**How to install Percona Monitoring and Management**

**References:**

* [**https://www.youtube.com/watch?v=H968kpPX-a8&t=79s**](https://www.youtube.com/watch?v=H968kpPX-a8&t=79s)
* [**https://www.percona.com/doc/percona-monitoring-and- management/deploy/server/docker.setting-up.html**](https://www.percona.com/doc/percona-monitoring-and-%20management/deploy/server/docker.setting-up.html)
* [**https://niravshah2705-software-engineering.blogspot.com/2016/12/setup-percona-management-for-mysql.html**](https://niravshah2705-software-engineering.blogspot.com/2016/12/setup-percona-management-for-mysql.html)

**Required System :**

**CentOS 7 x64 – hostname: localhos**t – **IP: 210.211.124.175 (PMM-Server)**

**CentOS6 x64 – hostname: test01: 210.211.124.173 ( Percona XtraDB Cluster) (PMM-Client)**

1. **Install Docker-CE to build Percona Monitoring and Management (PMM-Server)**

**Note :** Open port for **pmm-server** and **pmm-client :**

**On CentOS 7 :**

**#** Open port 80,443,9002 for web services:

firewall-cmd --zone=public --permanent --add-service=http

firewall-cmd --zone=public --permanent --add-service=https

firewall-cmd --zone=public --permanent --add-port=9002/tcp

# **42000** For PMM to collect genenal system metrics.

firewall-cmd --zone=public --permanent --add-port=42000/tcp

# **42001** This port is used by a service which collects query performance data and makes it available to QAN.

firewall-cmd --zone=public --permanent --add-port=42001/tcp

# **42002** For PMM to collect MySQL server metrics.

firewall-cmd --zone=public --permanent --add-port=42002/tcp

# **42003** For PMM to collect MongoDB server metrics.

firewall-cmd --zone=public --permanent --add-port=42003/tcp

# **42004** For PMM to collect ProxySQL server metrics.

firewall-cmd --zone=public --permanent --add-port=42004/tcp

# **42005** For PMM to collect PostgreSQL server metrics.

firewall-cmd --zone=public --permanent --add-port=42005/tcp

**On CentOS6 :**

# vi /etc/sysconfig/iptables

**Add this line below into iptables:**

#### Web Services ####

-A INPUT -m state --state NEW -m multiport -m tcp -p tcp --dports 9002,80,443 -j ACCEPT

-A OUTPUT -m state --state NEW -m multiport -m tcp -p tcp --dports 9002,80,443 -j ACCEPT

#### Percona Monitoring and Management ####

-A INPUT -m state --state NEW,ESTABLISHED -s localhost,192.168.0.0/24,210.211.124.173,210.211.124.175 -m multiport -m tcp -p tcp --dports 42000,42001,42002,42003,42004,42005 -j ACCEPT

-A OUTPUT -m state --state NEW,ESTABLISHED -d localhost,192.168.0.0/24,210.211.124.173,210.211.124.175 -m multiport -m tcp -p tcp --dports 42000,42001,42002,42003,42004,42005 -j ACCEPT

**Step 1: Let's update the package database:**

# sudo yum check-update

**Step 2: Now run this command. It will add the official Docker repository, download the latest version of Docker, and install it:**

# curl -fsSL https://get.docker.com/ | sh

**After installation has completed, start the Docker daemon:**

[root@localhost ~]# systemctl start docker

**Verify that it's running**:

[root@localhost ~]# systemctl status docker

â— docker.service - Docker Application Container Engine

Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)

Active: active (running) since Mon 2019-02-25 09:37:41 +07; 10s ago

Docs: https://docs.docker.com

Main PID: 12560 (dockerd)

Tasks: 36

Memory: 47.9M

CGroup: /system.slice/docker.service

â”œâ”€12560 /usr/bin/dockerd -H fd://

â””â”€12579 containerd --config /var/run/docker/containerd/containerd.toml --log-level info

**Lastly, make sure it starts at every server reboot:**

[root@localhost ~]# systemctl enable docker

Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.

