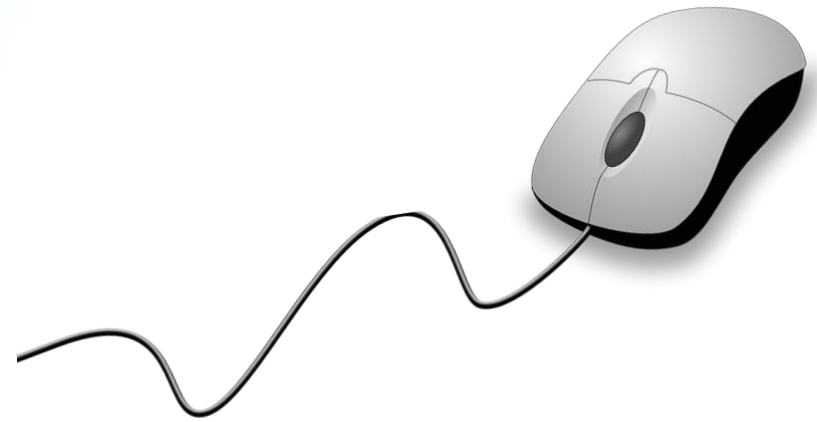


공개SW 솔루션 설치 & 활용 가이드

시스템SW > 데이터관리

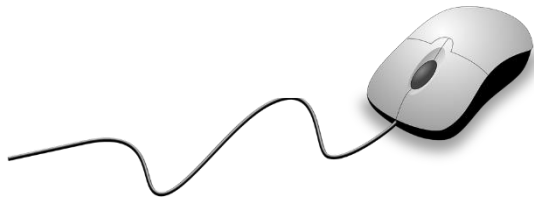


# 제대로 배워보자

How to Use Open Source Software

---

Open Source Software Installation & Application Guide



# CONTENTS

1. 개요
2. 실행환경
3. 설치 및 실행
4. 기능소개
5. FAQ

# 1. 개요



<b>소개</b>	<ul style="list-style-type: none"> <li>• InfluxData사의 Time-series DB: 시계열 데이터를 저장하고 활용하는데에 특화된 database</li> <li>• Timestamp 기반의 모든 데이터에 대한 백업 데이터 저장소로 활용</li> </ul>		
<b>주요기능</b>	<ul style="list-style-type: none"> <li>• 시계열 데이터를 위해 특별히 작성된 사용자 정의 고성능 데이터 스토어 TSM 엔진</li> <li>• 높은 수신 속도 및 데이터 압축을 허용</li> <li>• 전체적으로 Go 로 작성. 외부 종속성없이 단일 바이너리로 컴파일</li> <li>• 간단하고 고성능의 HTTP (S) API 작성 및 쿼리</li> <li>• 다양한 플러그인 제공 (다른 데이터 수집 프로토콜을 지원)</li> </ul>		
<b>대분류</b>	<ul style="list-style-type: none"> <li>• 시스템SW</li> </ul>	<b>소분류</b>	<ul style="list-style-type: none"> <li>• 데이터관리</li> </ul>
<b>라이선스형태</b>	<ul style="list-style-type: none"> <li>• MIT License</li> </ul>	<b>사전설치 솔루션</b>	<ul style="list-style-type: none"> <li>• 없음</li> </ul>
		<b>버전</b>	<ul style="list-style-type: none"> <li>• v1.12.0 (2019년 11월 기준)</li> </ul>
<b>특징</b>	<ul style="list-style-type: none"> <li>• influxdb와 collector인 telegraf를 통해 시계열 데이터를 적재</li> <li>• 오픈소스 대시보드인 grafana 를 통해 관제 대시보드 구성 및 알림 통보 (alert)를 구성</li> <li>• 특정 상황에서 사용자가 정의한 동작을 수행</li> </ul>		
<b>개발회사/커뮤니티</b>	<ul style="list-style-type: none"> <li>• InfluxData</li> </ul>		
<b>공식 홈페이지</b>	<ul style="list-style-type: none"> <li>• <a href="https://www.influxdata.com/">https://www.influxdata.com/</a></li> </ul>		

