### 7.12.5 Bugs Fixed

- [PMM-7131](#), [PMM-7555](#): QAN for PostgreSQL attempts to connect to a database with the same name as the username. (Thanks to Daniel Guzman Burgos for reporting this issue)
- [PMM-7481](#): Query Analytics is not showing “Query with Errors” in the Profile section.
- [PMM-7464](#): NGINX misconfiguration leads to log storm in push mode
- [PMM-7434](#): Integrated Alerting: Unknown parameters [threshold] error during Add/Update Alert Rule.
- [PMM-7231](#): Integrated Alerting: Disabling channels does nothing.
- [PMM-7379](#): Integrated Alerting: Can not edit Alert Rule Name through API.
- [PMM-7232](#): Integrated Alerting: Disabling IA does not disable rules evaluation and notifications sending.
- [PMM-7119](#): Integrated Alerting: Update error notification for adding/update Alert rule Template – There was inconsistent behavior if you tried to add a new Rule Template with an already-used name.
- [PMM-7543](#): Integrated Alerting: selected section disappears from a breadcrumb after clicking the tab for a second time.
- [PMM-7766](#): DBaaS: PMM Upgrade breaks DBaaS get credentials method.
- [PMM-7351](#): DBaaS: Safari does not accept float numbers as a custom option in the “Create Cluster” dialogue.
- [PMM-7701](#): DBaaS: PSMDB clusters stuck in initializing due to special characters in secrets.

---

Last update: 2021-08-03

## 7.13 Percona Monitoring and Management 2.15.1

Date: March 18, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.13.1 Release Highlights

This patch release fixes performance issues discovered in systems, together with other small fixes.

### 7.13.2 Bugs Fixed

- [PMM-7635](#): Fix high CPU consumption by Grafana server after upgrade by docker replacement to 2.15.0 with large numbers of services in 'push' mode.
  - [PMM-7713](#): Fix high CPU and Memory consumption by Victoria Metrics after upgrade by docker replacement to 2.15.0 with large numbers of services in 'pull' mode.
  - [PMM-7470](#): MongoDB exporter `IndexStatsCollections` is assigned values from wrong flag (intended for 2.15.0, omitted due to missing merge cutoff) (Thanks to Tim for reporting this issue).
  - [PMM-1531](#): Metrics not being collected due to rename of MySQL 8 information schema tables.
- 

Last update: 2021-06-25

## 7.14 Percona Monitoring and Management 2.15.0

Date: March 01, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.14.1 Release Highlights

- PMM 1 vs. 2 Parity

#### **Disable collectors during adding node/service to monitoring**

With this feature users can disable any collector used by PMM to get metrics. When metrics cannot be collected or are no longer needed, disabling the collector(s) prevents PMM from flooding logs and saves infrastructure resources.

Our vision for PMM collectors is to provide “stop from collecting” functionality to prevent possible harm to the user environment. This “disable” feature is an initial step towards the ideal functionality. The full and flexible management for “What metrics to collect and in what resolution” is slated for future releases.

#### **External services monitoring**

Since PMM 1.4.0, users had the ability to monitor external services Percona didn’t currently support (e.g., Redis). [This blog article from 2018](#) nicely described external services monitoring at that time. (At that time Percona was not natively supporting a PostgreSQL monitoring service and so this was listed as an external service. Today, PostgreSQL is natively supported by PMM.)

Until now, PMM 2.x didn’t support external services monitoring. With this release, any non-natively supported by PMM service will now become supported with external services monitoring. You can see the list of possible exporters to be used in <https://prometheus.io/docs/instrumenting/exporters/>. Natively-supported services will continue to deliver an expanded set of metrics and insights.

#### **Provide summary information for systems (`pt-*-summary` actions)**

With the addition of `pt-*-summary` in PMM 2, users can now view summary information about services and nodes on PMM’s dashboard. This summary information is in the industry common format of `pt-*-summary` tools output to simplify portability of this data. This format will also be preserved in the snapshot of the dashboard shared with Percona Support to simplify investigations of issues.

Note: `pt-*-summary` includes formats for:

- `pt-mysql-summary`
- `pt-mongodb-summary`
- `pt-pg-summary`
- `pt-summary`

#### • HAProxy support by PMM

Users are able to add HAProxy Services for monitoring in PMM2. The support level of them in PMM will be the same we have for ProxySQL, so they will be presented in Inventory and on Dashboard. This will allow users who use HAProxy in their HA configuration to also have this component monitored by PMM. In future releases PMM will start use HAProxy by default for the DBaaS feature and will also use this functionality to monitor HAProxy.

#### • DBaaS Preview improvements (Technical Preview)

From now you will be able to see the progress of internal steps the system makes when executing some operations with DBaaS. The Progress Bar will not be time-related and will present only steps. The Progress Bar component will also reflect the K8s/Operator-related errors to the user, so in the case of errors, you will have the

error text on the UI, and no need to use K8s tools to see the error. With the same UI, you will be able to see the latest logs from K8s so they will have even more information about why the error happened.

**Known Limitations:** The progress bar will not provide valuable information for the Delete operation (will be in a later version when we'll change the API with Operators Team), Operation of DB Cluster Modification will have "strange" behavior and will start changes from non-zero values of steps. (This will be modified after API changes.)

### 7.14.2 New Features

- [PMM-4172](#), [PMM-4306](#), [PMM-5784](#), [PMM-7177](#): Services and Nodes Summary presentation. Present information about DB's and Node status using `pt-mysql-summary`, `pt-mongodb-summary`, `pt-pg-summary` outputs (in API and on Dashboards).
- [PMM-7123](#): Ability to add External Services via the UI in PMM server.
- [PMM-6711](#): Add `external-group` flag for `pmm-admin inventory` commands for simpler work with External services.
- [PMM-7405](#): Check connection response format when adding External Service to monitoring.
- [PMM-6797](#): HAProxy monitoring: Ability to add HAProxy services with `pmm-admin [inventory] add [service] haproxy` command.
- [PMM-7487](#): HAProxy monitoring: Check connection to HAProxy services when adding them for monitoring.
- [PMM-7496](#): HAProxy monitoring: New HAProxy PXC dashboards.
- [PMM-6943](#): HAProxy monitoring: Show HAProxy type services in PMM Inventory.
- [PMM-6924](#): Integrated Alerting: Show 'breadcrumbs' navigation aid on non-dashboard pages as well as Grafana dashboard pages.
- [PMM-7294](#): Integrated Alerting: Pagination for viewing large numbers of Alert Rules.
- [PMM-7417](#): Security Threat Tool: Show list of all available security checks.
- [PMM-7418](#): Security Threat Tool: Ability to disable specific security checks.
- [PMM-7419](#): DBaaS: Ability to see DB Cluster creation/modification logs.
- [PMM-7266](#): DBaaS: Cluster creation progress bar – You can now see the progress of DBaaS DB cluster creation. (The progress bar is based on the number of back-end technical steps, not the time required to perform the tasks.)

### 7.14.3 Improvements

- [PMM-4679](#): Docker: `:latest` tag for `pmm-server` and `pmm-client` images has been moved from v1 latest release to v2 latest release. **Note:** use of the latest tag is not recommended in production environments, instead use `:2` tag.
- [PMM-7472](#): Remove Prometheus data source – If you were using custom dashboards with a specified data source (not using empty to use default one) you may need to edit your dashboards to use the proper data source. PMM is no longer using Prometheus but uses compatible storage for metrics from VictoriaMetrics. We renamed the data source to be more technology-agnostic.
- [PMM-6695](#): Software update: Grafana 7.1.3 to 7.3.7 (See [What's new in Grafana 7.2](#) and [What's new in Grafana 7.3](#).)
- [PMM-7471](#): Software update: VictoriaMetrics 1.52.0 to 1.53.1 (See [VictoriaMetrics 1.53.0](#) and [VictoriaMetrics 1.53.1](#).)
- [PMM-6693](#): API keys usage – PMM users can now use API keys (generated in Grafana UI) for interaction with PMM server instead of username/password pairs. The API key should have the same level of access (Admin or Viewer) as is required for username/password pairs.

