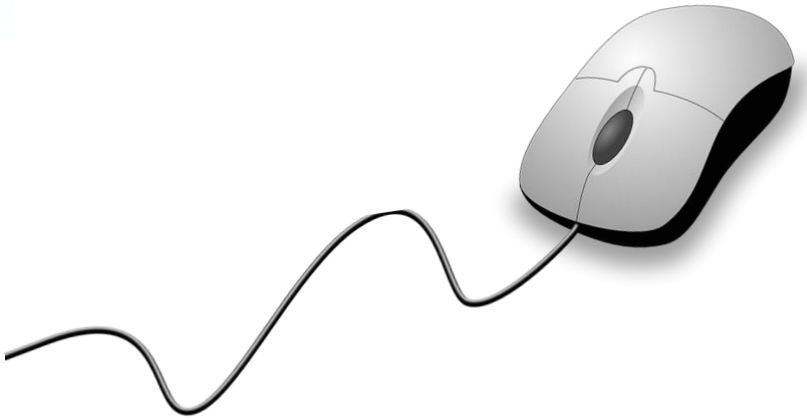
공개SW 솔루션 설치 & 활용 가이드

시스템SW > 자원관리



Prometheus

제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide

nipa 공개SW역량프라자

CONTENTS

1. 개요

소개	• 2012년 SoundCloud 에서 개발된 시스템 모니터링 및 경고 툴킷 • 2016년 Kubernetes 이후 두번째로 Cloud Native Computing Foundation 에 편입됨		
주요기능	• Metric 이름과 키 / 값 쌍으로 식별되는 시계열데이터가 있는 다차원 데이터 모델 • PromQL (Prometheus Query Language)이라는 기능적 쿼리 언어를 제공 • 다양한 그래프 및 대시 보드 지원 모드 • Target은 서비스 검색 또는 정적 구성을 통해 검색		
대분류	• 시스템SW	**소분류**	• 자원관리
라이선스형태	• Apache License v2.0	**사전설치 솔루션**	• 없음
		버전	• 2.11.0 (2019년 7월 기준)
특징	• Prometheus는 주로 Go로 작성 • Go,Java/JVM, C#/Net, Python, Ruby, Node.js, Haskell, Erlang, Rust등의 언어와 런타임에 클라이언트 라이브러리 지원 • Kubernetes 및 Docker 에는 Prometheus 클라이언트 라이브러리가 설치되어 있음 • Grafana 와 같은 대시보드 시스템에서 그래프로 나타낼 수 있음		
개발회사/커뮤니티	• Cloud Native Computing Foundation		
공식 홈페이지	• https://prometheus.io/		

2. 기능요약

Prometheus

- Prometheus 주요 기능

다차원 데이터	• 고도의 차원 데이터 모델을 구현합니다. 시계열은 Metric 이름과 키 - 값 쌍으로 식별됩니다.
강력한 쿼리	• PromQL을 사용하면 임시 그래프, 테이블 및 경고를 생성하기 위해 수집 된 시계열 데이터를 분할 및 다이싱 할 수 있습니다.
시각화	• 내장 된 표현 브라우저, Grafana 통합 및 콘솔 템플릿 언어와 같이 시각화를 위한 여러 모드가 있습니다.
효율적인 저장	• 효율적인 사용자 정의 형식으로 메모리 및 로컬 디스크에 Time series를 저장합니다. 스케일링은 Functional sharding 및 federation으로 달성됩니다.
간단한 조작	• 각 서버는 안정성을 위해 독립적이며 로컬 저장소에만 의존합니다. Go로 작성된 모든 바이너리는 정적으로 연결되어 배포가 용이합니다.

3. 실행환경

Prometheus

- OS 플랫폼 종류에 따른 지원
  - Linux: 32-bit, 64-bit, and ARM.
  - Mac OS X: 32-bit and 64-bit.
  - FreeBSD: 32-bit, 64-bit, and ARM.
  - OpenBSD: 32-bit, 64-bit, and ARM.
  - NetBSD: 32-bit, 64-bit, and ARM.
  - Microsoft Windows: 32-bit and 64-bit.
  - DragonFly: 64-bit

4. 설치 및 실행

세부 목차

4. 설치 및 실행

4.1 Prometheus 서버 설치 및 실행(CentOS - systemd)(1/5)

- OS Version : CentOS Linux release 7.6.1810 (Core)
- Prometheus Version : prometheus-2.11.0.linux-amd64
- Prometheus 공식 홈페이지 에서 다운

https://prometheus.io/download/

<2019년 7월 10일 기준으로 최신버전이 2.11>

1. 소스 파일 다운로드

   # wget https://github.com/Prometheus/Prometheus/releases/download/v2.11.0/Prometheus-2.11.0.linux-amd64.tar.gz

2. 소스 파일 압축 해제

   # tar zvxf prometheus-2.11.0.linux-amd64.tar.gz

3. 해당 폴더로 이동

   # cd prometheus-2.11.0.linux-amd64

4. 버전 확인

   # ./prometheus –version

```
[root@prometheus prometheus-2.11.0.linux-amd64]# ./prometheus --version
prometheus, version 2.11.0 (branch: HEAD, revision: 4ef66003d9855ed2b7a41e987b33828ec36db34d)
  build user:       root@0dc27cf95f36
  build date:       20190709-09:54:35
  go version:       go1.12.7
```

4. 설치 및 실행

4.1 Prometheus 서버 설치 및 실행(CentOS - systemd)(2/5)

5. User 생성

```
# useradd --no-create-home --shell /bin/false prometheus
```

6. 필요한 디렉토리 생성

```
# mkdir /etc/prometheus
# mkdir /var/lib/prometheus
```

7. 생성한 디렉토리에 소유권 설정

```
# chown Prometheus:Prometheus /etc/prometheus
# chown prometheus:prometheus /var/lib/prometheus
```

8. 이전에 압축해제한 폴더에서 바이너리 파일을 전역폴더로 복사

```
# cp prometheus-2.11.0.linux-amd64/prometheus /usr/local/bin/
# cp prometheus-2.11.0.linux-amd64/promtool /usr/local/bin/
```

4. 설치 및 실행

4.1 Prometheus 서버 설치 및 실행(CentOS - systemd)(3/5)

9. 복사한 파일에 대해서 소유권 설정

#chown prometheus:prometheus /usr/local/bin/prometheus

#chown Prometheus:Prometheus /usr/local/bin/promtool

10. 이전 소스폴더에서 consoles, console_libraries 디렉토리를 전체 복사한다.

cp -r prometheus-2.11.0.linux-amd64/consoles /etc/prometheus/

cp -r prometheus-2.11.0.linux-amd64/console_libraries /etc/prometheus/

11. prometheus 설정 파일 생성

vim /etc/Prometheus/Prometheus.yml

파일 구성

```
[root@prometheus ~]# cat /etc/prometheus/prometheus.yml
global:
  scrape_interval: 10s

scrape_configs:
  - job_name: 'prometheus_master'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']
```

4. 설치 및 실행

4.1 Prometheus 서버 설치 및 실행(CentOS - systemd)(4/5)

12. 설정파일 소유권 설정

```
# chown prometheus:prometheus /etc/prometheus/prometheus.yml
```

13. Systemd 설정 파일 구성

```
# vim /etc/systemd/system/prometheus.service
```

```
[root@prometheus ~]# cat /etc/systemd/system/prometheus.service
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus ₩
--config.file /etc/prometheus/prometheus.yml ₩
--storage.tsdb.path /var/lib/prometheus/ ₩
--web.console.templates=/etc/prometheus/consoles ₩
--web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
```

4. 설치 및 실행

4.1 Prometheus 서버 설치 및 실행(CentOS - systemd)(5/5)

14. Systemd 구성 및 시작

```
# systemctl daemon-reload
# systemctl start prometheus
```

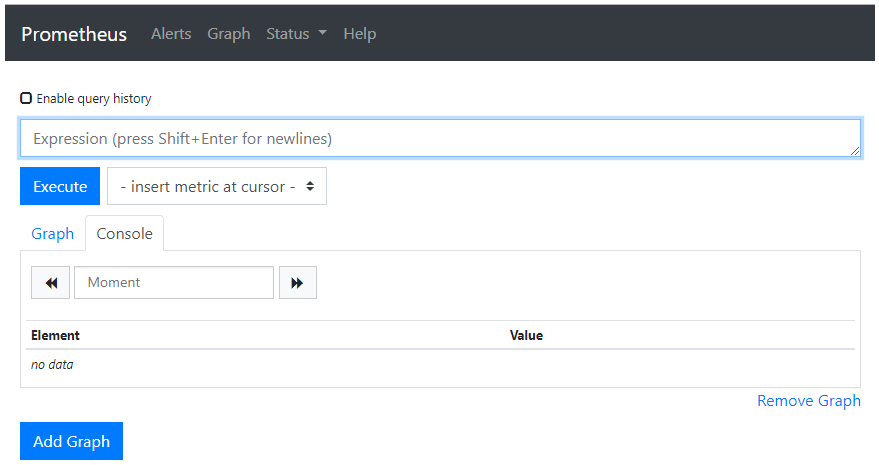
15. Firewall 포트 설정

```
# firewall-cmd --zone=public --add-port=9090/tcp --permanent
# systemctl reload firewalld
```

16. 브라우저에서 접속 테스트

```
# http://<IP주소>:9090
```

오른쪽 이미지처럼 나온다면 설치완료



4. 설치 및 실행

4.2 Node-exporter 설치 및 실행 (CentOS – Systemd기반)(5/5)

- OS Version : CentOS Linux release 7.6.1810 (Core)
- Node-exporter version
- Server의 상태를 모니터링 하기 위해 설치
- Prometheus 공식 홈페이지 에서 다운

https://prometheus.io/download/

<2019년 7월 10일 기준으로 최신버전이 2.11>

1. 소스 파일 다운로드

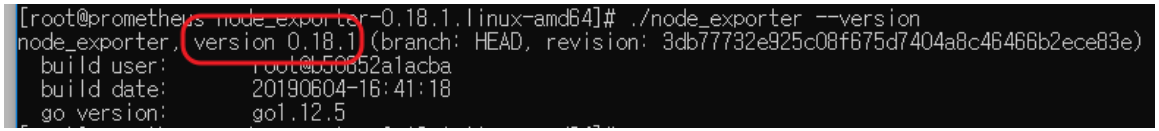
   # wget https://github.com/prometheus/node_exporter/releases/download/v0.18.1/node_exporter-0.18.1.linux-amd64.tar.gz

2. 소스 압축 해제

   # tar zvxf node_exporter-0.18.1.linux-amd64.tar.gz

3. 버전 확인

   # ./node_exporter --version



4. 설치 및 실행

4.2 Node-exporter 설치 및 실행 (CentOS – Systemd기반)(5/5)

4. Node-exporter를 위한유저 생성

   # useradd –rs /bin/false nodeusr

5. 이전에 압축해제한 폴더에서 바이너리 파일을 전역폴더로 복사

   # cp node_exporter-0.18.1.linux-amd64/node_exporter /usr/local/bin/

6. Systemd 설정 파일 구성

   # vim /etc/systemd/system/node_exporter.service

```
[root@prometheus bin]# cat /etc/systemd/system/node_exporter.service
[Unit]
Description=Node Exporter
After=network.target

[Service]
User=nodeusr
Group=nodeusr
Type=simple
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=multi-user.target
```