## 2. 실행환경



- 하드웨어 제약이 거의 없음
- OS 플랫폼 종류에 따른 지원
  - RPM 기반 리눅스 배포본 계열: Red Hat Enterprise Linux 6/7, Scientific Linux 6/7, CentOS 6/7, Fedora 27/28/29
- 요구사항
  - 메모리공간: 최소 8GB 이상
  - 디스크공간: 최소 100G 이상
  - 1000~3000 IOPS 수준의 볼륨 속도 지원

# 3. 설치 및 실행

세부 목차



## 3.1 설치 및 실행

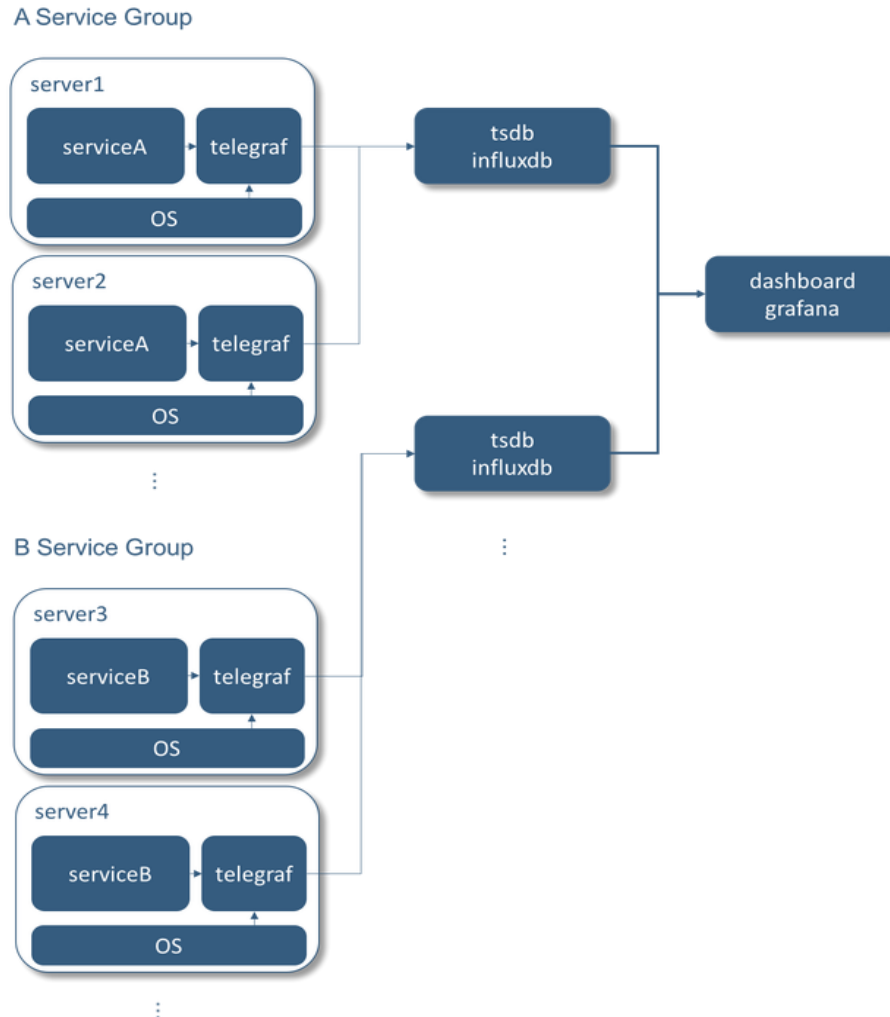


# 3. 설치 및 실행



## 세부 목차

### 3.1 서비스 그룹 아키텍처



# 3. 설치 및 실행



## 3.1 설치 및 실행

- OS Version : CentOS Linux release 7.5.1810  
Influxdb Version : influxdb-1.7.7.x86\_64.rpm

### 1. 패키지 매니저 업데이트

```
cat <<EOF | sudo tee /etc/yum.repos.d/influxdb.repo
[influxdb]
name = InfluxDB Repository - RHEL \${releasever}
baseurl = https://repos.influxdata.com/rhel/\${releasever}/\${basearch}/stable
enabled = 1
gpgcheck = 1
gpgkey = https://repos.influxdata.com/influxdb.key
EOF
```