- [PMM-7240](#): DBaaS: Change from Dashboard to Grafana Page – We changed the DBaaS page from a Grafana Dashboard to a Grafana Page to be better aligned with the DBaaS enable/disable status and avoid confusion when DBaaS is disabled.
- [PMM-7328](#): Security Threat Tool: Download and run checks when activated, immediately, repeating every 24 hours thereafter (Previously, downloading and running new checks happened every 24 hours but the cycle didn't begin when STT was activated.)
- [PMM-7329](#): Security Threat Tool: Hide check results tab if STT is disabled.
- [PMM-7331](#): Security Threat Tool: Failed checks have 'Read more' links with helpful content.
- [PMM-7422](#): Security Threat Tool: View all active and silenced alerts.
- [PMM-7257, PMM-7433](#): Integrated Alerting: Easier-to-read rule details in Alert Rules list (API and UI presentation).
- [PMM-7259](#): Integrated Alerting: Better UI error reporting for disabled Integrated Alerting. (Hint to users how to enable it.)
- [PMM-5533](#): Better indentation of columns in `pmm-admin list` output.
- [PMM-5888](#): Improve `pmm-admin --help` descriptions for external services.

#### 7.14.4 Bugs Fixed

- [PMM-5837](#): `pmm-agent` reports "Malformed DSN" error when adding PostgreSQL instance with a PMM user password containing = (equals sign) (Thanks to Alexandre Barth for reporting this issue).
- [PMM-5969](#): Removing Services or Nodes with `pmm-admin ... --force` mode does not stop running agents, VictoriaMetrics continues collecting data from exporters.
- [PMM-6685](#): In low screen resolutions Services submenu wraps, becomes obscured, and can't be accessed.
- [PMM-6681](#): Not all PMM admin users can download diagnostic logs, only those with Grafana admin rights.
- [PMM-7227](#): Table stats metrics not being collected in instances with millions of tables.
- [PMM-7426](#): `vmagent` continually restarts, blocking comms between `pmm-agent` & `pmm-managed` – Users running multiple services on the same PMM agent in 'push' mode could face this issue when restarting the agent after bulk-adding services.
- [PMM-6636](#): Dashboards: MySQL Replication Summary: 'Binlog Size', 'Binlog Data Written Hourly', 'Node' not being charted when the instance is RDS.
- [PMM-7325](#): Dashboards: MySQL User Details: user labels unreadable with high number (>20) of users (Thanks to Andrei Fedorov for reporting this issue).
- [PMM-7416](#): Dashboards: PostgreSQL Instance Summary: Some panels (e.g. Tuple) not using selected database.
- [PMM-7235](#): Integrated Alerting: Filtered out alerts are shown in the UI as firing.
- [PMM-7324](#): Integrated Alerting: Add Pager Duty Notification Channel: after user pastes copied key Add button is not enabled.
- [PMM-7346](#): Integrated Alerting: It is possible to create Alert Rule with negative duration time.
- [PMM-7366](#): Integrated Alerting: Entities (e.g. templates, channels, rules) are in inconsistent states.
- [PMM-7467](#): Integrated Alerting: < (less-than symbol) wrongly interpreted by Alert templates (as &lt; ; ).
- [PMM-7591](#): Integrated Alerting: User can not receive notifications on email after password update.
- [PMM-7343](#): Security Threat Tool: Check results show previously failed checks after STT re-enabled.
- [PMM-7250](#): DBaaS: Confusing error "Cannot get PSMDB/PXC cluster" appears after removing DB cluster.
- [PMM-7193](#): DBaaS: Number of Nodes can be set as float.
- [PMM-7349](#): DBaaS: Host and Password occasionally disappearing from Connection column.

Last update: 2021-06-25

## 7.15 Percona Monitoring and Management 2.14.0

Date: January 28, 2021

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.15.1 Release Highlights

- Switch to push metrics by default

In PMM 2.12.0, Percona replaced its metrics collection engine (formerly Prometheus) with VictoriaMetrics. Historically, PMM used a *pull* method with Prometheus while VictoriaMetrics can operate in either a *pull* or *push* method. When PMM 2.12.0 was released, Percona kept the default method as *pull*. Now with PMM 2.14.0, Percona is shifting the default to *push* for all newly-added instances. This [blog post](#) describes the two methods and why *push* benefits users. Also, [here is a post](#) by Peter Zaitsev of FAQs relating to the move to VictoriaMetrics and the push model. Documentation on the push method is [here](#).

**Note:** Installing the 2.14.0 or newer PMM server will change the default behavior on 2.12.0 and 2.13.0 clients from “pull” method to “push” for any newly added services. Existing services will remain in whatever mode they were prior to upgrade.

- DBaaS Preview phase 1.0 (Technical Preview)

In 2.13.0 we introduced Percona’s Database as a Service (DBaaS) which enables non-DBAs (software architects, developers, site reliability engineers, etc.) to perform typical DBA tasks to manage an organization’s database environment via user interfaces and automation orchestration. This release contains several enhancements and fixes, many directly from user feedback.

**Note:** This capability is feature-flagged and turned off by default. Users require a variable to be passed to PMM to expose this functionality.

- External services presentation on node summary dashboard

Improvements to the user experience for adding and viewing external services (any data that can be monitored by a Prometheus exporter such as: non-Percona supported databases like Redis, ElasticSearch, Cassandra, etc. or an organization’s external application) on the Node Summary dashboard of PMM.

### 7.15.2 New Features

- [PMM-5765](#): Ability to monitor External Services for situations where PMM Client can’t be installed – Uses a new command `pmm-admin add external-serverless`. (See [pmm-admin](#).) (This is a **Technical Preview** feature)
- [PMM-7015](#): DBaaS Preview: Create DB cluster with randomly-generated password
- [PMM-7007](#): Integrated Alerting: Ability to copy (duplicate) alert rules
- [PMM-7006](#): Integrated Alerting: Ability to delete alert rules
- [PMM-6941](#): Integrated Alerting: Ability to delete alert rule templates

### 7.15.3 Improvements

- [PMM-6985](#): DBaaS: Ability to force unregister Kubernetes cluster
- [PMM-7145](#): ‘Push’ metrics mode is default when adding services and nodes (All agents collecting data from Services and Nodes will now use PUSH model if not specified explicitly. You will still be able to use `--metrics-mode` flag to use Pull metrics if needed. All previously set up agents will keep their existing mode. To change these you need to remove and re-add them.)

- [PMM-7282](#): Integrated Alerting: Ability to create rule without channels and filters
- [PMM-7226](#): Integrated Alerting: Validate parameters during rule creation/update
- [PMM-7082](#): Integrated Alerting: Severity levels are color-coded
- [PMM-7065](#): Integrated Alerting: Show rule details for items in Alert Rules list
- [PMM-7048](#): DBaaS: Simplify Cluster creation by moving Create Cluster button to earlier steps
- [PMM-6993](#): Protect against possible problems with EXPLAIN of stored functions in MySQL – We are fixing possible problems caused by an attempt to analyze queries covered in <https://bugs.mysql.com/bug.php?id=67632>.

#### 7.15.4 Bugs Fixed

- [PMM-7312](#): Error when accessing Metrics data on Dashboards for large installations
- [PMM-7310](#): VictoriaMetrics consuming 100's Gb's of disk in `/tmp/searchResults` in PMM 2.13.0
- [PMM-5137](#): Swagger page redirect isn't working
- [PMM-7144](#): DBaaS: Creating DB cluster with same name (Thanks to Beata Handzelova for reporting this issue)
- [PMM-7323](#): DBaaS: 'Remove DB Cluster from Kubernetes Cluster' removes wrong one
- [PMM-7251](#): Integrated Alerting: Error `Rule with ID "mysql_version" not found` if both Security Threat Tool and Integrated Alerting enabled
- [PMM-7247](#): DBaaS: Disk size is always 0 for Percona XtraDB cluster
- [PMM-7178](#): `pg_stat_monitor` integration is broken with version 0.6.0 of the plugin
- [PMM-7169](#): Old data (from Prometheus) not deleted when Retention period expires
- [PMM-7105](#): Query Analytics: no 'Example' or 'Explain' data for MariaDB
- [PMM-7239](#): Integrated Alerting: Validate Slack channel names in Notification Channels
- [PMM-7213](#): MySQL InnoDB Details dashboard: remove color-coding on 'Data Buffer Pool Fit' element
- [PMM-7167](#): Some panels not visible when using long time intervals (e.g. 30 days)
- [PMM-7133](#): Incorrect descriptions for data links in dashboards
- [PMM-7103](#): VictoriaMetrics build logs not deleted from PMM Server Docker image
- [PMM-6904](#): `pmm-admin annotate` command crashes for non-generic node types
- [PMM-6902](#): No query Examples on PostgreSQL 12 with `pg_stat_monitor`
- [PMM-6838](#): ProxySQL Instance Summary dashboard: Incorrect "Hostgroup Size" formula
- [PMM-6490](#): `rds_exporter` crashes when more than 100 AWS RDS instances added (Thanks to <https://github.com/vlinevych> for fixing this)
- [PMM-6096](#): `pmm-agent` connection checker does not check authentication for MongoDB
- [PMM-7303](#): Disk Details, Nodes Compare dashboards: 'Disk Utilization' description is confusing

---

Last update: 2021-06-25

## 7.16 Percona Monitoring and Management 2.13.0

Date: December 29, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.16.1 Release Highlights

#### 1. Ability to monitor SSL-enabled MongoDB

Allows PMM administrators to set up configured SSL certificate “keys” to authenticate the connection to PMM, specifically for setting up MongoDB. This is a critical security requirement especially in large enterprise infrastructure environments.