4. 설치 및 실행

4.2 Node-exporter 설치 및 실행 (CentOS – Systemd기반)(5/5)

Prometheus

7. Systemd 구성 및 시작

```
# systemctl daemon-reload
# systemctl start node_exporter
# systemctl status node_exporter
# systemctl enable node_exporter
```

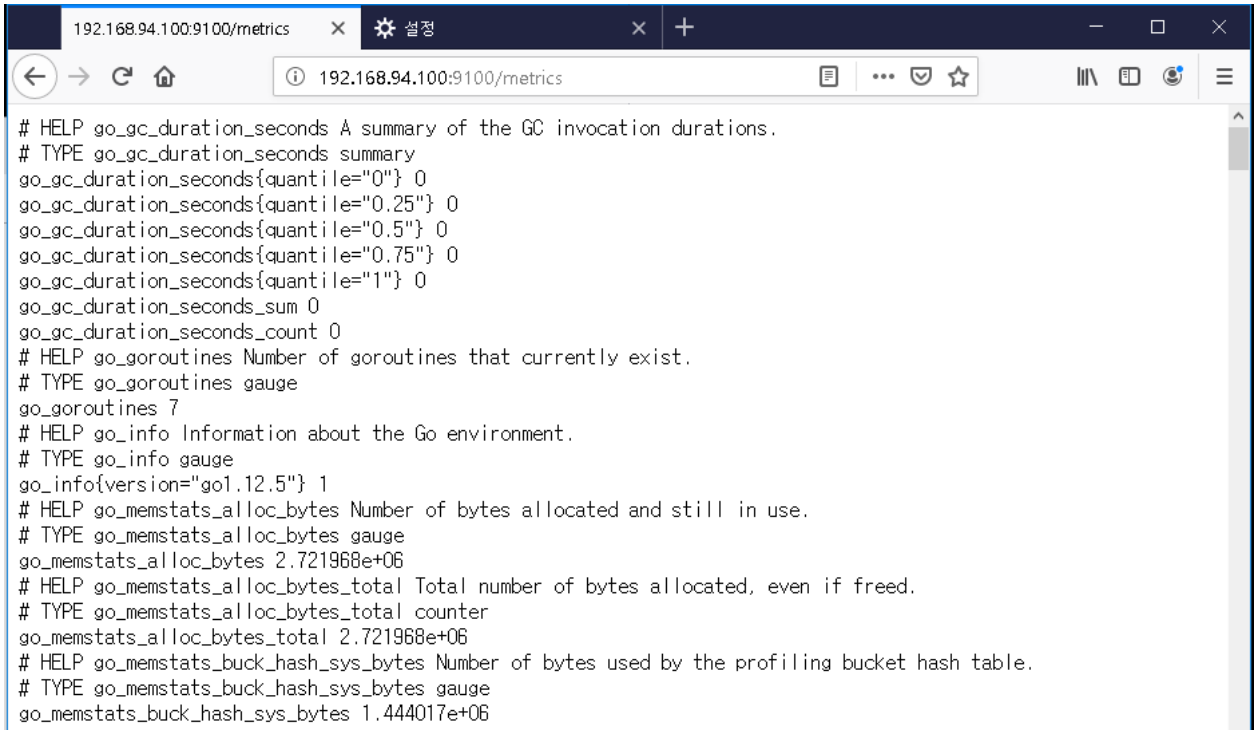
8. Firewall 포트 설정

```
# firewall-cmd --zone=public --add-port=9100/tcp –permanent
# systemctl reload firewalld
```

4. 설치 및 실행

4.2 Node-exporter 설치 및 실행 (CentOS – Systemd기반)(5/5)

9. Web 으로 접속하여 실행상태 확인



4. 설치 및 실행

4.1 Prometheus 서버 설치 및 실행(CentOS - systemd)(1/5)

- OS Version : CentOS Linux release 7.6.1810 (Core)
- Alertmanager Version : alertmanager-0.18.0.linux-amd64
- Prometheus 공식 홈페이지 에서 다운

https://prometheus.io/download/

<2019년 7월 10일 기준으로 최신버전이 0.18>

1. 소스 파일 다운로드

   # wget https://github.com/prometheus/alertmanager/releases/download/v0.18.0/alertmanager-0.18.0.linux-amd64.tar.gz

2. 소스 파일 압축 해제

   # tar zvxf alertmanager-0.18.0.linux-amd64.tar.gz

3. 해당 폴더로 이동

   # cd alertmanager-0.18.0.linux-amd64

4. 버전 확인

   # ./alertmanager –version

```
[root@prometheus alertmanager-0.18.0.linux-amd64]# ./alertmanager --version
alertmanager, version 0.18.0 (branch: HEAD, revision: 1ace0f76b7101cccc149d7298022df36039858ca)
  build user:       root@868685ed3ed0
  build date:       20190708-14:31:49
  go version:       go1.12.6
```

4. 설치 및 실행

4.2 Node-exporter 설치 및 실행 (CentOS – Systemd기반)(5/5)

5. 유저 생성

```
# useradd --no-create-home --shell /bin/false alertmanager
```

6. 필요한 디렉토리 생성

```
# sudo mkdir /etc/alertmanager
# sudo mkdir /etc/alertmanager/template
# sudo mkdir -p /var/lib/alertmanager/data
```

7. 생성한 디렉토리에 소유권 설정

```
# sudo chown -R alertmanager:alertmanager /etc/alertmanager
# sudo chown -R alertmanager:alertmanager /var/lib/alertmanager
```

8. 실행 파일 복사

```
# cp alertmanager-0.18.0.linux-amd64/alertmanager /usr/local/bin/
# cp alertmanager-0.18.0.linux-amd64/amtool /usr/local/bin/
```

4. 설치 및 실행

4.2 Node-exporter 설치 및 실행 (CentOS – Systemd기반)(5/5)

9. Systemd 설정 파일 구성

vim /etc/systemd/system/alertmanager.service

```
[root@prometheus ~]# cat /etc/systemd/system/alertmanager.service
[Unit]
Description=Prometheus Alertmanager Service
Wants=network-online.target
After=network.target

[Service]
User=alertmanager
Group=alertmanager
Type=simple
ExecStart=/usr/local/bin/alertmanager ₩
    --config.file /etc/alertmanager/alertmanager.yml ₩
    --storage.path /var/lib/alertmanager/data
Restart=always

[Install]
WantedBy=multi-user.target
```