# 3. 설치 및 실행



## 3.1 설치 및 실행

### 2. Yum을 통한 설치 진행

```
sudo yum install influxdb
sudo service influxdb start

# systemd(CentOS 7+, RHEL 7+)을 사용한다면..
sudo yum install influxdb
sudo systemctl start influxdb

Loaded plugins: extras_suggestions, langpacks, priorities, update-motd Examining
influxdb-1.7.7.x86_64.rpm: influxdb-1.7.7-1.x86_64 Marking influxdb-1.7.7.x86_64.rpm to
be installed Resolving Dependencies --> Running transaction check ---> Package
influxdb.x86_64 0:1.7.7-1 will be installed --> Finished Dependency Resolution amzn2-
core/2/x86_64 | 2.4 kB 00:00:00 Dependencies Resolved

=====
=====
== Package Arch Version Repository Size
=====
=====
== Installing: influxdb x86_64 1.7.7-1 /influxdb-1.7.7.x86_64 120 M Transaction Summary
=====
=====
== Install 1 Package
```





# 3. 설치 및 실행



## 3.1 설치 및 실행

### 3. InfluxDB 기동

Influxdb를 서비스를 통해 기동

```
서비스: /usr/lib/systemd/system/influxdb.service
[root@influxdb /etc/init.d]# systemctl enable influxdb
[root@influxdb /etc/init.d]# systemctl start influxdb
[root@influxdb /etc/init.d]# systemctl status influxdb
● influxdb.service - InfluxDB is an open-source, distributed, time series database
   Loaded: loaded (/usr/lib/systemd/system/influxdb.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2019-10-06 17:59:07 KST; 4s ago
     Docs: https://docs.influxdata.com/influxdb/
   Main PID: 27423 (influxd)
   CGroup: /system.slice/influxdb.service
           └─27423 /usr/bin/influxd -config /etc/influxdb/influxdb.conf

Oct 06 17:59:07 influxdb influxd[27423]: ts=2019-10-06T08:59:07.730023...m
Oct 06 17:59:07 influxdb influxd[27423]: ts=2019-10-06T08:59:07.730033...t
Oct 06 17:59:07 influxdb influxd[27423]: ts=2019-10-06T08:59:07.730043...r
Oct 06 17:59:07 influxdb influxd[27423]: ts=2019-10-06T08:59:07.730053...e
Oct 06 17:59:07 influxdb influxd[27423]: ts=2019-10-06T08:59:07.730060...r
Oct 06 17:59:07 influxdb influxd[27423]: ts=2019-10-06T08:59:07.730129...e
Oct 06 17:59:07 influxdb influxd[27423]: ts=2019-10-06T08:59:07.730161...m
Oct 06 17:59:07 influxdb influxd[27423]: ts=2019-10-06T08:59:07.730307...0
Oct 06 17:59:07 influxdb influxd[27423]: ts=2019-10-06T08:59:07.730612...s
Oct 06 17:59:07 influxdb influxd[27423]: ts=2019-10-06T08:59:07.730744...0
Hint: Some lines were ellipsized, use -l to show in full.
[root@influxdb /etc/init.d]#
```



# 3. 설치 및 실행



## 3.1 설치 및 실행

### 4. 설치 후 정상 기동 테스트 확인

Curl 명령어를 통하여 서비스에 대한 정상 기동여부를 확인함

```
[root@ip-172-31-10-88 ~]# curl "http://localhost:8086/query?q=show+databases"
{"results":[{"statement_id":0,"series":[{"name":"databases","columns":["name"],"values":[["_internal"]]}]}]}
[root@ip-172-31-10-88 ~]#
```

[웹브라우저 확인]



# 3. 설치 및 실행



## 3.1 설치 및 실행

### 5. 샘플 데이터베이스를 생성

influx 명령어를 통하여 서비스에 대한 정상 기동여부를 확인함

```
[root@ip-172-31-10-88 ~]# influx
Connected to http://localhost:8086 version 1.12.0
InfluxDB shell version: 1.12.0
> create database sample
> use sample
Using database sample
>
```

curl 명령어를 통하여 생성된 데이터베이스를 확인

```
http://52.78.85.168:8086/query?q=show+databases
{"results":[{"statement_id":0,"series":[{"name":"databases","columns":["name"],"values":["_internal"],["sample"]}]}]}
```

[웹브라우저 확인]



# 4. 기능소개

## 세부 목차

- 4.1 RDB와 InfluxDB 맵핑
- 4.2 클라이언트 명령어
- 4.3 데이터베이스 생성 및 데이터 입력
- 4.4 테이블 조회 및 데이터 조회
- 4.5 Continuous Query(CQ)
- 4.6 Grafana 연동한 시각화 기능