#### 2. Technical Previews

##### Caution

We do not recommend the use of technical preview features in enterprise or production environments until the functionality is released as general availability (GA). While in Technical Preview status, these features are not supported by Percona Support SLA, except by Product/Engineering on a best-efforts basis.

##### a. Integrated Alerting MVP

A new feature in PMM to set up parameters and revive alerts about the Services and Nodes monitored by PMM.

Read more [on our blog](#) and in our [documentation](#).

##### b. Node Summary/Nodes Overview dashboards: Show External services on dashboards

Improves the user experience for adding and viewing external services on the Node Summary dashboard of PMM. External services means any data that can be monitored by a Prometheus exporter, for example, non-Percona supported databases like Redis, ElasticSearch, Cassandra, etc. or an organization’s external application.

##### c. DBaaS Preview phase 1.0

We are also releasing the first preview of DBaaS functionality; when combined with a compatible Kubernetes environment and Percona Operators, you can create Percona XtraDB or MongoDB clusters with just a few clicks. (Read more about [configuration](#) and [usage](#).)

### 7.16.2 Improvements

- **PMM-5364:** Ability to monitor SSL-enabled MongoDB by passing certificate parameters in `pmm-admin add` command (Thanks to Hubertus Krogmann for reporting this issue)
- **PMM-7086:** Re-mapped `/prometheus/<end-point>` to `/victoriameetrics/<end-point>` but created aliases for users that still rely on the `/prometheus/<end-point>` in bookmarks and scripts (Thanks to Daniel Guzman Burgos for reporting this issue)
- **PMM-6713:** Node Summary/Nodes Overview dashboards: External exporters can now be added to dashboard and shown as part of grouping of a broader service
- **PMM-7173:** VictoriaMetrics updated to 1.50.2: Includes HTML pages vs JSON output and new functions available for alerting rules ([see all tags](#))

### 7.16.3 Bugs Fixed

- [PMM-7054](#): ProxySQL Instance Summary dashboard: no Node Metrics
  - [PMM-7092](#): PMM Server Docker update from 2.11.1 to 2.12.0 leaves container in unhealthy state (Thanks to Hubertus Krogmann for reporting this issue)
  - [PMM-7208](#): Confusing “Access denied” message for ‘Viewer’ users on many dashboards
  - [PMM-6987](#): No IP address shown in log file of OVF appliance running in headless mode
  - [PMM-7146](#): MongoDB Instance Summary dashboard: `ReplSet` element showing metric name instead of replication set
  - [PMM-6992](#): Administrators can't see user's actual IP address in Grafana profile-Preferences-Sessions
  - [PMM-6865](#): Rendered dashboard images partly obscured by error message
- 

Last update: 2021-06-25

## 7.17 Percona Monitoring and Management 2.12.0

Date: December 1, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.17.1 Release Highlights

- VictoriaMetrics replaces Prometheus and is now the default data source. VictoriaMetrics supports both PUSH (client to server) and PULL metrics collection modes. ([Read more.](#))
- PMM Client can be run as a Docker image.
- The 'Add Instance' page and forms have been redesigned and look much better.

### 7.17.2 New Features

- [PMM-5799](#): PMM Client now available as docker image in addition to RPM, DEB and `.tgz`
- [PMM-6968](#): Integrated Alerting: Basic notification channels actions API Create, Read, Update, Delete
- [PMM-6842](#): VictoriaMetrics: Grafana dashboards to monitor `VictoriaMetricsDB` as replacement for dashboards that used to monitor Prometheus DB
- [PMM-6395](#): Replace Prometheus with VictoriaMetrics in PMM for better performance and additional functionality

### 7.17.3 Improvements

- [PMM-6744](#): Prevent timeout of low resolution metrics in MySQL instances with many tables (~1000's)
- [PMM-6504](#): MySQL Replication Summary: MySQL Replication Delay graph not factoring in value of intentionally set `SQL_Delay` thus inflating time displayed
- [PMM-6820](#): `pmm-admin status --wait` option added to allow for configurable delay in checking status of `pmm-agent`
- [PMM-6710](#): `pmm-admin`: Allow user-specified custom 'group' name when adding external services
- [PMM-6825](#): Allow user to specify 'listen address' to `pmm-agent` otherwise default to 127.0.0.1
- [PMM-6793](#): Improve user experience of 'add remote instance' workflow
- [PMM-6759](#): Enable Kubernetes startup probes to get status of `pmm-agent` using 'GET HTTP' verb
- [PMM-6736](#): MongoDB Instance Summary dashboard: Ensure colors for ReplSet status matches those in MongoDB ReplSet Summary dashboard for better consistency
- [PMM-6730](#): Node Overview/Summary Cleanup: Remove duplicate service type 'DB Service Connections'
- [PMM-6542](#): PMM Add Instance: Redesign page for more intuitive experience when adding various instance types to monitoring
- [PMM-6518](#): Update default data source name from 'Prometheus' to 'Metrics' to ensure graphs are populated correctly after upgrade to VictoriaMetrics
- [PMM-6428](#): Query Analytics dashboard - Ensure user-selected filter selections are always visible even if they don't appear in top 5 results
- [PMM-5020](#): PMM Add Remote Instance: User can specify 'Table Statistics Limit' for MySQL and AWS RDS MySQL to disable table stat metrics which can have an adverse impact on performance with too many tables

### 7.17.4 Bugs Fixed

- [PMM-6811](#): MongoDB Cluster Summary: when secondary optime is newer than primary optime, lag incorrectly shows 136 years
- [PMM-6650](#): Custom queries for MySQL 8 fail on 5.x (on update to `pmm-agent` 2.10) (Thanks to user debug for reporting this issue)
- [PMM-6751](#): PXC/Galera dashboards: Empty service name with MySQL version < 5.6.40
- [PMM-5823](#): PMM Server: Timeout when simultaneously generating and accessing logs via download or API
- [PMM-4547](#): MongoDB dashboard replication lag count incorrect (Thanks to user `vvol` for reporting this issue)
- [PMM-7057](#): MySQL Instances Overview: Many monitored instances (~250+) gives 'too long query' error
- [PMM-6883](#): Query Analytics: 'Reset All' and 'Show Selected' filters behaving incorrectly
- [PMM-6686](#): Query Analytics: Filters panel blank on Microsoft Edge 44.18362.449.0
- [PMM-6007](#): PMM Server virtual appliance's IP address not shown in OVF console
- [PMM-6754](#): Query Analytics: Bad alignment of percentage values in Filters panel
- [PMM-6752](#): Query Analytics: Time interval not preserved when using filter panel dashboard shortcuts
- [PMM-6664](#): Query Analytics: No horizontal scroll bar on Explain tab
- [PMM-6632](#): Node Summary - Virtual Memory Utilization chart: incorrect formulas
- [PMM-6537](#): MySQL InnoDB Details - Logging - Group Commit Batch Size: giving incorrect description
- [PMM-6055](#): PMM Inventory - Services: 'Service Type' column empty when it should be 'External' for external services

### 7.17.5 Known Issues

- [PMM-7092](#): Update docker `pmm-server` 2.11.1 to 2.12.0 results in an unhealthy container.