[root@localhost ~]#

**Step 3: Executing Docker Command Without Sudo (Optional**) **( If run with root, this step is not necessary)**

By default, running the docker command requires root privileges — that is, you have to prefix the command with sudo. It can also be run by a user in the docker group, which is automatically created during the installation of Docker. If you attempt to run the docker command without prefixing it with sudo or without being in the docker group, you'll get an output like this:

Output:

docker: Cannot connect to the Docker daemon. Is the docker daemon running on this host?.

See 'docker run --help'.

If you want **to avoid typing sudo** whenever **you run the docker command**, **add** your **username** to the **docker group**:

# sudo usermod -aG docker $(whoami)

You will need to log out of the Droplet and back in as the same user to enable this change.

If you need to add a user to the docker group that you're not logged in as, declare that username explicitly using:

#sudo usermod -aG docker username

The rest of this article assumes you are running the docker command as a user in the docker user group. If you choose not to, please prepend the commands with sudo.

1. **Installing Percona Monitoring and Management**

## Step 1 : [Pulling the PMM Server Docker Image](https://www.percona.com/doc/percona-monitoring-and-management/deploy/server/docker.setting-up.html#pmm-server-docker-image-pulling)

To pull the latest version from Docker Hub:

$ docker pull percona/pmm-server:latest

## 

## To check images by command :

# docker images

## 

This step is not required if you are running PMM Server for the first time. However, it ensures that if there is an older version of the image tagged with latest available locally, it will be replaced by the actual latest version.

## Step 2: [Creating the pmm-data Container](https://www.percona.com/doc/percona-monitoring-and-management/deploy/server/docker.setting-up.html#data-container)

To create a container for persistent PMM data, run the following command:

$ docker create \

-v /opt/prometheus/data \

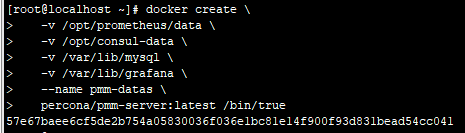
-v /opt/consul-data \

-v /var/lib/mysql \

-v /var/lib/grafana \

--name pmm-data \

percona/pmm-server:latest /bin/true

****

**Note**

This container does not run, it simply exists to make sure you retain all PMM data when you upgrade to a newer PMM Server image. Do not remove or re-create this container, unless you intend to wipe out all PMM data and start over.

The previous command does the following:

* The **docker create** command instructs the Docker daemon to create a container from an image.
* The -v options initialize data volumes for the container.
* The --name option assigns a custom name for the container that you can use to reference the container within a Docker network. In this case: pmm-data.
* percona/pmm-server:latest is the name and version tag of the image to derive the container from.
* /bin/true is the command that the container runs.

Important

Make sure that the data volumes that you initialize with the -v option match those given in the example. PMM Server expects that those directories are bind mounted exactly as demonstrated.

## Step 3: [Creating and Launching the PMM Server Container](https://www.percona.com/doc/percona-monitoring-and-management/deploy/server/docker.setting-up.html#server-container)

To create and launch PMM Server in one command, use **docker run**:

$ docker run -d \

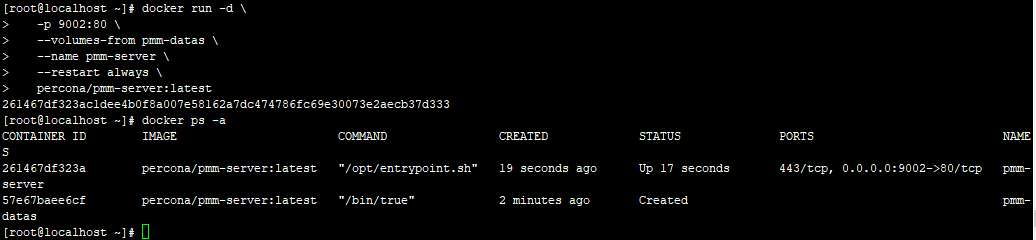
-p 9002:80 \

--volumes-from pmm-data \

--name pmm-server \

--restart always \

percona/pmm-server:latest



Check container by command :