10. Systemd 구성

systemctl daemon-reload

systemctl enable alertmanager

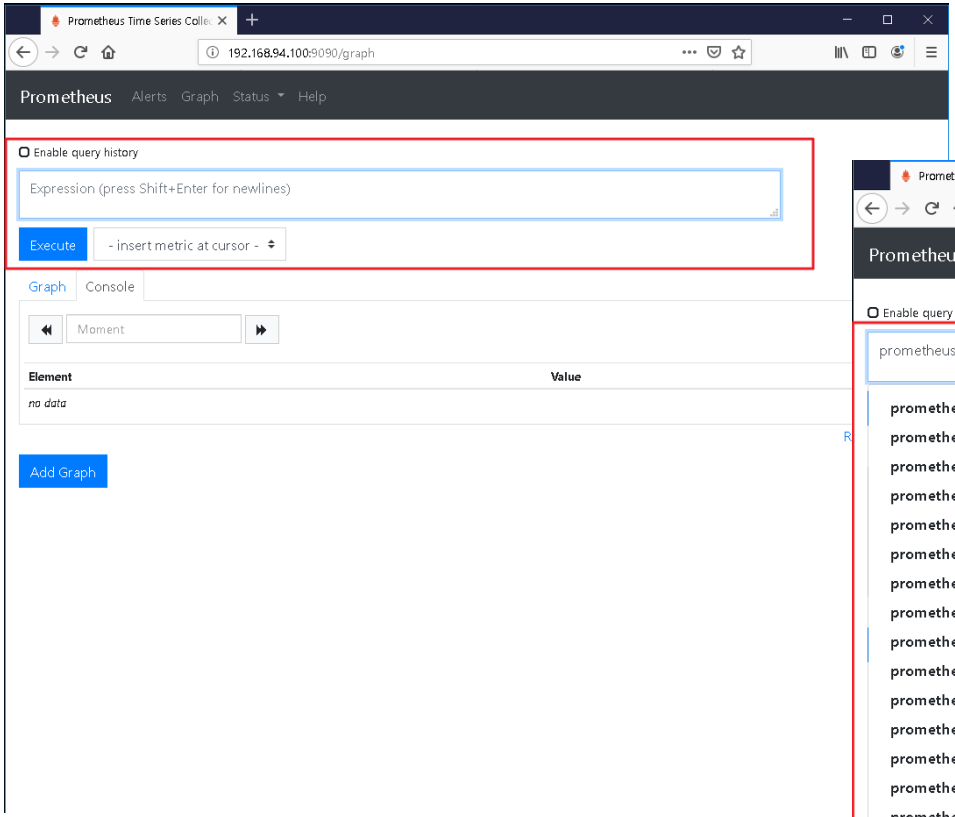
5. 기능소개

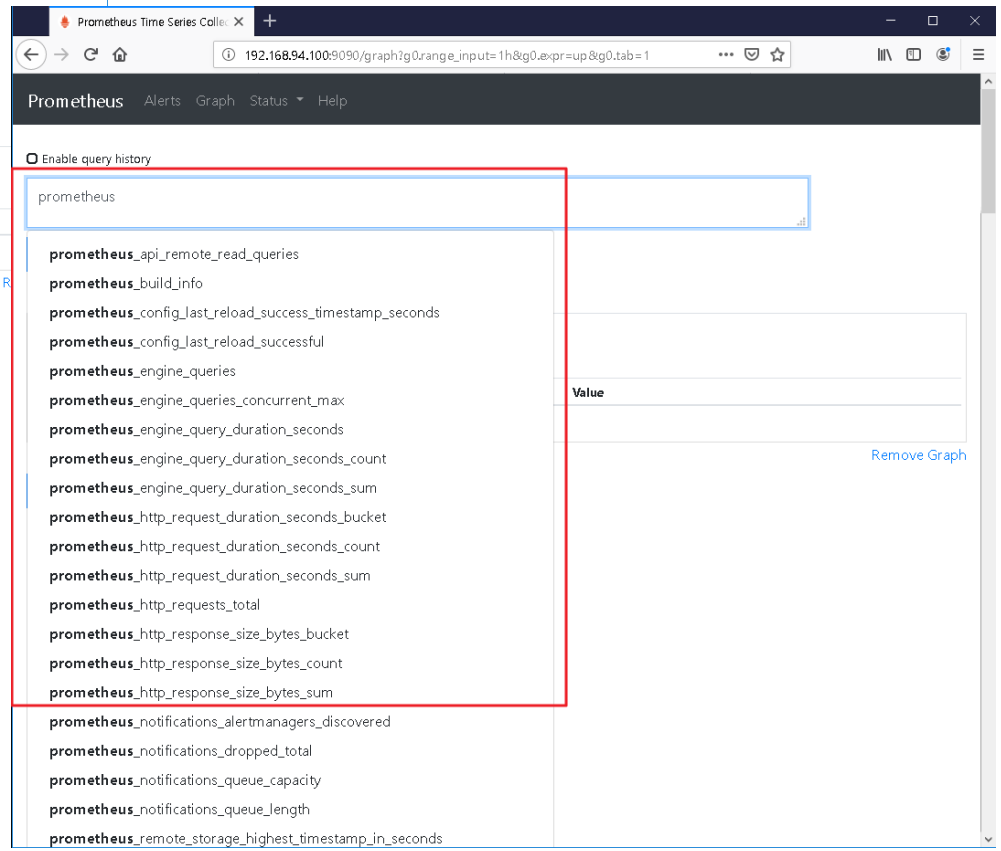
세부 목차

5. 기능소개

5.1 Prometheus Dashboard에서 PromQL으로 검색

- Prometheus의 검색창에서 다양한 값으로 검색할 수 있음

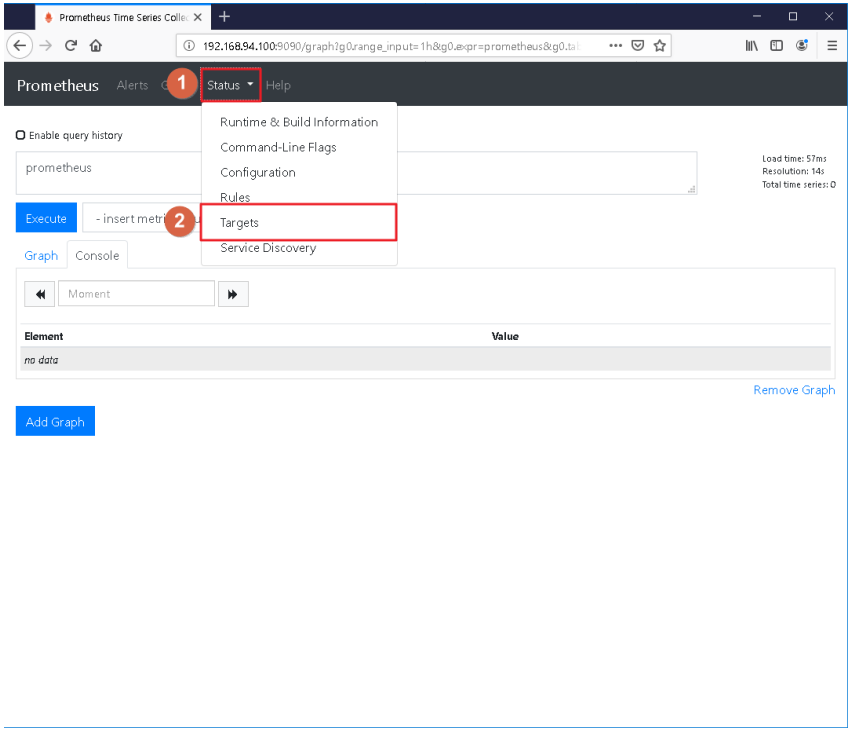


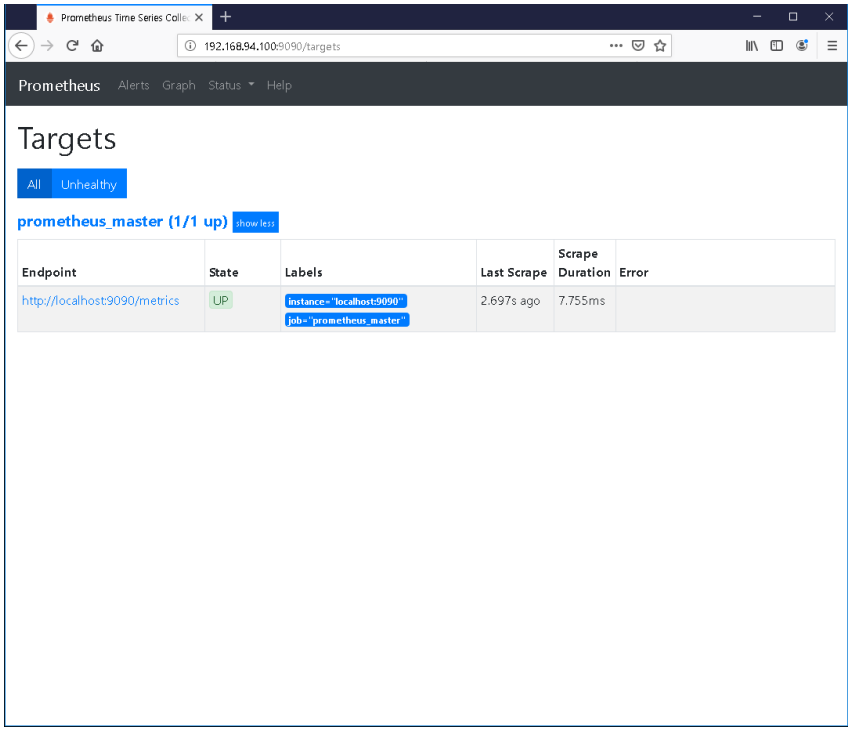


5. 기능소개

5.2 모니터링 대상 확인

- Prometheus의 Dashboard에서 모니터링 되고 있는 대상이 확인이 가능함

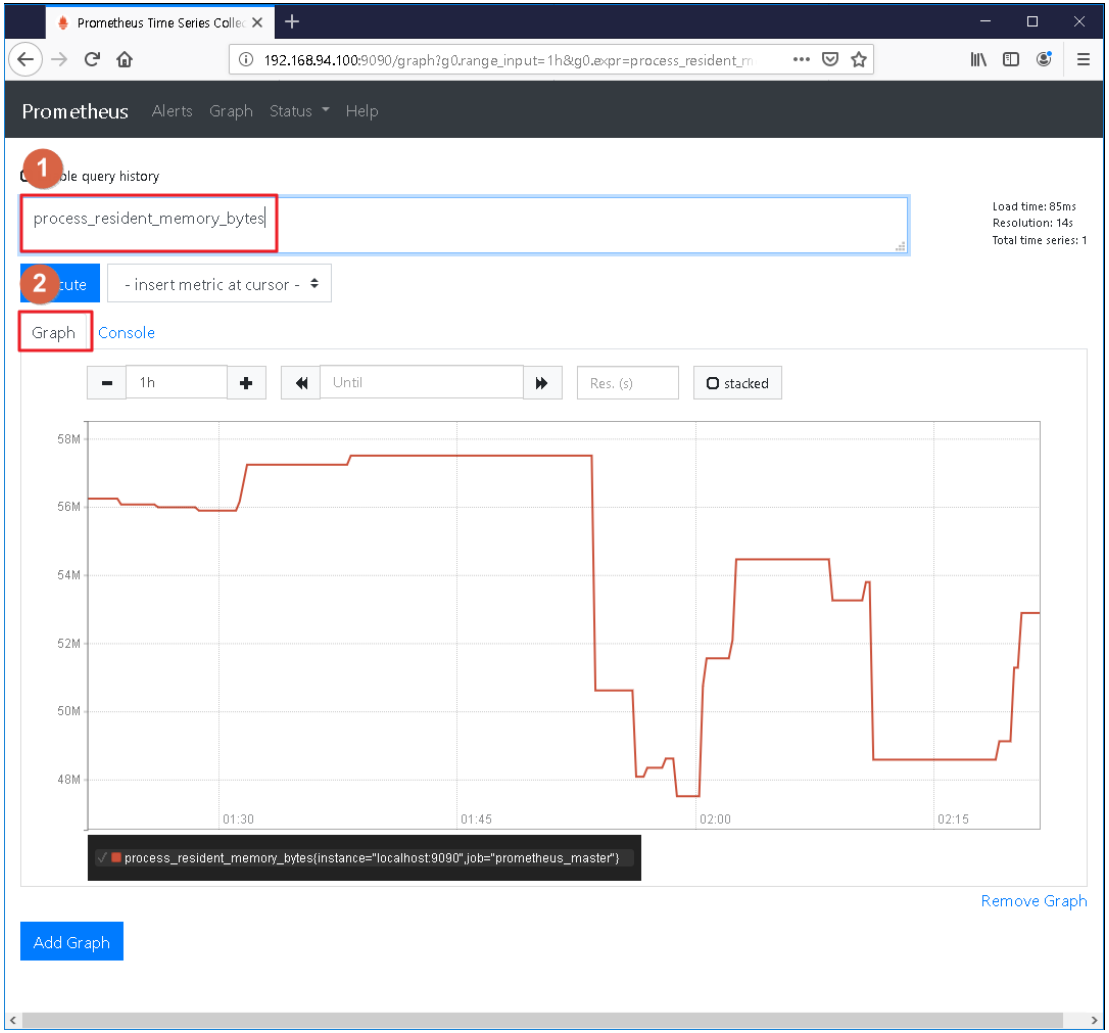




5. 기능소개

5.2 모니터링 대상 확인

- Prometheus의 Dashboard에서 모니터링 되고 있는 그래프 확인 가능



6. 활용예제

세부 목차

6. 활용예제

6.1 모니터링 대상 노드 추가

- Prometheus에 노드를 추가하여 모니터링

1. Prometheus 설정파일 맨 아래라인에 빨간 내용을 추가

```
# vim /etc/prometheus/prometheus.yml
```

```
scrape_configs:
  - job_name: 'prometheus_master'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']
  - job_name: node
    static_configs:
     - targets: ['localhost:9100']
```