Workaround: A folder is not created on container upgrade and will need to be created manually for one of the components. Before starting the new pmm-server 2.12.0, execute:

```
docker exec -ti pmm-server mkdir -p /srv/victoriametrics/data
docker exec -ti pmm-server chown -R pmm:pmm /srv/victoriametrics/
docker restart pmm-server
```

---

Last update: 2021-06-25

## 7.18 Percona Monitoring and Management 2.11.1

Date: October 19, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.18.1 Bugs Fixed

- [PMM-6782](#): High CPU usage after update to 2.11.0
- 

Last update: 2021-06-25

## 7.19 Percona Monitoring and Management 2.11.0

Date: October 14, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.19.1 New Features

- [PMM-6567](#): Technical preview of new PostgreSQL extension `pg_stat_monitor`
- [PMM-6515](#): Link added directly to Node/Service page from Query Analytics filters, opens in new window

### 7.19.2 Improvements

- [PMM-6727](#): Grafana plugin updates: `grafana-polystat-panel=1.2.2`, `grafana-piechart-panel=1.6.1`
- [PMM-6625](#): Default sort to “Average - descending” on all dashboards
- [PMM-6609](#): MySQL Instances Compare & Summary dashboards: Changed metric in ‘MySQL Internal Memory Overview’
- [PMM-6598](#): Dashboard image sharing (Share Panel): Improved wording with link to configuration instructions
- [PMM-6557](#): Update Prometheus to 2.21.0
- [PMM-6554](#): MySQL InnoDB Details dashboard: Add “sync flushing” to “InnoDB Flushing by Type”

### 7.19.3 Bugs Fixed

- [PMM-4547](#): MongoDB dashboard replication lag count incorrect (Thanks to user `vvol` for reporting this issue)
- [PMM-6639](#): Integrated update does not detect all container types
- [PMM-6765](#): Tables information tab reports ‘table not found’ with new PostgreSQL extension `pg_stat_monitor`
- [PMM-6764](#): Query Analytics: cannot filter items that are hidden - must use “Show all”
- [PMM-6742](#): Upgrade via PMM UI stalls (on `yum update pmm-update`)
- [PMM-6689](#): No PostgreSQL queries or metrics in Query Analytics with PostgreSQL 13 (`postgresql_pgstatements_agent` in Waiting status)
- [PMM-6738](#): PostgreSQL examples shown despite `--disable-queryexamples` option
- [PMM-6535](#): Unable to open ‘Explore’ in new window from Grafana menu
- [PMM-6532](#): Click-through URLs lose time ranges when redirecting to other dashboards
- [PMM-6531](#): Counter-intuitive coloring of element “Update Stats when Metadata Queried”
- [PMM-6645](#): Clean up unnecessary errors in logs (`vertamedia-clickhouse-datasource` plugin)
- [PMM-6547](#): Hexagonal graph tooltip text overflows bounding box

---

Last update: 2021-06-25

## 7.20 Percona Monitoring and Management 2.10.1

Date: September 22, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.20.1 Bugs Fixed

- [PMM-6643](#): New MongoDB exporter has higher CPU usage compared with old
- 

Last update: 2021-06-25

## 7.21 Percona Monitoring and Management 2.10.0

Date: September 15, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.21.1 New Features

- [PMM-2045](#): New dashboard: MySQL Group Replication Summary
- [PMM-5738](#): Enhanced exporter: replaced original `mongodb-exporter` with a completely rewritten one with improved functionality
- [PMM-5126](#): Query Analytics Dashboard: Search by query substring or dimension (Thanks to user `debug` for reporting this issue)
- [PMM-6360](#): Grafana Upgrade to 7.1.3
- [PMM-6355](#): Upgrade Prometheus to 2.19.3
- [PMM-6597](#): Documentation: Updated Image rendering instructions for PMM
- [PMM-6568](#): Reusable user interface component: Pop-up dialog. Allows for more consistent interfaces across PMM
- [PMM-6375](#), [PMM-6373](#), [PMM-6372](#): Sign in, Sign up and Sign out UI for Percona Account inside PMM Server
- [PMM-6328](#): Query Analytics Dashboard: Mouse-over crosshair shows value on sparklines
- [PMM-3831](#): Node Summary Dashboard: Add `pt-summary` output to dashboard to provide details on system status and configuration

### 7.21.2 Improvements

- [PMM-6647](#): MongoDB dashboards: RocksDB Details removed, MMAPv1 & Cluster Summary changed
- [PMM-6536](#): Query Analytics Dashboard: Improved filter/time search message when no results
- [PMM-6467](#): PMM Settings: User-friendly error message
- [PMM-5947](#): Bind services to internal address for containers

### 7.21.3 Bugs Fixed

- [PMM-6336](#): Suppress sensitive data: honor `pmm-admin` flag `--disable-queryexamples` when used in conjunction with `--query-source=perfschema`
- [PMM-6244](#): MySQL InnoDB Details Dashboard: Inverted color scheme on “BP Write Buffering” panel
- [PMM-6294](#): Query Analytics Dashboard doesn't resize well for some screen resolutions (Thanks to user `debug` for reporting this issue)
- [PMM-5701](#): Home Dashboard: Incorrect metric for `DB uptime` (Thanks to user `hubi_oediv` for reporting this issue)
- [PMM-6427](#): Query Analytics dashboard: Examples broken when switching from MongoDB to MySQL query
- [PMM-5684](#): Use actual data from `INFORMATION_SCHEMA` vs relying on cached data (which can be 24 hrs old by default)
- [PMM-6500](#): PMM Database Checks: Unwanted high-contrast styling
- [PMM-6440](#): MongoDB ReplSet Summary Dashboard: Primary shows more lag than replicas

- [PMM-6436](#): Query Analytics Dashboard: Styles updated to conform with upgrade to Grafana 7.x
- [PMM-6415](#): Node Summary Dashboard: Redirection to database's Instance Summary dashboard omits Service Name
- [PMM-6324](#): Query Analytics Dashboard: Showing stale data while fetching updated data for query details section
- [PMM-6316](#): Query Analytics Dashboard: Inconsistent scrollbar styles
- [PMM-6276](#): PMM Inventory: Long lists unclear; poor contrast & column headings scroll out of view
- [PMM-6529](#): Query Analytics filter input margin disappears after scrolling

#### 7.21.4 Known Issues

- [PMM-6643](#): High CPU usage for new MongoDB exporter (fixed in Percona Monitoring and Management 2.10.1)
- 

Last update: 2021-06-25

## 7.22 Percona Monitoring and Management 2.9.1

Date: August 4, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.22.1 Improvements

- [PMM-6230](#): Custom dashboards set as Home remain so after update
- [PMM-6300](#): Query Analytics Dashboard: Column sorting arrows made easier to use (Thanks to user debug for reporting this issue)
- [PMM-6208](#): Security Threat Tool: Temporarily silence viewed but un-actioned alerts
- [PMM-6315](#): Query Analytics Dashboard: Improved metrics names and descriptions
- [PMM-6274](#): MySQL User Details Dashboard: View selected user's queries in Query Analytics Dashboard
- [PMM-6266](#): Query Analytics Dashboard: Pagination device menu lists 25, 50 or 100 items per page
- [PMM-6262](#): PostgreSQL Instance Summary Dashboard: Descriptions for all 'Temp Files' views
- [PMM-6253](#): Query Analytics Dashboard: Improved SQL formatting in Examples panel
- [PMM-6211](#): Query Analytics Dashboard: Loading activity spinner added to Example, Explain and Tables tabs
- [PMM-6162](#): Consistent sort order in dashboard drop-down filter lists
- [PMM-5132](#): Better message when filter search returns nothing

### 7.22.2 Bugs Fixed

- [PMM-5783](#): Bulk failure of SHOW ALL SLAVES STATUS scraping on PS/MySQL distributions triggers errors
- [PMM-6294](#): Query Analytics Dashboard doesn't resize well for some screen resolutions (Thanks to user debug for reporting this issue)
- [PMM-6420](#): Wrong version in successful update pop-up window
- [PMM-6319](#): Query Analytics Dashboard: Query scrolls out of view when selected
- [PMM-6302](#): Query Analytics Dashboard: Unnecessary EXPLAIN requests
- [PMM-6256](#): Query Analytics Dashboard: `InvalidNamespace` EXPLAIN error with some MongoDB queries
- [PMM-6329](#): Query Analytics Dashboard: Unclear origin of sparkline tool-tip on mouse-over
- [PMM-6259](#): Query Analytics Dashboard: Slow appearance of query time distribution graph for some queries
- [PMM-6189](#): Disk Details Dashboard: Disk IO Size chart larger by factor of 512
- [PMM-6269](#): Query Analytics Dashboard: Metrics drop-down list obscured when opened
- [PMM-6247](#): Query Analytics Dashboard: Overview table not resizing on window size change
- [PMM-6227](#): Home Dashboard redirection to Node Summary Dashboard not working

---

Last update: 2021-06-25

## 7.23 Percona Monitoring and Management 2.9.0

Date: July 14, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.23.1 Release Highlights

This release brings a major rework of the Query Analytics (QAN) component, completing the migration from Angular to React, and adding new UI functionality and features.