# 4. 기능소개



## 4.1 RDB와 InfluxDB의 기능 맵핑

MySQL과 같은 RDB 개념과 InfluxDB의 맵핑된 개념

RDB 개념	influxDB 개념
database	database
table	measurement
column	key
PK or Indexed column	tag key(only string)
unindexed column	field key
SET of index entries	series

# 4. 기능소개



## 4.2 클라이언트 명령어

클라이언트에서 많이 사용되는 명령어

명령어	설명
use [database 이름]	현재 데이터베이스 지정
show measurements	measurement(테이블) 목록 보기
show tag keys	모든 measurement의 모든 tag key 목록 보기
show field keys	모든 measurement의 모든 field key 목록 보기
show series	모든 measurement의 서로소인 tag key 튜플의 목록 보기



# 4. 기능소개



## 4.3 데이터베이스 생성 및 데이터 입력

Database 생성

```
create database ui_monitoring_new  
> use ui_monitoring_new  
Using database ui_monitoring_new
```

InfluxDB는 테이블이란 용어 대신 'measurements' 라는 개념을 사용한다. measurement는 별도로 생성할 필요는 없기 때문에 곧바로 데이터를 insert 하면 measurement를 사용하는 것이 가능

데이터 입력

```
insert ui_monitoring_new,zon_type=ap,pagetype=hoem,site_cd=sg,page_url=/sg/  
result=true,resultval=0.001  
> insert ui_monitoring_new,zon_type=ap,pagetype=hoem,site_cd=sg,page_url=/sg/  
result=false, resultval=1
```

# 4. 기능소개

## 4.4 테이블 조회 및 데이터 조회

Node JS를 통한 테이블 조회

```
show measurements;  
name: measurements  
name  
----  
ui_monitoring_new
```

데이터 조회

```
SELECT * FROM "ui_monitoring_new" LIMIT 1
```





# 4. 기능소개



## 4.5 Continuous Query(CQ)

```
CREATE CONTINUOUS QUERY cq_1min_mysql ON test_database
BEGIN
  select
    mean(commands_delete) as commands_delete, mean(commands_insert) as commands_insert,
    mean(commands_update) as commands_update, mean(commands_select) as commands_select
  INTO Retention_5y_ds1min.mysql          /* Retention_5y_ds1min 은 미리 정의된 Retention Policy 이름 */
  FROM realtime.mysql                    /* realtime 도 미리 정의된 Retention Policy 이름 */
  GROUP BY time(1m),host,"server"       /* 다운샘플링 1분, TAG 는 GROUP BY 절에 이어서 붙여 줌. 예약어
                                         (server)는 쌍 따옴표로 감싸줌 */
END
```

### # 생성된 CONTINUOUS QUERY 확인

```
SHOW CONTINUOUS QUERIES
```

1초 마다 체크하여 다운샘플링 결과를 따로 저장. Background 로 자동 수행되며 체크 주기는 설정 가능. (/etc/influxdb/influxdb.conf)  
Oracle database 의 MVIEW 와 유사한 개념.