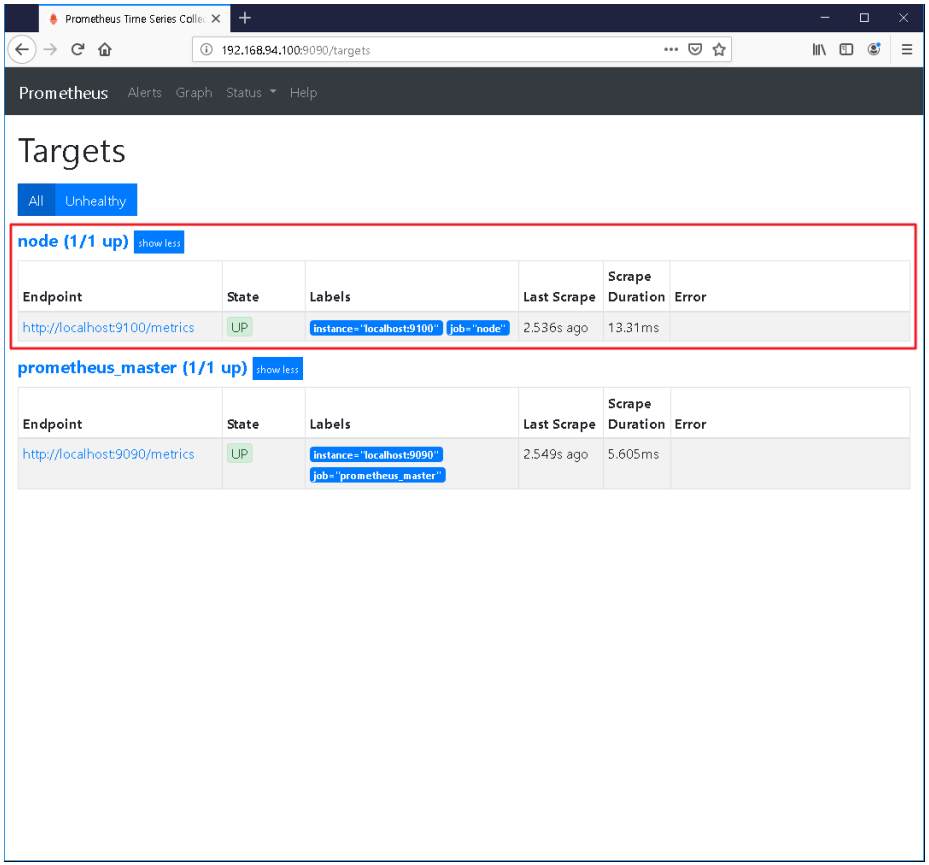
6. 활용예제

6.1 모니터링 대상 노드 추가

2. Prometheus 서버 재시작

```
# systemctl restart Prometheus
```

3. Prometheus Dashboard 에서 노드가 추가된 것 확인



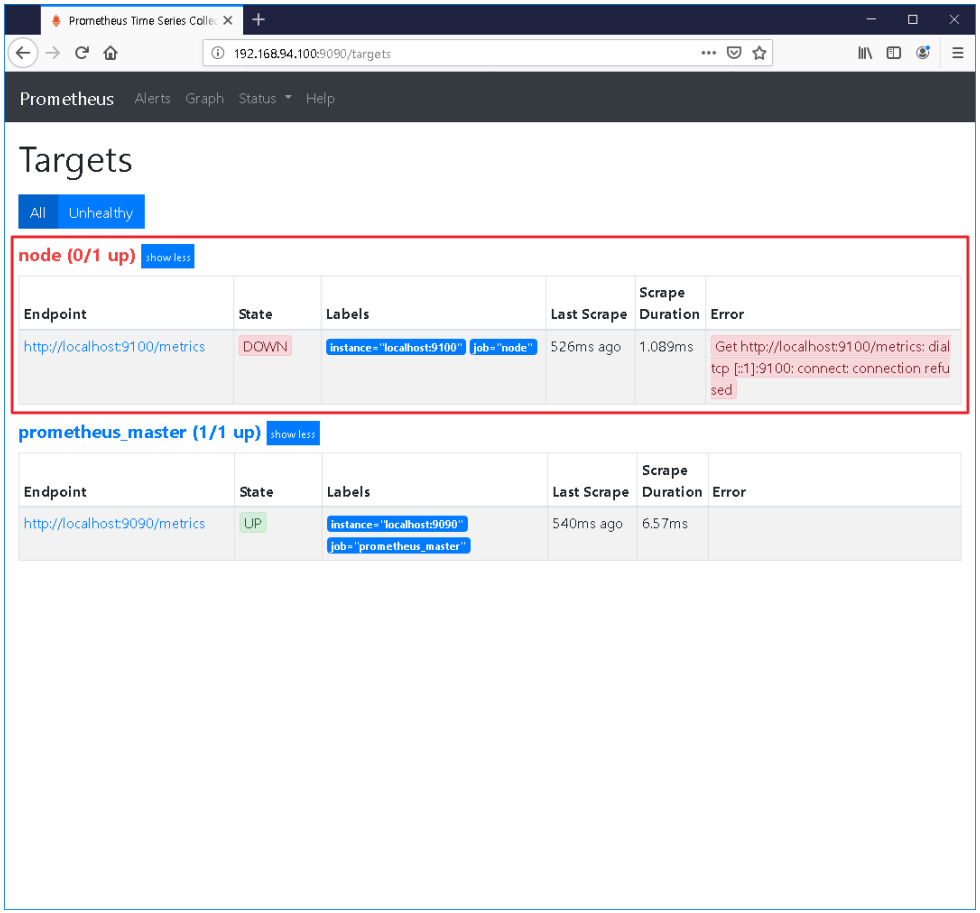
6. 활용예제

6.2 Alertmanager를 통해 이메일로 통보

- Alertmanager를 통해 이메일로 통보를 받는 설정

1. 먼저 알람 상태를 만들기 위해 node_exporter 정지하여 down 상태 설정

```
# systemctl stop node_exporter
```



6. 활용예제

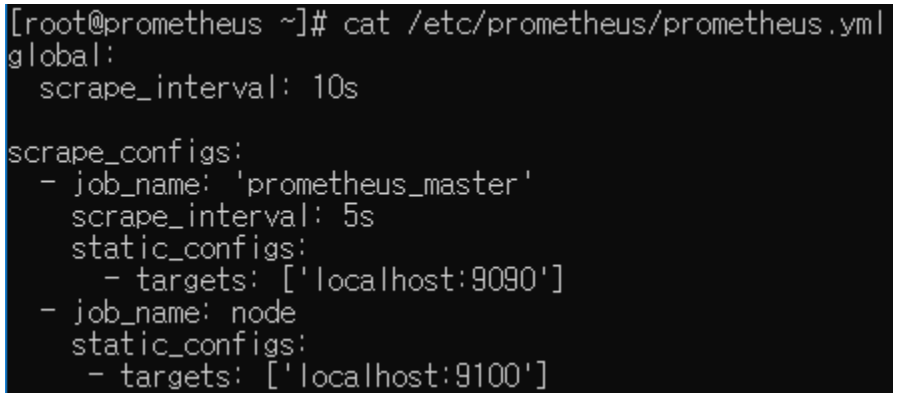
6.2 Alertmanager를 통해 이메일로 통보

2. Prometheus 설정파일에 아래 빨간 부분을 추가

vim /etc/prometheus/prometheus.yml

```
global:
  scrape_interval: 10s
rule_files:
  - rules.yml
alerting:
  alertmanagers:
  - static_configs:
    - targets: ['localhost:9093']


scrape_configs:
  - job_name: 'prometheus_master'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']
  - job_name: node
    static_configs:
      - targets: ['localhost:9100']
```



6. 활용예제

6.2 Alertmanager를 통해 이메일로 통보

3. rules.yml 파일 설정

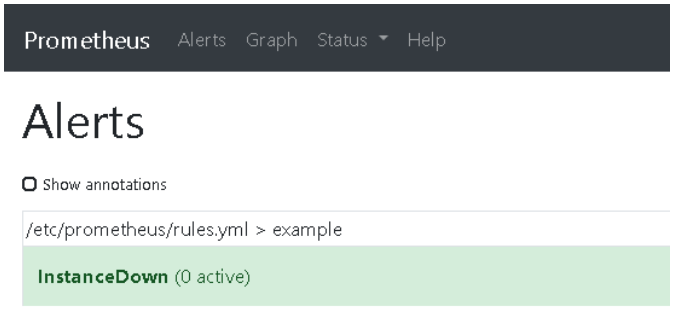
vim /etc/prometheus/rules.yml

```
groups:
 - name: example
   rules:
   - alert: InstanceDown
     expr: up == 0
     for: 1m
```

5. Prometheus 서버 재시작

systemctl restart Prometheus

6. Prometheus 에서 alert이 생성된것을 확인



6. 활용예제

6.2 Alertmanager를 통해 이메일로 통보

7. Alermanager 설정파일을 다음과 같이 구성

```
# vim /etc/alertmanager/alertmanager.yml
```

```
Route:
 group_by: [Alertname]
 # Send all notifications to me.
 receiver: email-notifications


receivers:
- name: email-notifications
 email_configs:
 - to: example@example.com
   from: example@example.com
   smarthost: smpt.address:<port number>
   auth_username: “<example@example.com>"
   auth_identity: “<example@example.com>"
   auth_password: “<user-password>"
```

NOTE
Google과 같은 계정을 통해 이메일 통보를 설정할 수 있음

```
route:
 group_by: [Alertname]
 # Send all notifications to me.
 receiver: email-notifications

receivers:
- name: email-notifications
 email_configs:
 - to: <username>@gmail.com
   from: <username>@gmail.com
   smarthost: smtp.gmail.com:587
   auth_username: “<username>@gmail.com"
   auth_identity: “<username>@gmail.com"
   auth_password: “<user-password>"
```