For details, see:

- [PMM-5125](#): Implement new version of QAN
- [PMM-5516](#): QAN migration to React and new UI implementation

You can read more in the accompanying blog post ([here](#)).

### 7.23.2 New Features

- [PMM-6124](#): New dashboards: MongoDB Replica Set Summary and MongoDB Cluster Summary
- [PMM-1027](#): New dashboard: MySQL User Details ( `INFORMATION_SCHEMA.CLIENT_STATISTICS` )
- [PMM-5604](#): User interface for MongoDB EXPLAIN
- [PMM-5563](#): Per-Service and per-Node Annotations (This completes the work on improvements to the Annotation functionality.)

### 7.23.3 Improvements

- [PMM-6114](#): Sort Agents, Nodes, and Services alphabetically by name in Inventory page (Thanks to user `debug` for reporting this issue)
- [PMM-6147](#): Update Grafana plugins to latest versions

### 7.23.4 Bugs Fixed

- [PMM-5800](#): QAN explain and tables tabs not working after removing MySQL metrics agent
- [PMM-5812](#): Prometheus relabeling broken ( `relabel_configs` un-marshals errors) (Thanks to user `b4bufrik` for reporting this issue)
- [PMM-6184](#): MongoDB Instances Compare dashboard shows MySQL metric
- [PMM-5941](#): Stacked Incoming/Outgoing Network Traffic graphs in MySQL Instances Overview dashboard prevents comparison
- [PMM-6194](#): Missing UID for Advanced Data Exploration dashboard
- [PMM-6191](#): Incorrect computation for Prometheus Process CPU Usage panel values in Prometheus dashboard
- [PMM-6175](#): Node Overview dashboard shows unit for unit-less value 'Top I/O Load'

---

Last update: 2021-06-25

## 7.24 Percona Monitoring and Management 2.8.0

Date: June 25, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.24.1 Improvements

- [PMM-544](#): Agents, Services and Nodes can now be removed via the 'PMM Inventory' page
- [PMM-5706](#): User-installed Grafana plugins unaffected by PMM upgrade

### 7.24.2 Bugs Fixed

- [PMM-6153](#): PMM 2.7.0 inoperable when no Internet connectivity
- [PMM-5365](#): Client fails to send non-UTF-8 query analytics content to server (Thanks to user `romulus` for reporting this issue)
- [PMM-5920](#): Incorrect metric used in formula for "Top Users by Rows Fetched/Read" graph
- [PMM-6084](#): Annotations not showing consistently on dashboards
- [PMM-6011](#): No data in MongoDB Cluster summary, RocksDB & MMAPv1 details
- [PMM-5987](#): Incorrect total value for virtual memory utilization

---

Last update: 2021-06-25

## 7.25 Percona Monitoring and Management 2.7.0

Date: June 9, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

In this release, we have updated Grafana to version 6.7.4 to fix [CVE-2020-13379](#). We recommend updating to the latest version of PMM as soon as possible.

### 7.25.1 New Features

- [PMM-5257](#), [PMM-5256](#), & [PMM-5243](#): `pmm-admin` socket option (`--socket`) to specify UNIX socket path for connecting to MongoDB, PostgreSQL, and ProxySQL instances

### 7.25.2 Improvements

- [PMM-2244](#): `pmm-admin status` command output shows both `pmm-admin` and `pmm-agent` versions
- [PMM-5968](#): Disallow PMM Server node or agent removal via API
- [PMM-5946](#): MySQL Table Details dashboard filter on Service Name prevents display of services without data
- [PMM-5926](#): Expose PMM agent version in `pmm-admin status` command
- [PMM-5891](#): PMM Home page now includes News panel
- [PMM-5906](#): Independent update of PMM components deactivated

### 7.25.3 Bugs Fixed

- [PMM-6004](#): MySQL exporter reporting wrong values for cluster status (`wsrep_cluster_status`)
- [PMM-4547](#): MongoDB dashboard replication lag count incorrect
- [PMM-5524](#): Prometheus alerting rule changes needs docker restart to activate
- [PMM-5949](#): Unwanted filters applied when moving from QAN to Add Instance page
- [PMM-5870](#): MySQL Table Details dashboard not showing separate service names for tables
- [PMM-5839](#): PostgreSQL metrics disparity between query time and block read/write time
- [PMM-5348](#): Inventory page has inaccessible tabs that need reload to access
- [PMM-5348](#): Incorrect access control vulnerability fix (CVE-2020-13379) by upgrading Grafana to 6.7.4

---

Last update: 2021-06-25

## 7.26 Percona Monitoring and Management 2.6.1

Date: May 18, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.26.1 Improvements

- [PMM-5936](#): Improved Summary dashboard for Security Threat Tool 'Failed Checks'
- [PMM-5937](#): Improved Details dashboard for Security Threat Tool 'Failed Database Checks'

### 7.26.2 Bugs Fixed

- [PMM-5924](#): Alertmanager not running after PMM Server upgrade via Docker
- [PMM-5915](#): `supervisord` not restarting after restart of PMM Server virtual appliances (OVF/AMI)
- [PMM-5945](#): 'Updates' dashboard not showing available updates
- [PMM-5870](#): MySQL Table Details dashboard not showing separate service names for tables

---

Last update: 2021-06-25

## 7.27 Percona Monitoring and Management 2.6.0

Date: May 11, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.27.1 New Features

- [PMM-5728](#): Technical preview of External Services monitoring feature. A new command provides integration with hundreds of third-party systems (<https://prometheus.io/docs/instrumenting/exporters/>) via the Prometheus protocol so that you can monitor external services on a node where PMM agent is installed.
- [PMM-5822](#): PMM now includes a Security Threat Tool to help users avoid the most common database security issues. Read more [here](#).
- [PMM-5559](#): Global annotations can now be set with the `pmm-admin annotate` command.
- [PMM-4931](#): PMM now checks Docker environment variables and warns about invalid ones.

### 7.27.2 Improvements

- [PMM-1962](#): The PMM Server API (via `/v1/readyz`) now also returns Grafana status information in addition to that for Prometheus.
- [PMM-5854](#): The Service Details dashboards were cleaned up and some unused selectors were removed.
- [PMM-5775](#): It is now clearer which nodes are Primary and which are Secondary on MongoDB Instance dashboards.
- [PMM-5549](#): PMM's Grafana component is now the latest, 6.7.3.
- [PMM-5393](#): There's a new 'Node Summary' row in the services Summary and Details dashboards summarizing the system update, load average, RAM and memory.
- [PMM-4778](#): `mongodb_exporter` is now the latest version, 0.11.0.
- [PMM-5734](#): Temporary files activity and utilization charts (`rate` & `irate`) were added to the PostgreSQL Instance overview.
- [PMM-5695](#): The error message explains better when using the `--socket` option incorrectly.

### 7.27.3 Bugs Fixed

- [PMM-4829](#): The MongoDB Exporter wasn't able to collect metrics from hidden nodes without either the latest driver or using the `connect-direct` parameter.
- [PMM-5056](#): The average values for Query time in the Details and Profile sections were different.
- [PMM-2717](#): Updating MongoDB Exporter resolves an error (`Failed to execute find query on 'config.locks': not found.`) when used with `shardedCluster` 3.6.4.
- [PMM-4541](#): MongoDB exporter metrics collection was including system collections from `collStats` and `indexStats`, causing "log bloat".
- [PMM-5913](#): Only totals were shown in QAN when filtering on `Cluster=MongoDB`.
- [PMM-5903](#): When applying a filter the QAN Overview was being refreshed twice.
- [PMM-5821](#): The Compare button was missing from HA Dashboard main menus.
- [PMM-5687](#): Cumulative charts for Disk Details were not showing any data if metrics were returning `Nan` results.

- [PMM-5663](#): The ‘version’ value was not being refreshed in various MySQL dashboards.
- [PMM-5643](#): Advanced Data Exploration charts were showing ‘N/A’ for Metric Resolution and ‘No data to show’ in the Metric Data Table.
- [PMM-4756](#): Dashboards were not showing services with empty environments.
- [PMM-4562](#): MongoDB and MySQL registered instances with empty cluster labels (`--environment=<label>`) were not visible in the dashboard despite being added instances.
- [PMM-4906](#): The MongoDB exporter for MongoDB 4.0 and above was causing a “log bloat” condition.

Help us improve our software quality by reporting any bugs you encounter using [our bug tracking system](#).