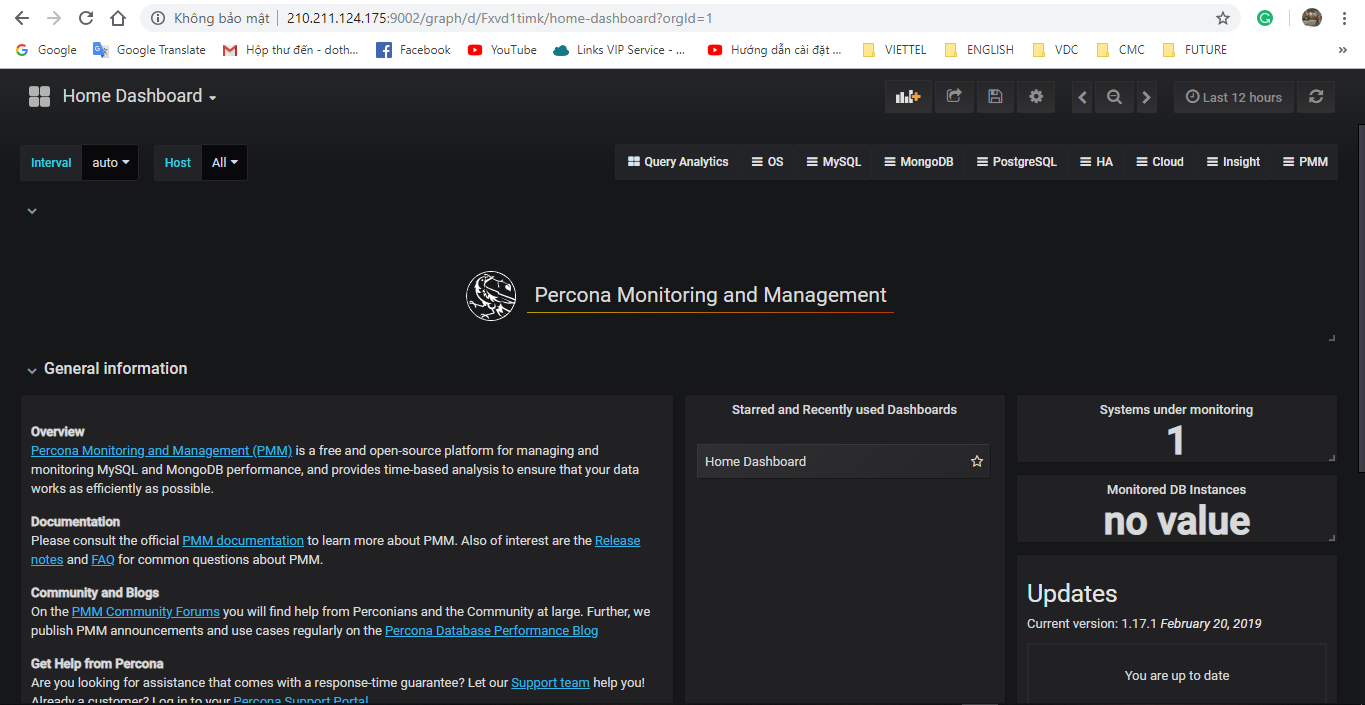
# docker ps -a

Note: By port default is 80:80 but we need to chage port differently to the defaut, Here we change it into 9002:80

This command does the following:

* The **docker run** command runs a new container based on the percona/pmm-server:latest image.
* The -d option starts the container in the background (detached mode).
* The -p option maps the port for accessing the PMM Server web UI. For example, if port **80** is not available, you can map the landing page to port 8080 using -p 8080:80.
* The -v option mounts volumes from the pmm-data container (see [Creating the pmm-data Container](https://www.percona.com/doc/percona-monitoring-and-management/deploy/server/docker.setting-up.html#data-container)).
* The --name option assigns a custom name to the container that you can use to reference the container within the Docker network. In this case: pmm-server.
* The --restart option defines the container’s restart policy. Setting it to always ensures that the Docker daemon will start the container on startup and restart it if the container exits.
* percona/pmm-server:latest is the name and version tag of the image to derive the container from.