6. 활용예제

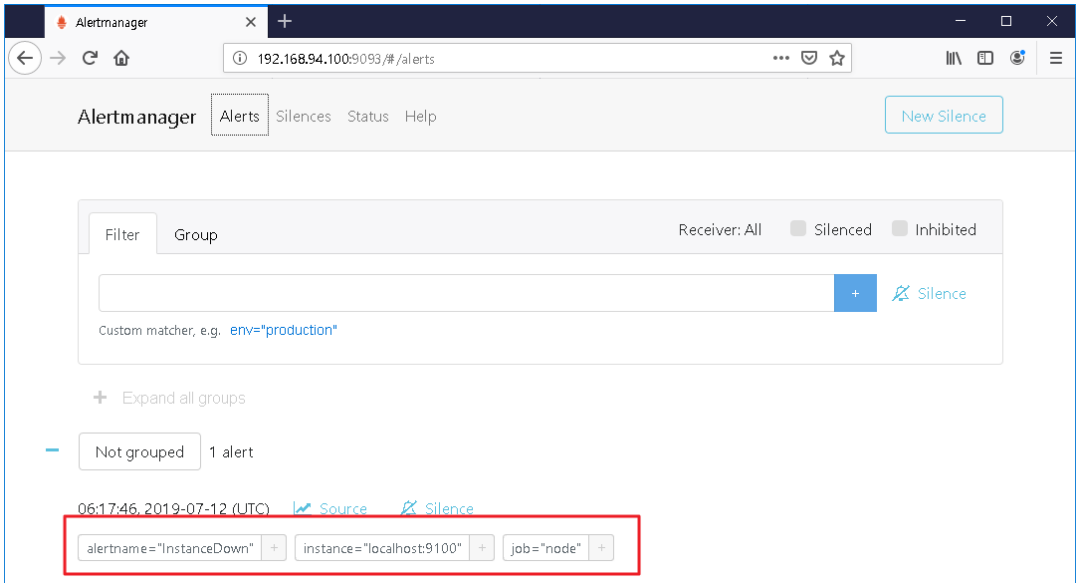
6.2 Alertmanager를 통해 이메일로 통보

8. Alertmanager 를 재시작

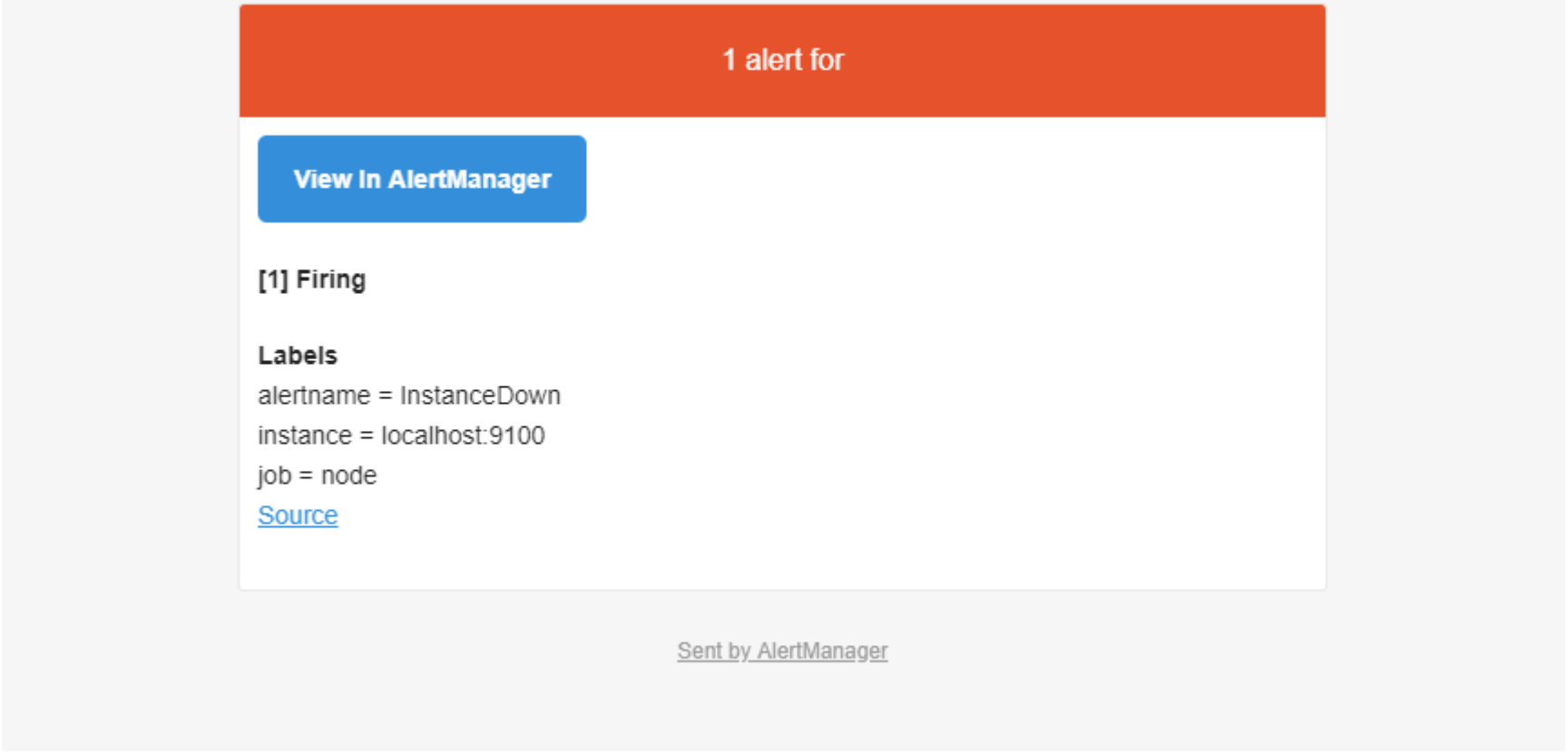
systemctl restart alertmanager

9. Prometheus Alertmanager 에서 확인

9093 포트로 접속 다음과 같이 Alert 이 등록된것을 확인



10. Alertmanager에 의해서 보내진 이메일 확인

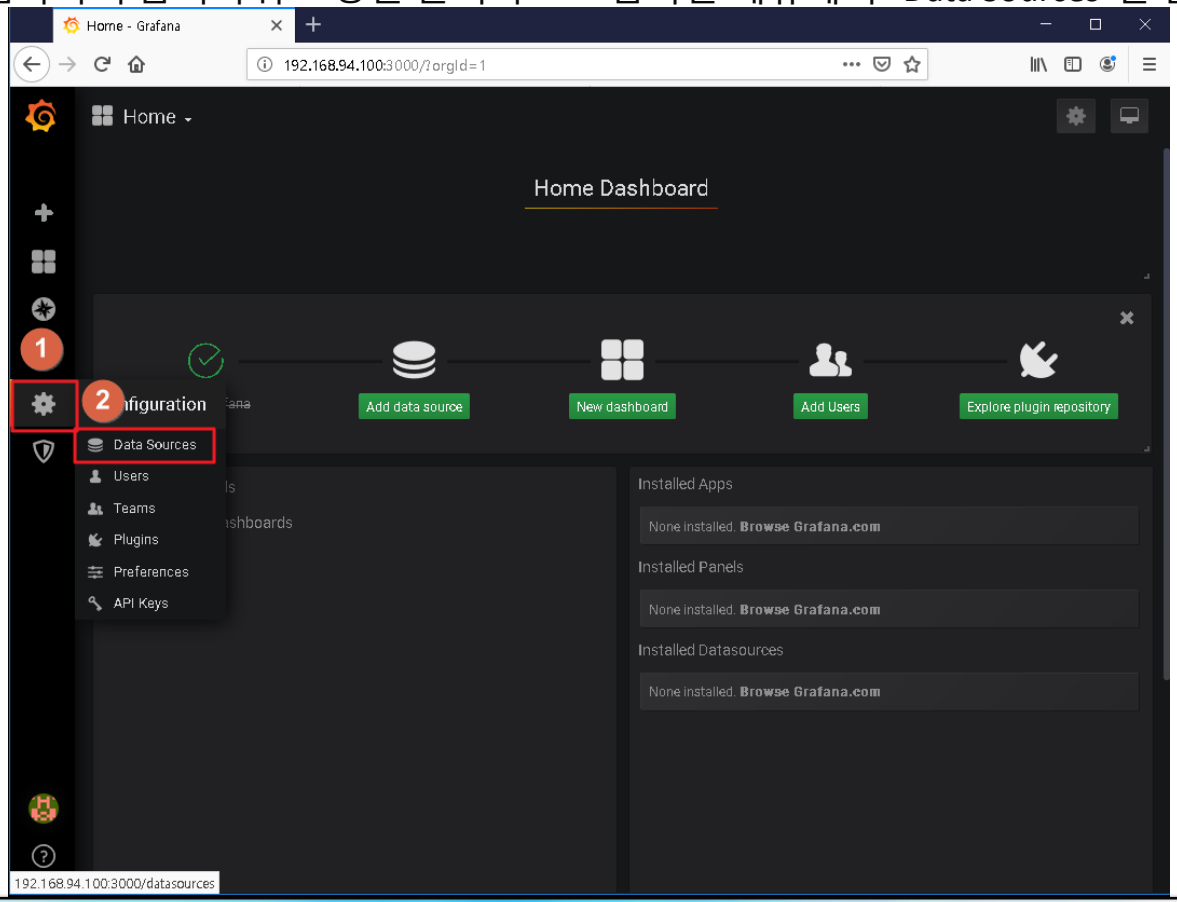


6. 활용예제

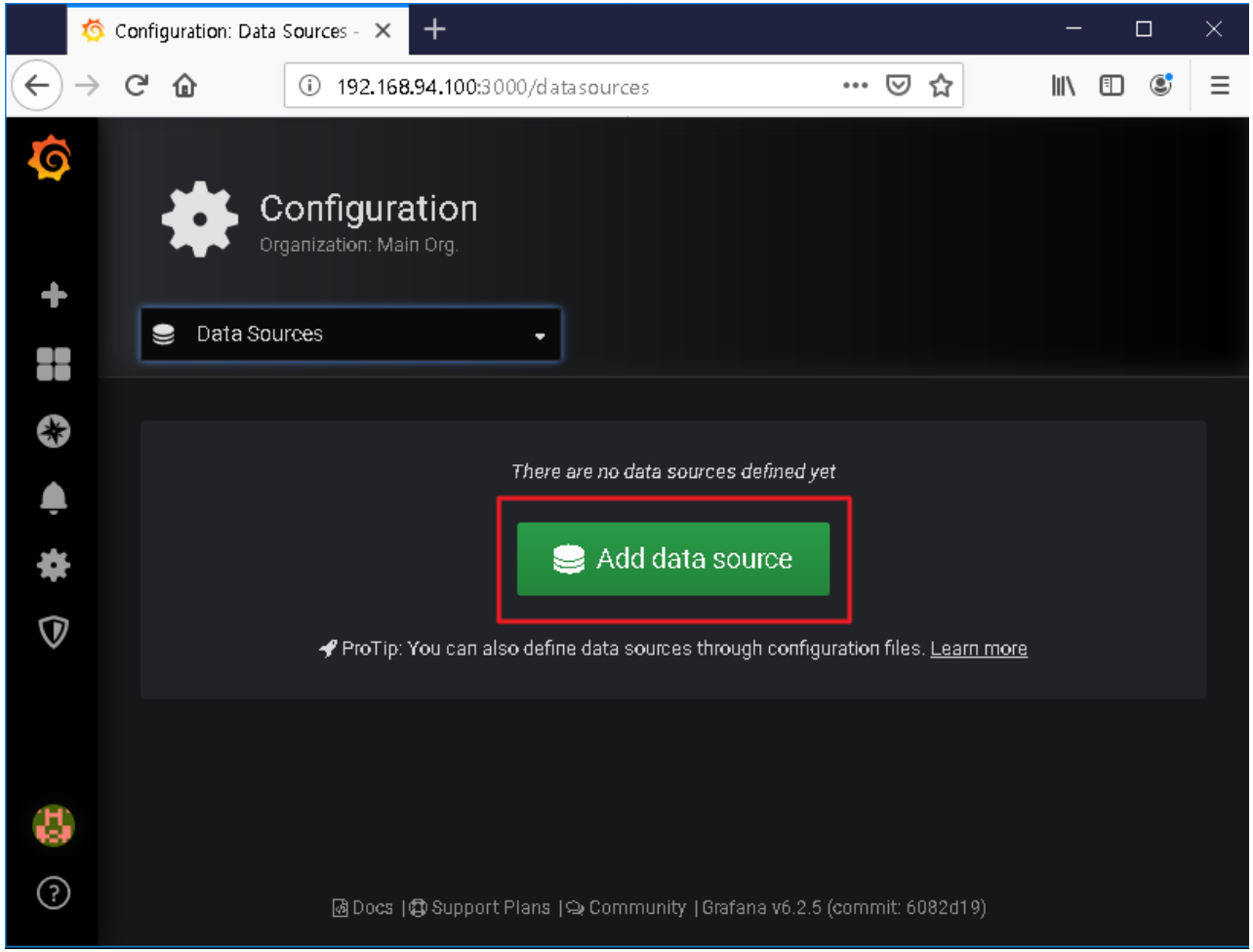
6.3 Grafana와 연동하여 시각화 설정

- Grafana는 데이터를 전달받아 Metric을 쿼리하고 시각화하는 도구