---

Last update: 2021-07-02

## 7.28 Percona Monitoring and Management 2.5.0

Date: April 14, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.28.1 New Features

- [PMM-5042](#) and [PMM-5272](#): PMM can now connect to MySQL instances by specifying a UNIX socket. This can be done with a new `--socket` option of the `pmm-admin add mysql` command. (Note: Updates to both PMM Client and PMM Server were done to allow UNIX socket connections.)
- [PMM-4145](#): Amazon RDS instance metrics can now be independently enabled/disabled for Basic and/or Enhanced metrics.

### 7.28.2 Improvements

- [PMM-5581](#): PMM Server Grafana plugins can now be updated on the command line with the `grafana-cli` command-line utility.
- [PMM-5536](#): Three Grafana plugins were updated to the latest versions: `vertamedia-clickhouse-datasource` to 1.9.5, `grafana-polystat-panel` to 1.1.0, and `grafana-piechart-panel` to 1.4.0.
- [PMM-4252](#): The resolution of the PMM Server `favicon` image has been improved.

### 7.28.3 Bugs Fixed

- [PMM-5547](#): PMM dashboards were failing when presenting data from more than 100 monitored instances (error message `proxy error: context canceled`).
- [PMM-5624](#): Empty charts were being shown in some Node Temperature dashboards.
- [PMM-5637](#): The Data retention value in Settings was incorrectly showing the value as minutes instead of days.
- [PMM-5613](#): Sorting data by Query Time was not working properly in Query Analytics.
- [PMM-5554](#): Totals in charts were inconsistently plotted with different colors across charts.
- [PMM-4919](#): The force option (`--force`) in `pmm-admin config` was not always working.
- [PMM-5351](#): The documentation on MongoDB user privileges has been corrected.

Help us improve our software quality by reporting any bugs you encounter using [our bug tracking system](#).

---

Last update: 2021-06-25

## 7.29 Percona Monitoring and Management 2.4.0

Date: March 18, 2020

Installation: [Installing Percona Monitoring and Management](#)

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

### 7.29.1 New Features

- [PMM-3387](#): Prometheus custom configuration is now supported by PMM Server. The feature is targeted at experienced users and is done by [adding the base configuration file into the PMM Server container](#) to be parsed and included into the managed Prometheus configuration.
- [PMM-5186](#): Including `--pprof` option in the `pmm-admin summary` command adds `pprof` debug profiles to the diagnostics data archive
- [PMM-5102](#): The new “Node Details” dashboard now displays data from the hardware monitoring sensors in `hwmon`. The [new dashboard](#) is based on the `hwmon` collector data from the `node_exporter`. Please note that data may be unavailable for some nodes because of the configuration or virtualization parameters.

### 7.29.2 Improvements

- [PMM-4915](#): The Query Analytics dashboard now shows Time Metrics in the Profile Section as “AVG per query” instead of “AVG per second”
- [PMM-5470](#): ClickHouse query optimized for Query Analytics to improve its speed and reduce the load on the back-end
- [PMM-5448](#): The default high and medium metrics resolutions were changed to 1-5-30 and 5-10-60 sec. To reduce the effect of this change on existing installations, systems having the “old” high resolution chosen on the PMM Settings page (5-5-60 sec.) will be automatically re-configured to the medium one during an upgrade. If the resolution was changed to some custom values via API, it will not be affected
- [PMM-5531](#): A health check indicator was implemented for the PMM Server Docker image. It is based on the Docker `HEALTHCHECK`. This feature can be used as follows:

```
docker inspect -f {{.State.Health.Status}}
until [ "`docker inspect -f {{.State.Health.Status}} pmm-server`" == "healthy" ]; do
sleep 1; done
```

- [PMM-5489](#): The “Total” line in all charts is now drawn with the same red color for better consistency
- [PMM-5461](#): Memory graphs on the node-related dashboards were adjusted to have fixed colors that are more distinguishable from each other
- [PMM-5329](#): Prometheus in PMM Server was updated to version 2.16.0. This update has brought several improvements. Among them are significantly reduced memory footprint of the loaded TSDB blocks, lower memory footprint for the compaction process (caused by the more balanced choice of what to buffer during compaction), and improved query performance for the queries that only touch the most recent 2 hours of data.
- [PMM-5210](#): Data Retention is now specified in days instead of seconds on the PMM Settings page. Please note this is a UI-only change, so the actual data retention precision is not changed
- [PMM-5182](#): The `logs.zip` archive available on the PMM Settings page now includes additional self-monitoring information in a separate `client` subfolder. This subfolder contains information collected on the PMM Server and is equivalent to the one collected on a node by the `pmm-admin summary` command.
- [PMM-5112](#): The Inventory API List requests now can be filtered by the Node/Service/Agent type

### 7.29.3 Bugs Fixed

- [PMM-5178](#): Query Detail Section of the Query Analytics dashboard didn't show tables definitions and indexes for the internal PostgreSQL database
- [PMM-5465](#): MySQL Instance related dashboards had row names not always matching the actual contents. To fix this, elements were re-ordered and additional rows were added for better matching of the row name and the corresponding elements
- [PMM-5455](#): Dashboards from the Insight menu were fixed to work correctly when the low resolution is set on the PMM Settings page
- [PMM-5446](#): A number of the Compare Dashboards were fixed to work correctly when the low resolution is set on the PMM Settings page
- [PMM-5430](#): MySQL Exporter section on the Prometheus Exporter Status dashboard now collapsed by default to be consistent with other database-related sections
- [PMM-5445](#), [PMM-5439](#), [PMM-5427](#), [PMM-5426](#), [PMM-5419](#): Labels change (which occurs e.g. when the metrics resolution is changed on the PMM Settings page) was breaking dashboards
- [PMM-5347](#): Selecting queries on the Query Analytics dashboard was generating errors in the browser console
- [PMM-5305](#): Some applied filters on the Query Analytics dashboard were not preserved after changing the time range
- [PMM-5267](#): The Refresh button was not working on the Query Analytics dashboard
- [PMM-5003](#): pmm-admin list and status use different JSON naming for the same data
- [PMM-5526](#): A typo was fixed in the Replication Dashboard description tooltip

Help us improve our software quality by reporting any bugs you encounter using [our bug tracking system](#).

---

Last update: 2021-06-25

## 7.30 Percona Monitoring and Management 2.3.0

Date: February 19, 2020

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

For PMM install instructions, see [Installing PMM Server](#) and [Installing PMM Client](#).

 **Caution**

PMM 2 is designed to be used as a new installation — please don't try to upgrade your existing PMM 1 environment.

### 7.30.1 Improvements and new features

- [PMM-5064](#) and [PMM-5065](#): Starting from this release, users will be able to integrate PMM with an external Alertmanager by specifying the Alertmanager URL and the Alert Rules to be executed inside the PMM server (**This feature is for advanced users only at this point**)
- [PMM-4954](#): Query Analytics dashboard now shows units both in the list of queries in a summary table and in the Details section to ease understanding of the presented data
- [PMM-5179](#): Relations between metrics are now specified in the Query Analytics Details section
- [PMM-5115](#): The CPU frequency and temperature graphs were added to the CPU Utilization dashboard
- [PMM-5394](#): A special treatment for the node-related dashboards was implemented for the situations when the data resolution change causes new metrics to be generated for existing nodes and services, to make graphs show continuous lines of the same colors

### 7.30.2 Fixed bugs

- [PMM-4620](#): The high CPU usage by the pmm-agent process related to MongoDB Query Analytics was fixed
- [PMM-5377](#): `singlestats` showing percentage had sparklines scaled vertically along with the graph swing, which made it difficult to visually notice the difference between neighboring `singlestats`.
- [PMM-5204](#): Changing resolution on the PMM settings page was breaking some `singlestats` on the Home and MySQL Overview dashboards
- [PMM-5251](#): Vertical scroll bars on the graph elements were not allowed to do a full scroll, making last rows of the legend unavailable for some graphs
- [PMM-5410](#): The “Available Downtime before SST Required” chart on the PXC/Galera Node Summary dashboard was not showing data because it was unable to use metrics available with different scraping intervals

---

Last update: 2021-06-25

## 7.31 Percona Monitoring and Management 2.2.2

Date: February 4, 2020

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

For PMM install instructions, see [Installing PMM Server](#) and [Installing PMM Client](#).

 **Caution**

PMM 2 is designed to be used as a new installation — please don't try to upgrade your existing PMM 1 environment.