## Step 4 : Login website 210.211.124.175:9002



## Setup Percona Management Client

### **Step1 : Install Client tool**

**Method 1:**

yum install <http://www.percona.com/downloads/percona-release/redhat/0.1-4/percona-release-0.1-4.noarch.rpm>

yum install pmm-client

or

get file : pmm-client-1.17.1-1.el6.x86\_64.rpm and install

**Method 2:**  
There are multiple ways of installation. One can use what suites them. We will perform with .tar.gz

wget https://www.percona.com/downloads/pmm-client/LATEST/binary/tarball/pmm-client-1.0.7.tar.gz

tar -xzf pmm-client-1.0.7.tar.gz

cd pmm-client-1.0.7

sudo ./install

### **Step 2: Configure Client to collect data**

* Before configure client we need to grant privileges to user ‘monitor’ of MySQL:

[MM 2019-02-25 14:09 ICT][root@test01:~]# mysql -u root -p

Enter password:

mysql > GRANT ALL PRIVILEGES ON \*.\* TO 'monitor'@'localhost' IDENTIFIED BY 'PmmRoot123' WITH MAX\_USER\_CONNECTIONS 5;

mysql > FLUSH PRIVILEGES;

mysql > exit

Or

mysql > GRANT SELECT, PROCESS, SUPER, REPLICATION CLIENT, RELOAD ON \*.\* TO 'monitor'@'localhost' IDENTIFIED BY 'PmmRoot123' WITH MAX\_USER\_CONNECTIONS 10;

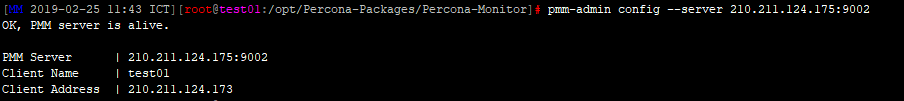
mysql > GRANT SELECT ON performance\_schema.\* TO 'monitor'@'localhost';

mysql > FLUSH PRIVILEGES;

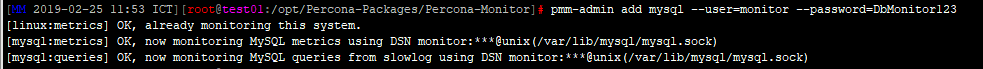
mysql > exit

* Configure pmm-client on server CentOS 6 : test01

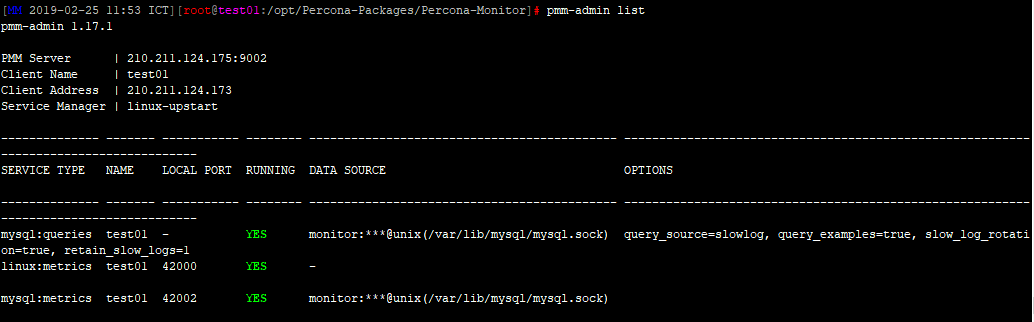
# pmm-admin config --server 210.211.124.175:9002



# pmm-admin add mysql --user=monitor --password=DbMonitor123



### Verify Client working fine



### Default UI credentials

We will connect to Percona UI with ssh tunnel setup, due to security reasons. Please see below URLs for using the application:

PMM landing page  http://localhost:9000/

Query Analytics (QAN web app)  http://localhost:9000/qan/

Metrics Monitor (Grafana)

http://localhost:9000/graph/

user name: admin

password: admin

Orchestrator  http://localhost:9000/orchestrator

### Upgrade PMM

Upgrading PMM environment is even simpler. As it is packaged with docker. We just have to replace it with newer docker images. Data will be preserved on server itself.

**Upgrade server:**

docker stop pmm-server && docker rm pmm-server  
docker run -d -p 80:80 --volumes-from pmm-data --name pmm-server --restart always percona/pmm-server:1.1.2  
docker pmm-server restart

**Upgrade client:**

rpm -Uvh pmm-client-1.1.2-1.x86\_64.rpm