- Grafana가 설치되어 있다고 가정하고 설정을 진행

1. Grafana에 접속하여 톱니바퀴 모양을 클릭하고 드롭다운 메뉴에서 “Data Sources”를 클릭.



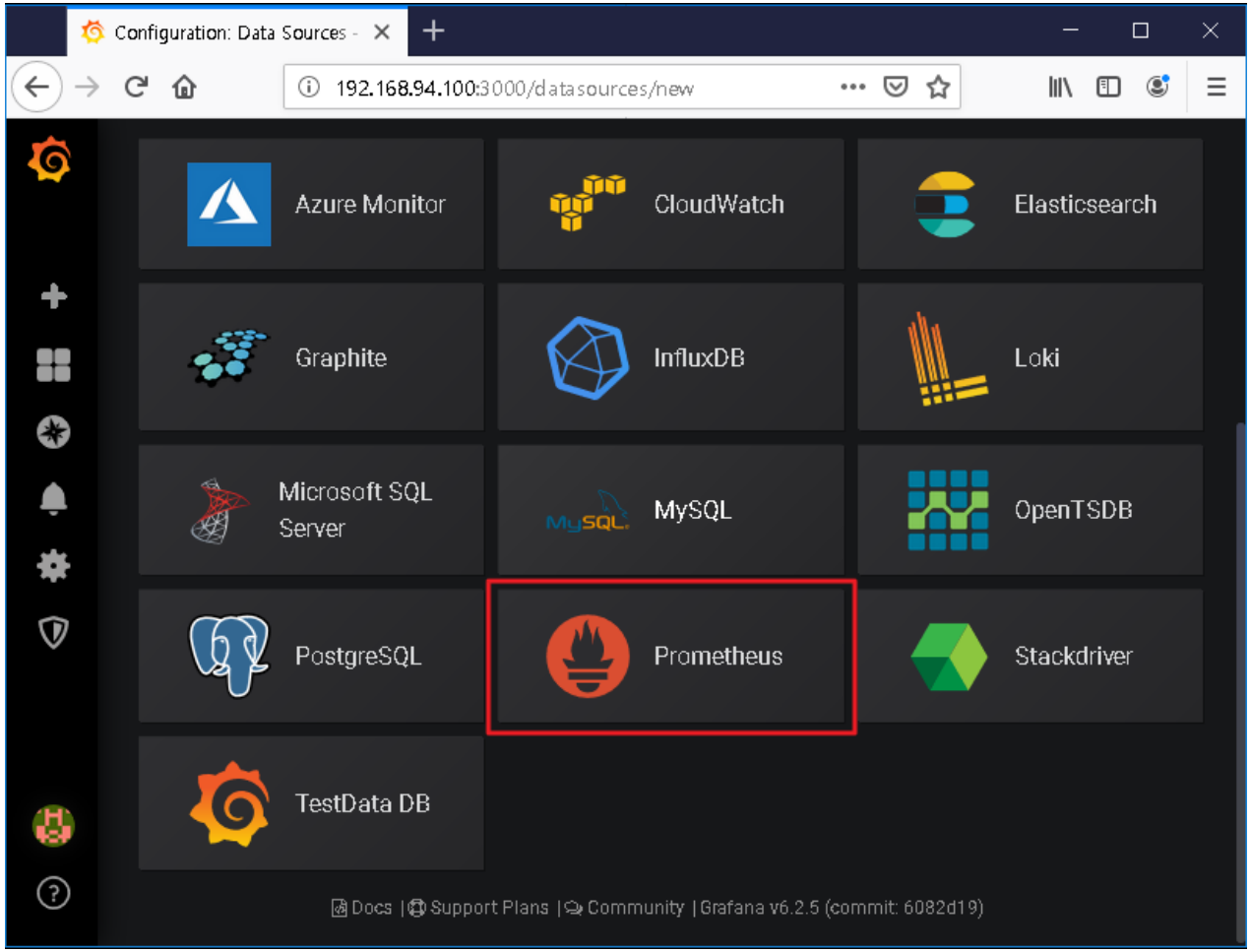
2. “Add data source” 클릭



6. 활용예제

6.3 Grafana와 연동하여 시각화 설정

3. Data Sources 리스트에서 “Prometheus” 를 클릭



6. 활용예제

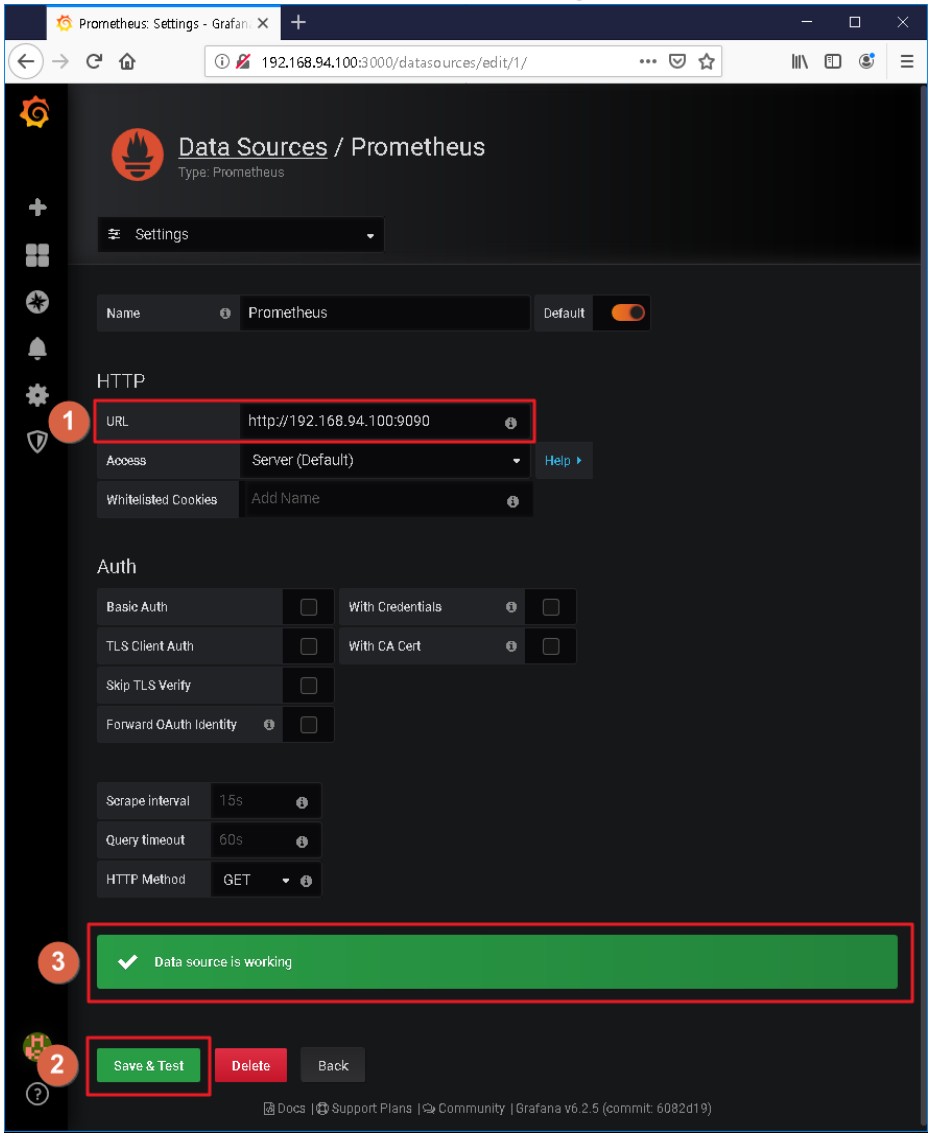
6.3 Grafana와 연동하여 시각화 설정

4. Prometheus 연동하기 위한 설정

4-1. “URL” 주소에 Prometheus 서버주소 및 포트번호 입력

4-2. “Save & Test”를 클릭하여 저장