### 7.31.1 Improvements and new features

- [PMM-5321](#): The optimization of the Query Analytics parser code for PostgreSQL queries allowed us to reduce the memory resources consumption by 1-5%, and the parsing time of an individual query by 30 to 40%
- [PMM-5184](#): The `pmm-admin summary` command have gained a new `--skip-server` flag which makes it operating in a local-only mode, creating summary file without contacting the PMM Server

### 7.31.2 Fixed bugs

- [PMM-5340](#): The Scraping Time Drift graph on the Prometheus dashboard was showing wrong values because the actual metrics resolution wasn't taken into account
- [PMM-5060](#): Query Analytics Dashboard did not show the row with the last query of the first page, if the number of queries to display was 11

---

Last update: 2021-06-25

## 7.32 Percona Monitoring and Management 2.2.1

Date: January 23, 2020

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance.

For PMM install instructions, see [Installing PMM Server](#) and [Installing PMM Client](#).

 **Caution**

PMM 2 is designed to be used as a new installation — please don't try to upgrade your existing PMM 1 environment.

PMM Server version 2.2.0 suffered an unauthenticated denial of service vulnerability (CVE-2020-7920). Any other PMM versions do not carry the same code logic, and are thus unaffected by this issue. **Users who have already deployed PMM Server 2.2.0 are advised to upgrade to version 2.2.1 which resolves this issue.**

### 7.32.1 Improvements and new features

- [PMM-5229](#): The new RDS Exporter section added to the Prometheus Exporter Status dashboard shows singlestats and charts related to the `rds_exporter`
- [PMM-5228](#) and [PMM-5238](#): The Prometheus dashboard and the Exporters Overview dashboard were updated to include the `rds_exporter` metrics in their charts, allowing better understanding of the impacts of monitoring RDS instances
- [PMM-4830](#): The consistency of the applied filters between the Query Analytics and the Overview dashboards was implemented, and now filters selected in QAN will continue to be active after the switch to any of the Overview dashboards available in the Services menu
- [PMM-5235](#): The DB `uptime` `singlestats` in node rows on the Home dashboard were changed to show minimal values instead of average ones to be consistent with the top row
- [PMM-5127](#): The “Search by” bar on the Query Analytics dashboard was renamed to “Filter by” to make its purpose more clear
- [PMM-5131](#): The Filter panel on the Query Analytics dashboard now shows the total number of available Labels within the “See all” link, which appears if the Filter panel section shows only top 5 of its Labels

### 7.32.2 Fixed bugs

- [PMM-5232](#): The `pmm-managed` component of the PMM Server 2.2.0 is vulnerable to DoS attacks, that could be carried out by anyone who knows the PMM Server IP address (CVE-2020-7920). Versions other than 2.2.0 are not affected.
- [PMM-5226](#): The `handlebars` package was updated to version 4.5.3 because of the Prototype Pollution vulnerability in it (CVE-2019-19919). Please note PMM versions were not affected by this vulnerability, as handlebars package is used as a build dependency only.
- [PMM-5206](#): Switching to the Settings dashboard was breaking the visual style of some elements on the Home dashboard
- [PMM-5139](#): The breadcrumb panel, which shows all dashboards visited within one session starting from the root, was unable to fully show breadcrumb longer than one line
- [PMM-5212](#): The explanatory text was added to the Download PMM Server Logs button in the Diagnostic section of the PMM Settings dashboard, and a link to it was added to the Prometheus dashboard which was the previous place to download logs

- [PMM-5215](#): The unneeded `mariadb-libs` package was removed from the PMM Server 2.2.0 OVF image, resulting in both faster updating with the `yum update` command and avoiding dependency conflict messages in the update logs
  - [PMM-5216](#): PMM Server Upgrade to 2.2.0 was showing Grafana Update Error page with the Refresh button which had to be clicked to start using the updated version
  - [PMM-5211](#): The “Where do I get the security credentials for my Amazon RDS DB instance” link in the Add AWS RDS MySQL or Aurora MySQL instance dialog was not targeted at the appropriate instruction
  - [PMM-5217](#): PMM 2.x OVF Image memory size was increased from 1 Gb to 4 Gb with the additional 1 Gb swap space because the previous amount was hardly housing the PMM Server, and it wasn’t enough in some cases like performing an upgrade
  - [PMM-5271](#): LVM logical volumes were wrongly resized on AWS deployment, resulting in “no space left on device” errors
  - [PMM-5295](#): InnoDB Transaction Rollback Rate values on the MySQL InnoDB Details dashboard were calculated incorrectly
  - [PMM-5270](#): PXC/Galera Cluster Summary dashboard was showing empty Cluster drop-down list, making it impossible to choose the cluster name
  - [PMM-4769](#): The wrongly named “Timeout value used for retransmitting” `singlestat` on the Network Details dashboard was renamed to “The algorithm used to determine the timeout value” and updated to show the algorithm name instead of a digital code
  - [PMM-5260](#): Extensive resource consumption by `pmm-agent` took place in case of Query Analytics for PostgreSQL; it was fixed by a number of optimizations in the code, resulting in about 4 times smaller memory usage
  - [PMM-5261](#): CPU usage charts on all dashboards which contain them have undergone colors update to make `softIRQ` and Steal curves better differentiated
  - [PMM-5244](#): High memory consumption in the PMM Server with a large number of agents sending data simultaneously was fixed by improving bulk data insertion to the ClickHouse database
- 

Last update: 2021-06-25

## 7.33 Percona Monitoring and Management 2.2.0

Date: December 24, 2019

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance. You can run *PMM* in your own environment for maximum security and reliability. It provides thorough time-based analysis for MySQL, MongoDB, and PostgreSQL servers to ensure that your data works as efficiently as possible.

Main improvements in this release are:

- Alternative installation methods available for PMM 1.x are re-implemented for PMM 2: now PMM Server can be installed as a virtual appliance, or run using AWS Marketplace
- AWS RDS and remote instances monitoring re-added in this release include AWS RDS MySQL / Aurora MySQL instances, and remote PostgreSQL, MySQL, MongoDB, and ProxySQL ones
- The new Settings dashboard allows configuring PMM Server via the graphical interface

For PMM install instructions, see [Installing PMM Server](#) and [Installing PMM Client](#).

### Caution

PMM 2 is designed to be used as a new installation — please don't try to upgrade your existing PMM 1 environment.

### 7.33.1 Improvements and new features

- [PMM-4575](#): The new PMM Settings dashboard allows users to configure various PMM Server options: setting metrics resolution and data retention, enabling or disabling send usage data statistics back to Percona and checking for updates; this dashboard is now the proper place to upload your public key for the SSH login and to download PMM Server logs for diagnostics
- [PMM-4907](#) and [PMM-4767](#): The user's AMI Instance ID is now used to setup running PMM Server using AWS Marketplace as an additional verification on the user, based on the Amazon Marketplace rules
- [PMM-4950](#) and [PMM-3094](#): Alternative [AWS partitions](#) are now supported when adding an AWS RDS MySQL or Aurora MySQL Instance to PMM
- [PMM-4976](#): Home dashboard clean-up: "Systems under monitoring" and "Network IO" `singlestats` were refined to be based on the `host` variable; also avoiding using color as an indicator of state; "All" row elements were relinked to the "Nodes Overview" dashboard with regards to the selected host.
- [PMM-4800](#): The `pmm-admin add mysql` command has been modified to make help text more descriptive: now when you enable `tablestats` you will get more detail on if they're enabled for your environment and where you stand with respect to the auto-disable limit
- [PMM-4969](#): Update Grafana to version 6.5.1
- [PMM-5053](#): A tooltip was added to the Head Block graph on the Prometheus dashboard
- [PMM-5068](#): Drill-down links were added to the Node Summary dashboard graphs
- [PMM-5050](#): Drill-down links were added to the graphs on all Services Compare dashboards
- [PMM-5037](#): Drill-down links were added to all graphs on the Services Overview dashboards
- [PMM-4988](#): Filtering in Query Analytics have undergone improvements to make group selection more intuitive: Labels unavailable under the current selection are shown as gray-disabled, and the percentage values are dynamically recalculated to reflect Labels available within the currently applied filters

- [PMM-4966](#): All passwords are now substituted with asterisk signs in the exporter logs for security reasons when not in debug mode
- [PMM-527](#): `node_exporter` is now providing hardware monitoring information such as CPU temperatures and fan statuses; while this information is being collected by PMM Server, it will not be shown until a dedicated dashboard is added in a future release
- [PMM-3198](#): Instead of showing All graphs for all services by default, MySQL Command/Handler Counters Compare dashboard now shows the predefined set of ten most informative ones, to reduce load on PMM Server at its first open

