MongoDB 모니터링 PMM 구성

1. PMM 구성 전 Docker 설치

1-1 . Yum-utils 업데이트

```
# yum install -y yum-utils
```

1-2 Docker-ce 레포 추가

```
# yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
```

1-3 Docker 설치

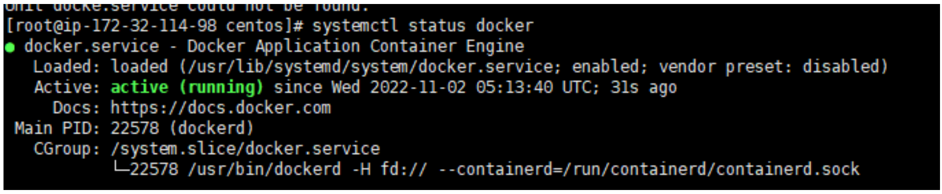
```
# yum install docker-ce docker-ce-cli containerd.io -y
```

1-4 Docker 시작 및 서비스 등록

```
# systemctl start docker

# systemctl enable docker

# systemctl status docker
```



```
Unit docke.service could not be found.
[root@ip-172-32-114-98 centos]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2022-11-02 05:13:40 UTC; 31s ago
     Docs: https://docs.docker.com
 Main PID: 22578 (dockerd)
   CGroup: /system.slice/docker.service
           └─22578 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

← 테스트 화면

2. 서버 설치

2-1. 데이터 볼륨 생성

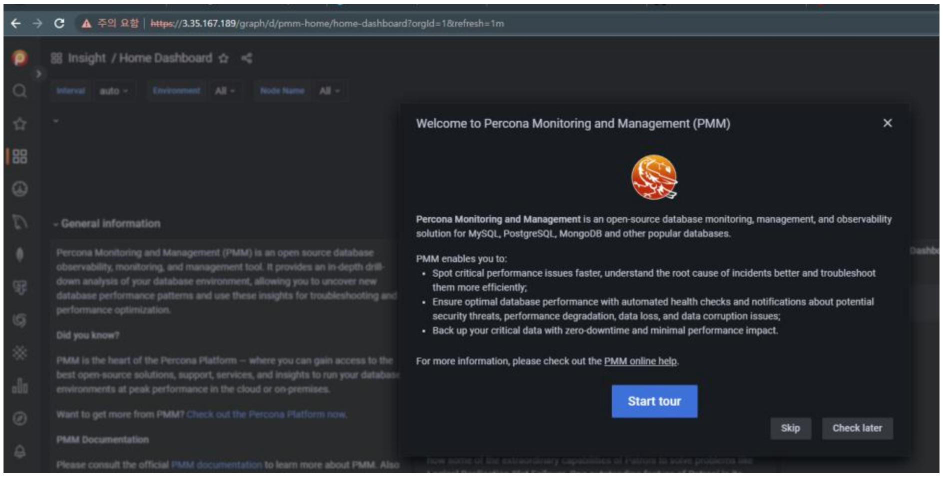
```
 # docker create -v /srv --name pmm-data percona/pmm-server:2 /bin/true
```

2-2. PMM-Server 도커 컨테이너 생성

```
 # docker run -d -p 80:80 -p 443:443 --volumes-from pmm-data --name
pmm-server --restart always percona/pmm-server:2
```

2-3 서버 접속

https://localhost:443 or https://<Public_IP>:443 으로 접속 → 초기 ID / 비밀번호 admin/admin



General information

Percona Monitoring and Management (PMM) is an open source database observability, monitoring, and management tool. It provides an in-depth drill-down analysis of your database environment, allowing you to uncover new database performance patterns and use these insights for troubleshooting and performance optimization.

Did you know?

PMM is the heart of the Percona Platform – where you can gain access to the best open-source solutions, support, services, and insights to run your database environments at peak performance in the cloud or on-premises.

Want to get more from PMM? Check out the Percona Platform now.

PMM Documentation

Please consult the official PMM documentation to learn more about PMM. Also

Welcome to Percona Monitoring and Management (PMM)

Percona Monitoring and Management is an open-source database monitoring, management, and observability solution for MySQL, PostgreSQL, MongoDB and other popular databases.

PMM enables you to:

- Spot critical performance issues faster, understand the root cause of incidents better and troubleshoot them more efficiently;
- Ensure optimal database performance with automated health checks and notifications about potential security threats, performance degradation, data loss, and data corruption issues;
- Back up your critical data with zero-downtime and minimal performance impact.

For more information, please check out the PMM online help.

3. 클라이언트 설치

3-1. PMM Client docker 이미지 설치

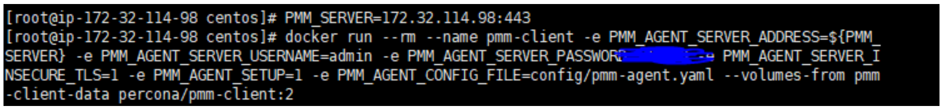
```
# docker pull ₩
> percona/pmm-client:2
```

3-2 Use the image as a template to create a persistent data store that preserves local data when the image is updated.

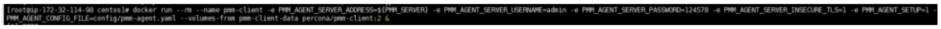
```
 # docker create ₩
> --volume /srv ₩
> --name pmm-client-data ₩
> percona/pmm-client:2 /bin/true
```

3-3. PMM Agent setup mode 로 container 를 실행해준다.