4-3. “Data source is working”표시가 나오며 동작하는 것을 확인



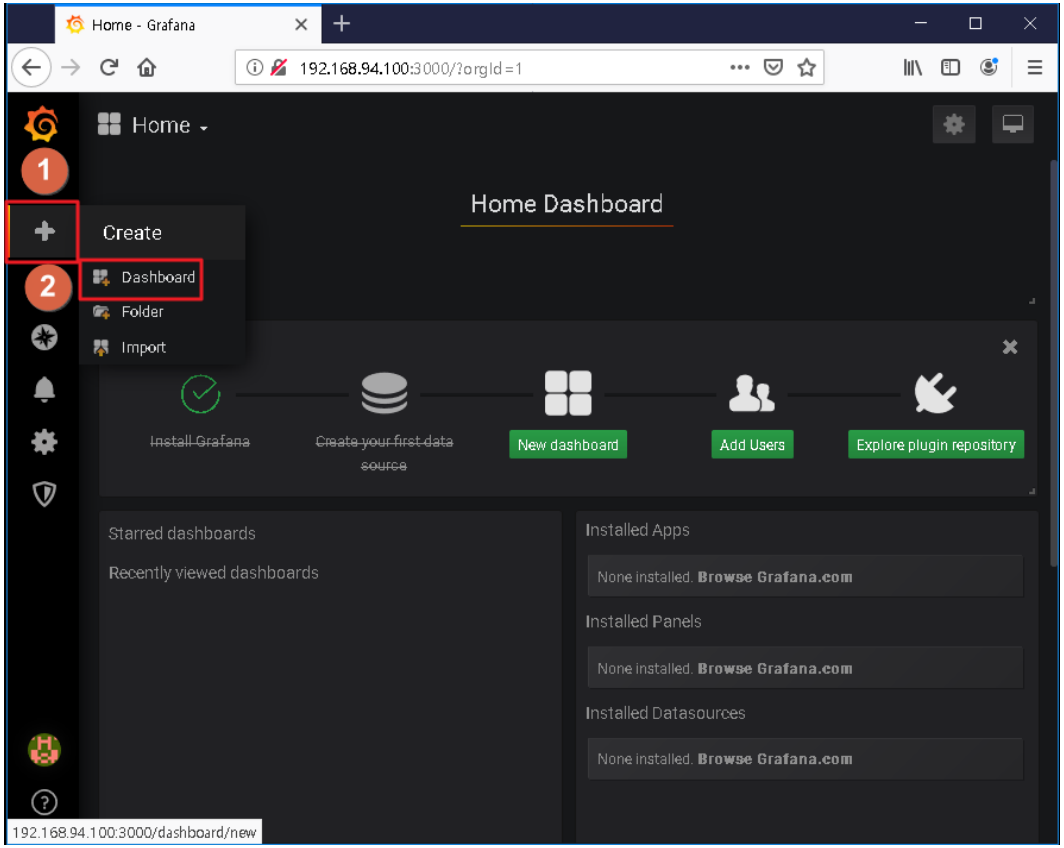
6. 활용예제

6.3 Grafana와 연동하여 시각화 설정

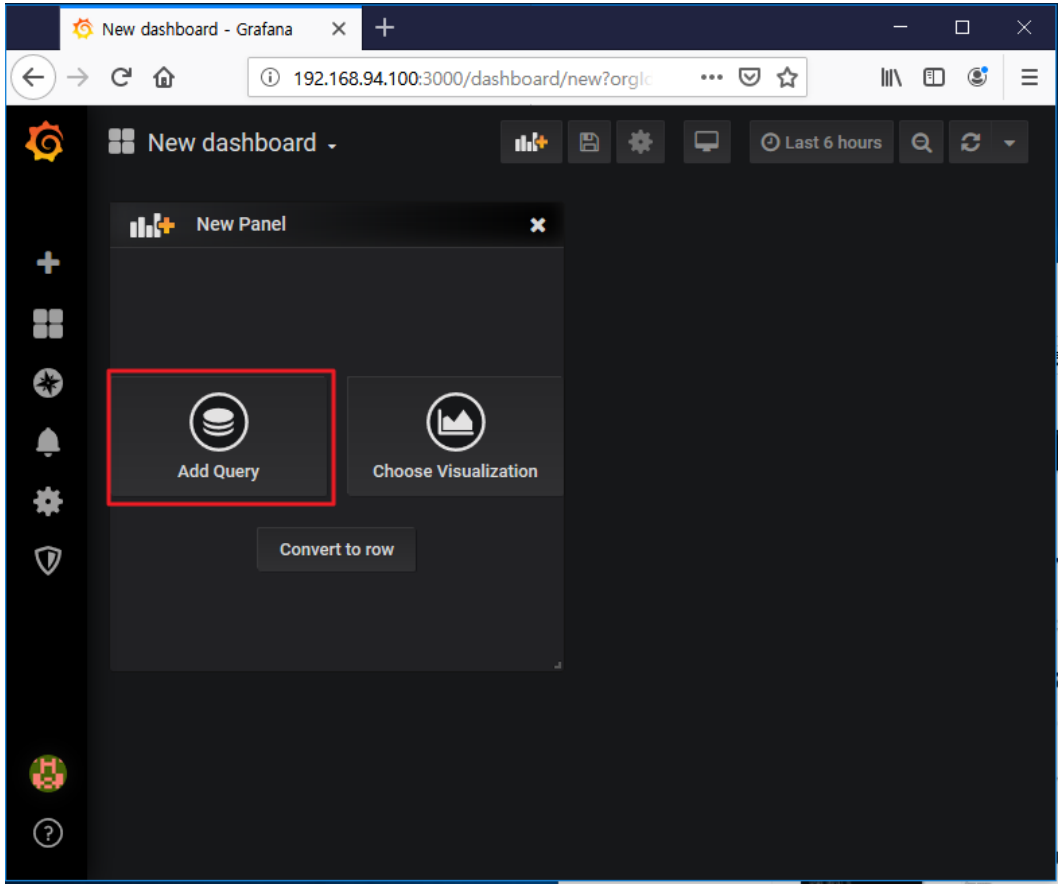
5. Dashboard 생성

5-1. “+” 를 클릭

5-2. “Dashboard”를 클릭



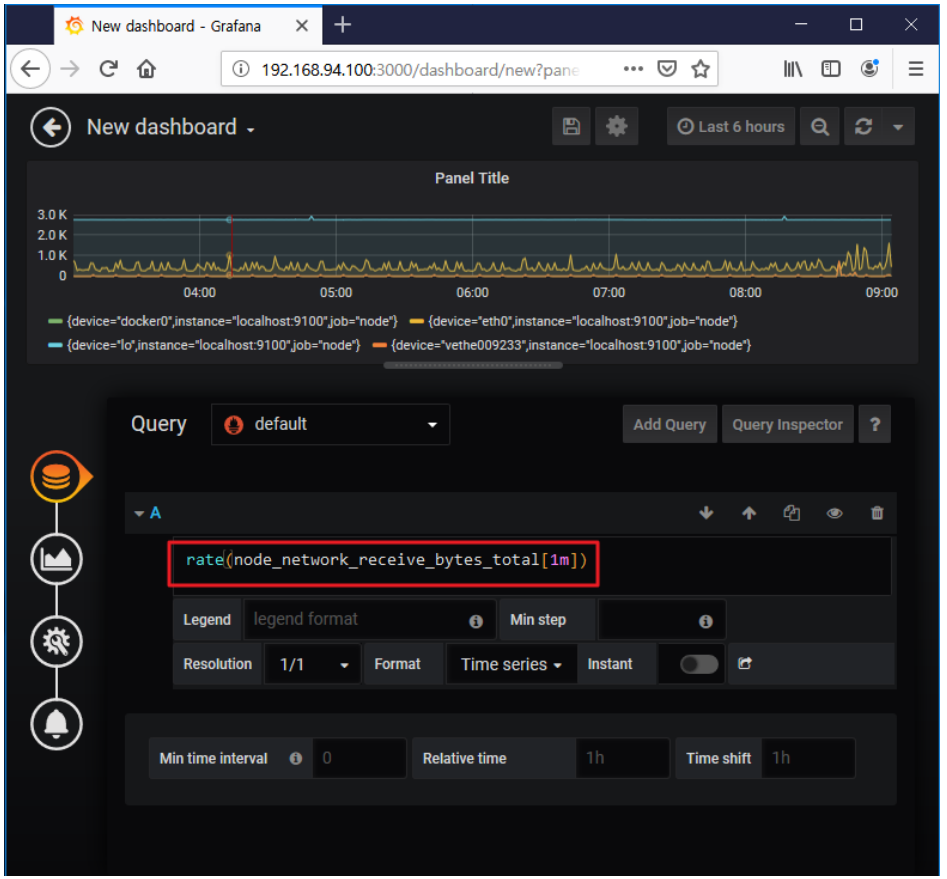
6. “Add Query” 클릭



6. 활용예제

6.3 Grafana와 연동하여 시각화 설정

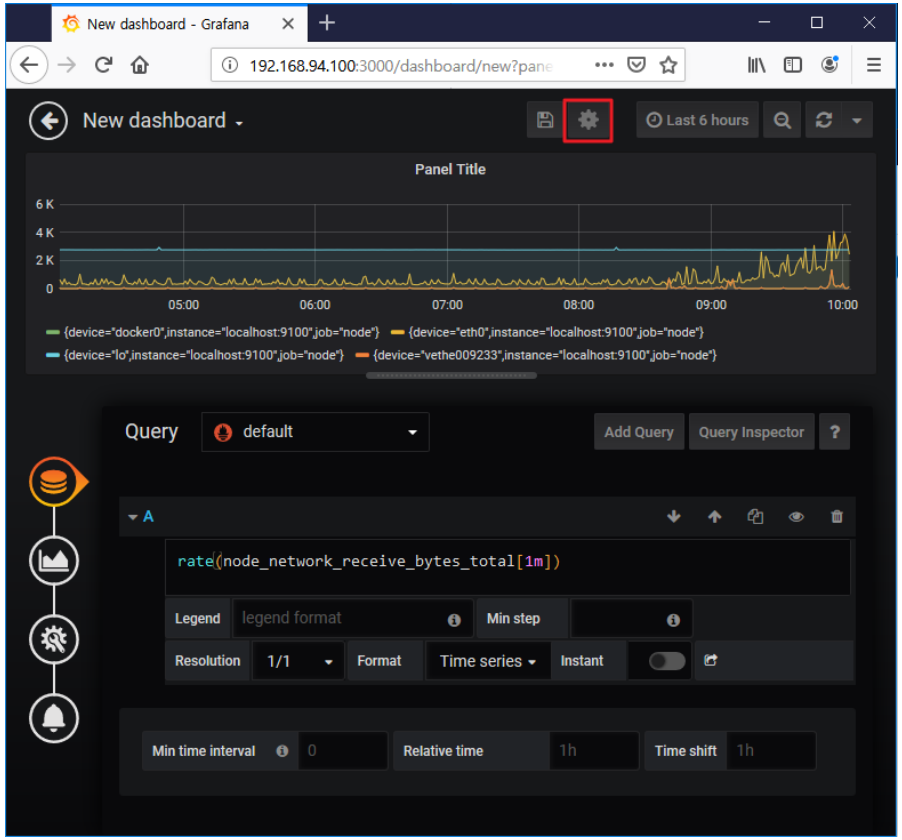
7. 입력창에 “rate(node_network_receive_bytes_total[1m])”를 입력하여 그래프가 정상적으로 생성되는지 확인