### 7.33.2 Fixed bugs

- [PMM-4978](#): The “Top MySQL Questions” `singlestat` on the MySQL Instances Overview dashboard was changed to show ops instead of percentage
- [PMM-4917](#): The “Systems under monitoring” and “Monitored DB Instances” `singlestats` on the Home dashboard now have a sparkline to make situation more clear with recently shut down nodes/instances
- [PMM-4979](#): Set decimal precision `2` for all the elements, including charts and `singlestats`, on all dashboards
- [PMM-4980](#): Fix “Load Average” `singlestat` on the Node Summary dashboard to show decimal value instead of percent
- [PMM-4981](#): Disable automatic color gradient in filled graphs on all dashboards
- [PMM-4941](#): Some charts were incorrectly showing empty fragments with high time resolution turned on
- [PMM-5022](#): Fix outdated drill-down links on the Prometheus Exporters Overview and Nodes Overview dashboards
- [PMM-5023](#): Make the All instances `uptime singlestat` on the Home dashboard to show `Min` values instead of `Avg`
- [PMM-5029](#): Option to upload dashboard snapshot to Percona was disappearing after upgrade to 2.1.x
- [PMM-4946](#): Rename `singlestats` on the Home dashboard for better clarity: “Systems under monitoring” to “Nodes under monitoring” and “Monitored DB Instances” to “Monitored DB Services”, and make the last one to count remote DB instances also
- [PMM-5015](#): Fix format of Disk Page Buffers `singlestat` on the Compare dashboard for PostgreSQL to have two digits precision for the consistency with other `singlestats`
- [PMM-5014](#): LVM logical volumes were wrongly sized on a new AWS deployment, resulting in “no space left on device” errors.
- [PMM-4804](#): Incorrect parameters validation required both `service-name` and `service-id` parameters of the `pmm-admin remove` command to be presented, while the command itself demanded only one of them to identify the service.
- [PMM-3298](#): Panic errors were present in the `rds_exporter` log after adding an RDS instance from the second AWS account
- [PMM-5089](#): The `serialize-javascript` package was updated to version 2.1.1 because of the possibility of regular expressions cross-site scripting vulnerability in it (CVE-2019-16769). Please note PMM versions were not affected by this vulnerability, as `serialize-javascript` package is used as a build dependency only.
- [PMM-5149](#): Disk Space `singlestat` was unable to show data for RDS instances because of not taking into account sources with unknown file system type

---

Last update: 2021-06-25

## 7.34 Percona Monitoring and Management 2.1.0

Date: November 11, 2019

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance. You can run PMM in your own environment for maximum security and reliability. It provides thorough time-based analysis for MySQL, MongoDB, and PostgreSQL servers to ensure that your data works as efficiently as possible.

For install instructions, see [Installing Percona Monitoring and Management](#).

### Caution

PMM 2 is designed to be used as a new installation — please don't try to upgrade your existing PMM 1 environment.

### 7.34.1 Improvements and new features

- [PMM-4063](#): Update QAN filter panel to show only labels available for selection under currently applied filters
- [PMM-815](#): Latency Detail graph added to the MongoDB Instance Summary dashboard
- [PMM-4768](#): Disable heavy-load collectors automatically when there are too many tables
- [PMM-4821](#): Use color gradient in filled graphs on all dashboards
- [PMM-4733](#): Add more log and configuration files to the downloadable `logs.zip` archive
- [PMM-4672](#): Use integer percentage values in QAN filter panel
- [PMM-4857](#): Update tooltips for all MongoDB dashboards
- [PMM-4616](#): Rename column in the Query Details section in QAN from Total to Sum
- [PMM-4770](#): Use Go 1.12.10
- [PMM-4780](#): Update Grafana to version 6.4.1
- [PMM-4918](#): Update Grafana plugins to newer versions, including the `clickhouse-datasource` plugin

### 7.34.2 Fixed bugs

- [PMM-4935](#): Wrong instance name displayed on the MySQL Instance Summary dashboard due to the incorrect string crop
- [PMM-4916](#): Wrong values are shown when changing the time range for the Node Summary Dashboard in case of remote instances
- [PMM-4895](#) and [PMM-4814](#): The update process reports completion before it is actually done and therefore some dashboards, etc. may not be updated
- [PMM-4876](#): PMM Server access credentials are shown by the `pmm-admin status` command instead of hiding them for security reasons
- [PMM-4875](#): PostgreSQL error log gets flooded with warnings when `pg_stat_statements` extension is not installed in the database used by PMM Server or when PostgreSQL user is unable to connect to it
- [PMM-4852](#): Node name has an incorrect value if the Home dashboard opened after QAN
- [PMM-4847](#): Drill-downs from the Environment Overview dashboard doesn't show data for the preselected host
- [PMM-4841](#) and [PMM-4845](#): `pg_stat_statement` QAN Agent leaks database connections
- [PMM-4831](#): Clean-up representation of selectors names on MySQL-related dashboards for a better consistency

- [PMM-4824](#): Incorrectly calculated singlestat values on MySQL Instances Overview dashboard
- [PMM-4819](#): In case of the only one monitored host, its uptime is shown as a smaller value than the all hosts uptime due to the inaccurate rounding
- [PMM-4816](#): Set equal thresholds to avoid confusing singlestat color differences on a Home dashboard
- [PMM-4718](#): Labels are not fully displayed in the filter panel of the Query Details section in QAN
- [PMM-4545](#): Long queries are not fully visible in the Query Examples section in QAN

Help us improve our software quality by reporting any Percona Monitoring and Management bugs you encounter using our [bug tracking system](#).

---

Last update: 2021-06-25

## 7.35 Percona Monitoring and Management 2.0.1

Date: October 9, 2019

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance. You can run PMM in your own environment for maximum security and reliability. It provides thorough time-based analysis for MySQL, MongoDB, and PostgreSQL servers to ensure that your data works as efficiently as possible.

For install instructions, see [Installing Percona Monitoring and Management](#).

### Caution

PMM 2 is designed to be used as a new installation — please don't try to upgrade your existing PMM 1 environment.

### 7.35.1 Improvements

- [PMM-4779](#): Securely share dashboards with Percona
- [PMM-4735](#): Keep one old slowlog file after rotation
- [PMM-4724](#): Alt+click on check updates button enables force-update
- [PMM-4444](#): Return “what’s new” URL with the information extracted from the pmm-update package change log

### 7.35.2 Fixed bugs

- [PMM-4758](#): Remove Inventory rows from dashboards
- [PMM-4757](#): `qan_mysql_perfschema_agent` failed querying `events_statements_summary_by_digest` due to data types conversion
- [PMM-4755](#): Fixed a typo in the InnoDB AHI Miss Ratio formula
- [PMM-4749](#): Navigation from Dashboards to QAN when some Node or Service was selected now applies filtering by them in QAN
- [PMM-4742](#): General information links were updated to go to PMM 2 related pages
- [PMM-4739](#): Remove request instances list
- [PMM-4734](#): A fix was made for the collecting `node_name` formula at MySQL Replication Summary dashboard
- [PMM-4729](#): Fixes were made for formulas on MySQL Instances Overview
- [PMM-4726](#): Links to services in MongoDB singlestats didn’t show Node name
- [PMM-4720](#): `machine_id` could contain trailing `\n`
- [PMM-4640](#): It was not possible to add MongoDB remotely if password contained a `#` symbol

Help us improve our software quality by reporting any Percona Monitoring and Management bugs you encounter using our [bug tracking system](#).

---

Last update: 2021-06-25

## 7.36 Percona Monitoring and Management 2.0.0

Date: September 19, 2019

Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL, MongoDB, and PostgreSQL performance. You can run *PMM* in your own environment for the best security and reliability. It provides thorough time-based analysis for MySQL, MongoDB, and PostgreSQL servers to ensure that your data works as efficiently as possible.

For install instructions, see [Installing Percona Monitoring and Management](#).

### Caution

PMM 2 is designed to be used as a new installation — please don't try to upgrade your existing PMM 1 environment.

The new PMM2 introduces a number of enhancements and additional feature improvements, including:

- Detailed query analytics and filtering technologies which enable you to identify issues faster than ever before.
- A better user experience: Service-level dashboards give you immediate access to the data you need.
- The new addition of PostgreSQL query tuning.
- Enhanced security protocols to ensure your data is safe.
- Our new API allows you to extend and interact with third-party tools.

More details about new and improved features available within the release can be found [in the corresponding blog post](#).

Help us improve our software quality by reporting any Percona Monitoring and Management bugs you encounter using our [bug tracking system](#).

---

Last update: 2021-07-29