```
 # PMM_SERVER=X.X.X.X:443    ← 서버 IP 를 입력해주어야한다.
 # 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 ₩                ← 비밀번호를
수정했다면 바꿔주자
> -e PMM_AGENT_SERVER_INSECURE_TLS=1 ₩
> -e PMM_AGENT_SETUP=1 ₩
> -e PMM_AGENT_CONFIG_FILE=config/pmm-agent.yaml ₩
> --volumes-from pmm-client-data ₩
> percona/pmm-client:2
```



```
[root@ip-172-32-114-98 centos]# PMM_SERVER=172.32.114.98:443
[root@ip-172-32-114-98 centos]# docker run --rm --name pmm-client -e PMM_AGENT_SERVER_ADDRESS=${PMM_
SERVER} -e PMM_AGENT_SERVER_USERNAME=admin -e PMM_AGENT_SERVER_PASSWOR[redacted] PMM_AGENT_SERVER_I
NSECURE_TLS=1 -e PMM_AGENT_SETUP=1 -e PMM_AGENT_CONFIG_FILE=config/pmm-agent.yaml --volumes-from pmm
-client-data percona/pmm-client:2
```



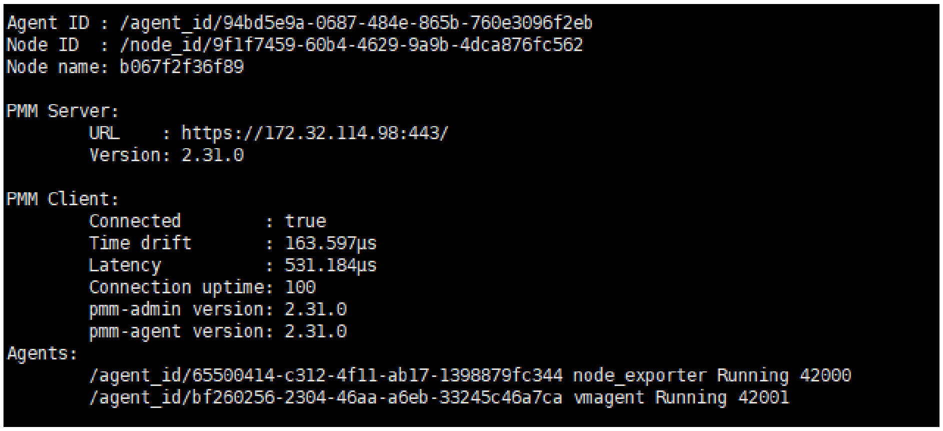
```
[root@ip-172-32-114-98 centos]# docker run --rm --name pmm-client -e PMM_AGENT_SERVER_ADDRESS=${PMM_SERVER} -e PMM_AGENT_SERVER_USERNAME=admin -e PMM_AGENT_SERVER_PASSWORD=124578 -e PMM_AGENT_SERVER_INSECURE_TLS=1 -e PMM_AGENT_SETUP=1 -
PMM_AGENT_CONFIG_FILE=config/pmm-agent.yaml --volumes-from pmm-client-data percona/pmm-client:2 &
```

```
 # docker run --rm --name pmm-client -e
PMM_AGENT_SERVER_ADDRESS=${PMM_SERVER} -e
PMM_AGENT_SERVER_USERNAME=admin -e
PMM_AGENT_SERVER_PASSWORD=124578 -e
PMM_AGENT_SERVER_INSECURE_TLS=1 -e PMM_AGENT_SETUP=1 -e
PMM_AGENT_CONFIG_FILE=config/pmm-agent.yaml --volumes-from pmm-
client-data percona/pmm-client:2 &
```

ctrl + c 를 누르고

```
 # docker exec pmm-client ₩
> pmm-admin status
```

로 상태를 확인해볼 수 있다.



```
Agent ID : /agent_id/94bd5e9a-0687-484e-865b-760e3096f2eb
Node ID  : /node_id/9f1f7459-60b4-4629-9a9b-4dca876fc562
Node name: b067f2f36f89

PMM Server:
        URL    : https://172.32.114.98:443/
        Version: 2.31.0

PMM Client:
        Connected        : true
        Time drift       : 163.597µs
        Latency          : 531.184µs
        Connection uptime: 100
        pmm-admin version: 2.31.0
        pmm-agent version: 2.31.0
Agents:
        /agent_id/65500414-c312-4f11-ab17-1398879fc344 node_exporter Running 42000
        /agent_id/bf260256-2304-46aa-a6eb-33245c46a7ca vmagent Running 42001
```