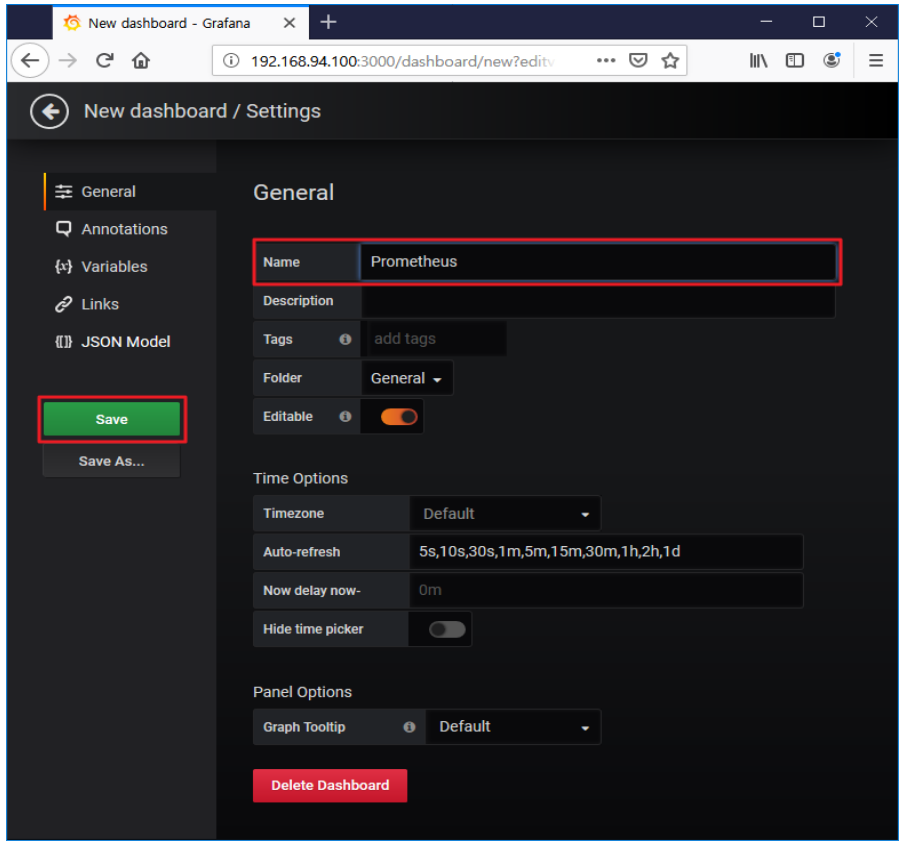


6. 활용예제

6.3 Grafana와 연동하여 시각화 설정

8. 톱니바퀴 모양을 클릭 후 대시보드의 이름을 설정

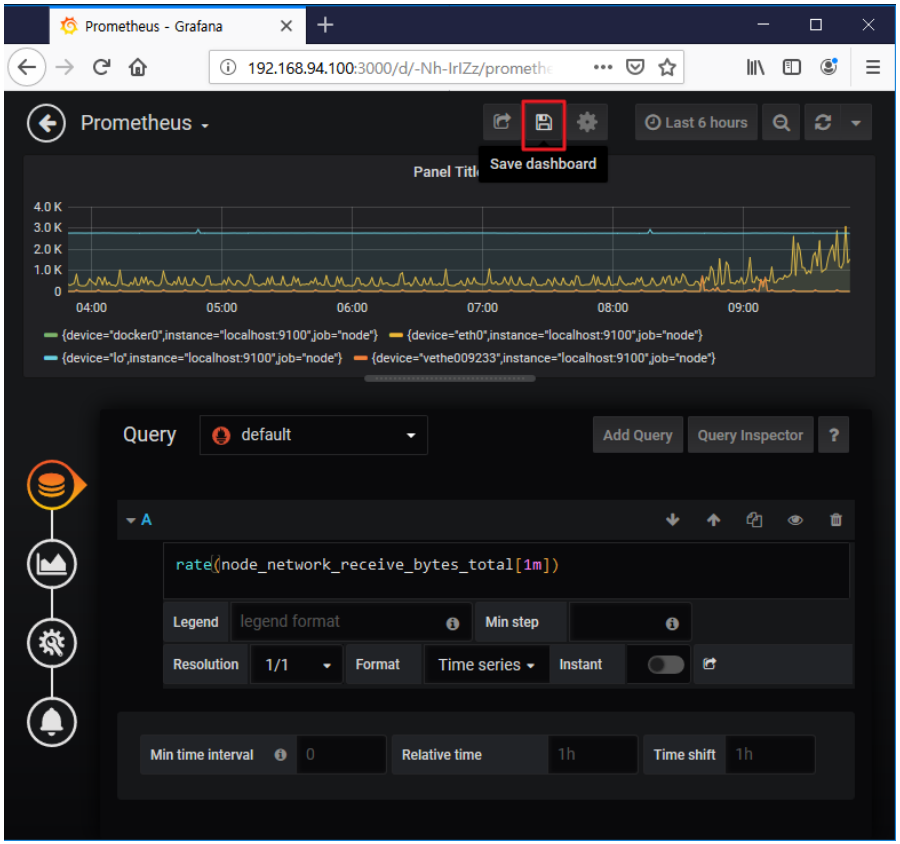


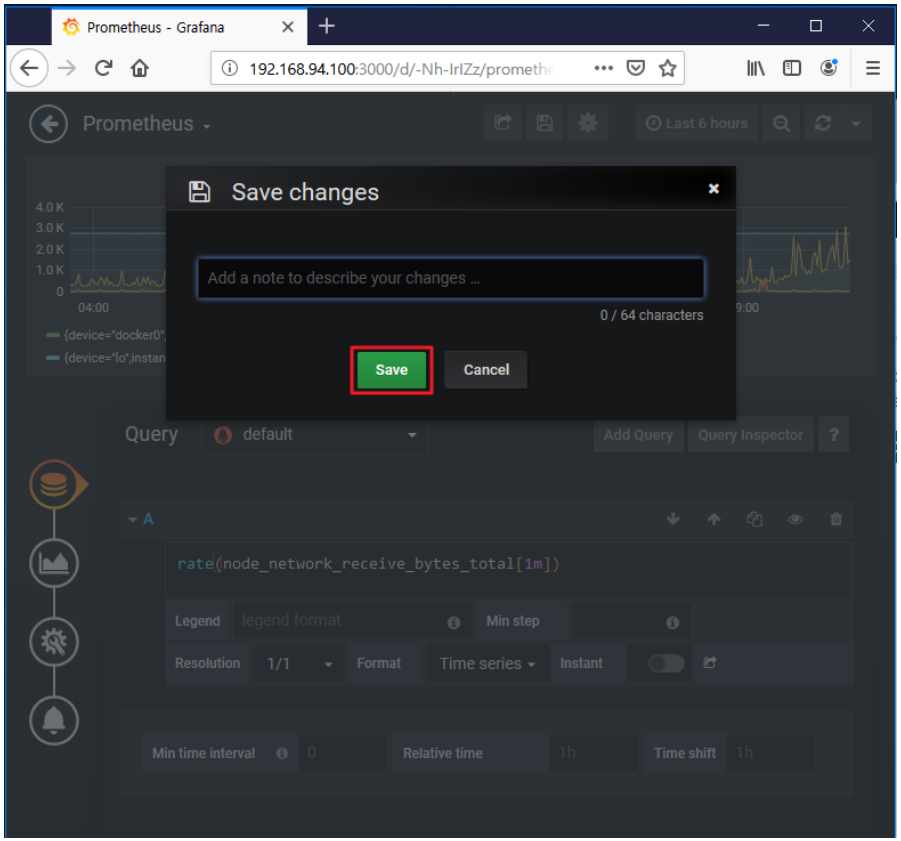


6. 활용예제

6.3 Grafana와 연동하여 시각화 설정

9. Save dashboard를 클릭하여 Dashboard를 저장.

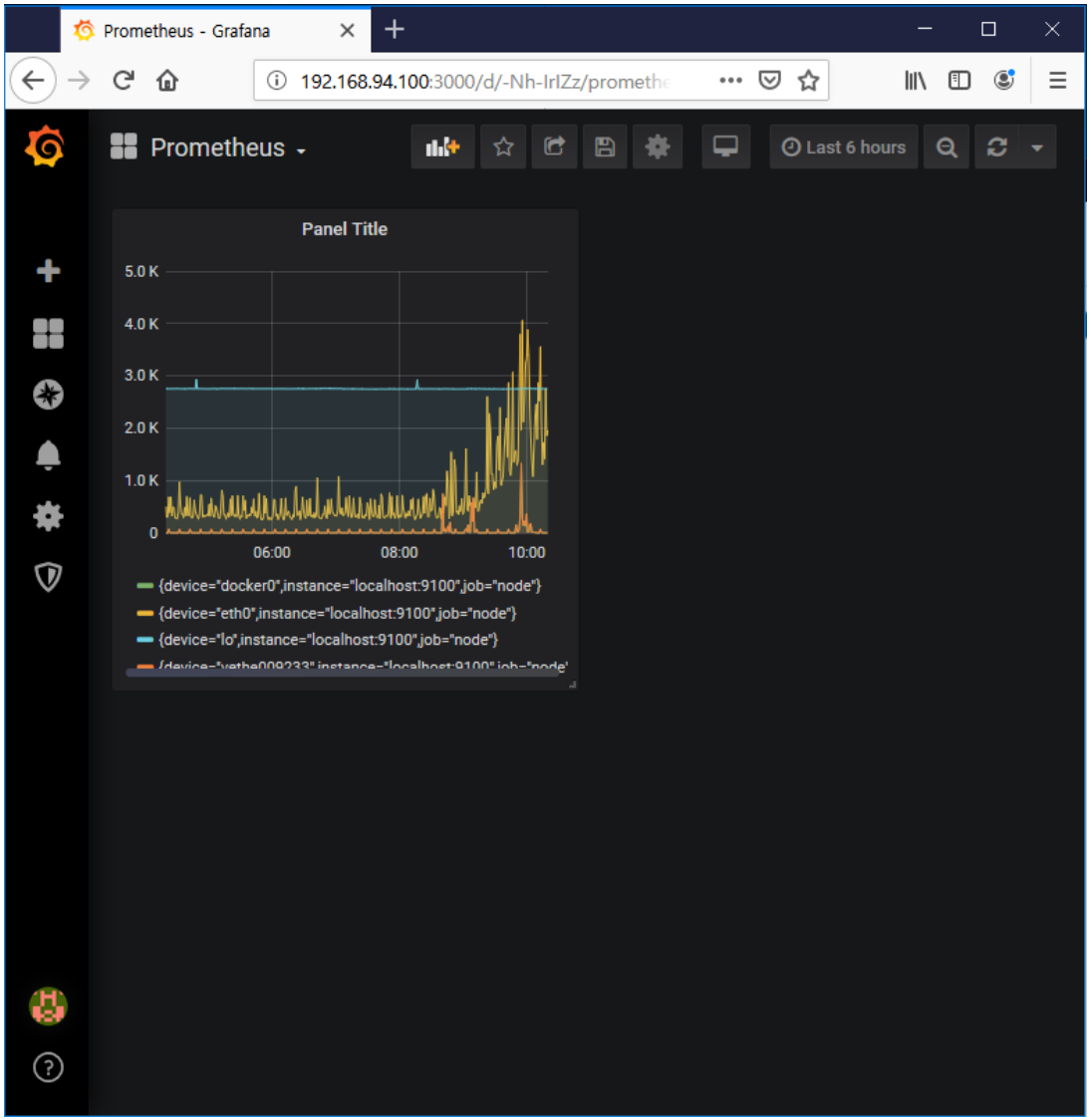




6. 활용예제

6.3 Grafana와 연동하여 시각화 설정

10. Prometheus와 Grafana 연동완료



7. FAQ

Prometheus

Q Prometheus를 통해서 로그를 수집할 수 있나요?

A Prometheus는 이벤트 로깅 시스템이 아니라 측정 기준을 수집하고 처리하는 시스템입니다. 로그를 수집하려면 ELK같은 솔루션을 사용해야 합니다.

Q Prometheus는 HA구성이 가능한가요?

A 동일한 Prometheus 서버를 여러대 실행하여 HA구성이 가능합니다.

7. FAQ

Prometheus

Q Alert를 보낼수 있나요?

A Alertmanager는 다음의 외부시스템을 Alert을 발송할수 있습니다.

Email, Generic Webhooks, HipChat, OpsGenie, PagerDuty,

Pushover, Slack

Q Prometheus는 Dashboard를 생성할수 있나요?

A Grafana를 통해 시각화를 할 수 있습니다.

8. 용어정리

용어	설 명
Metric	시간상 특정 지점에서 시스템의 일부측면을 설명하는 숫자 값. Log와는 달리 주기적으로 보내게 됩니다. Log는 보통 무언가가 발생했을 때 로그 파일에 추가합니다. 정기적으로 수집해서 시스템의 추이를 살펴볼수 있는 정보.
PromQL	Prometheus는 사용자가 실시간으로 시계열 데이터를 선택하고 집계 할 수있는 PromQL (Prometheus Query Language)이라는 기능적 쿼리 언어를 제공합니다.
TSDB	Time Series DBMS는 time series data를 처리하기 위해 최적화 된 데이터베이스 관리 시스템입니다.
Alertmanager	Alertmanager는 Prometheus 서버와 같은 클라이언트 응용 프로그램에서 보낸 경고를 처리합니다.
Expoter	필요한 매트릭을 수집하고 이를 HTTP 엔트포인트로 Prometheus에 노출하는 역할을 합니다.

Open Source Software Installation & Application Guide

nipa 공개SW역량프라자