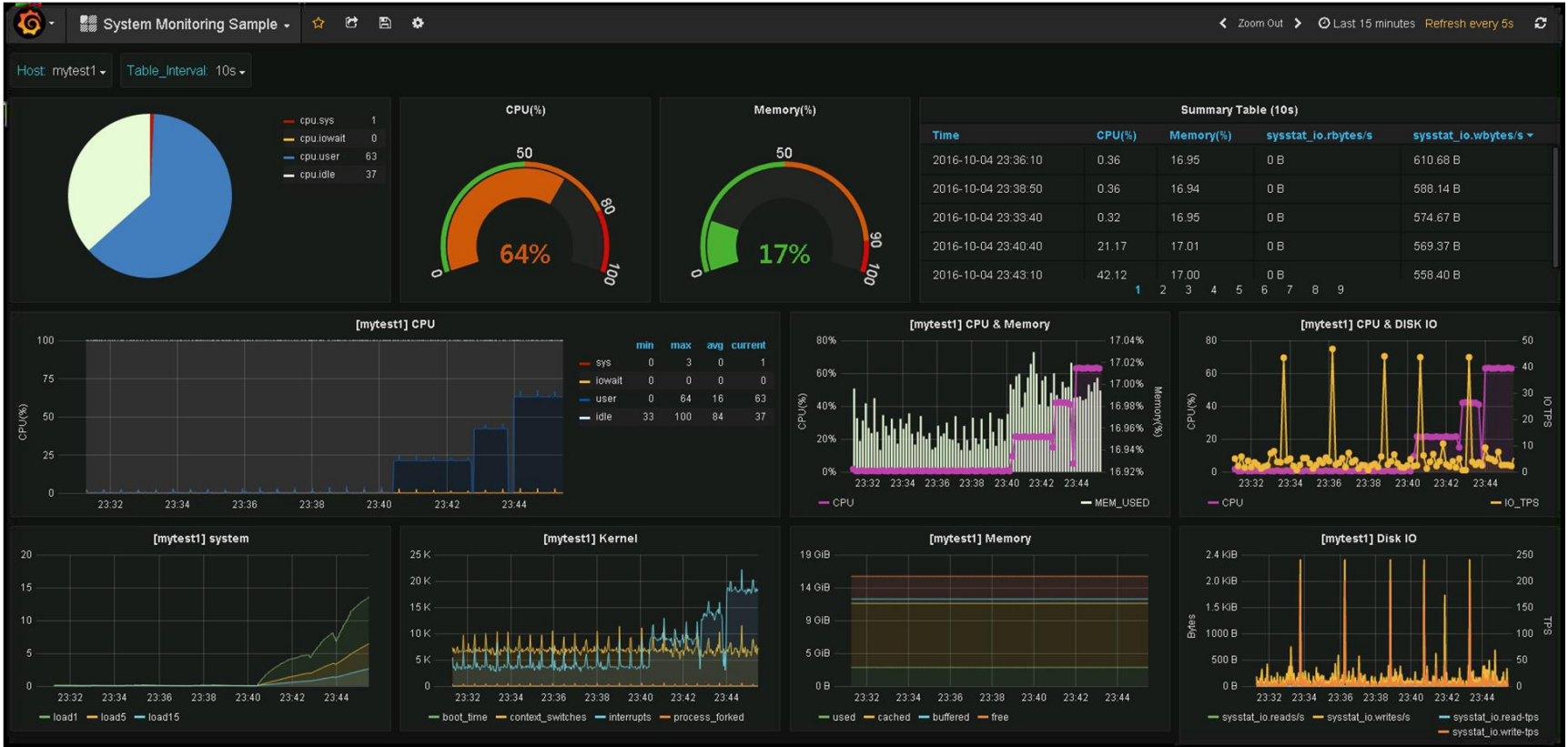


# 4. 기능소개



## 4.6 Grafana와 연동한 시각화 기능

Grafana와 연동하여 시계열 데이터에 대한 시각화 가능





**Q** InfluxDB를 Prometheus 차이는 무엇인가요?

**A** 다음과 같은 차이점이 있습니다.

### 데이터 수집 방식: Push vs. Pull

InfluxDB는 push 기반의 시스템이다. InfluxDB에 데이터를 쌓기 위해서는 누군가 InfluxDB로 데이터를 넣어줘야 한다. 어플리케이션 서버가 직접 api를 호출해 데이터를 넣거나, [fluentd](#) 등을 이용해 데이터를 넣어줘야 한다.

Prometheus는 pull 기반의 시스템이다. 데이터를 가져 갈 수 있는 통로를 열어 놓고 이를 prometheus 서버의 설정에 넣어주면 알아서 데이터를 가져 간다. 다양한 클라이언트 라이브러리들을 통해 데이터를 수집하고 이를 prometheus가 가져갈 수 있도록 포트나 api를 열어 주면 된다. 지표 수집 서버보다는 실제 서비스 서버가 훨씬 대수가 많은 것이 보통이다. Push 방식에서 각 서비스 서버의 데이터 수집 옵션을 수정하고 배포하는 것보다, 소수의 지표 수집 서버의 설정만 바꿔주면 되는 pull 방식이 더 효율적이다.

### 데이터 저장 방식

InfluxDB는 metric value와 이것의 index들이 모두 database로 관리된다.

Prometheus는 index에 대해서만 디비로 관리하고 metric value들은 file로 관리 된다.

이 때문에 속도와 디스크 효율 면에서 prometheus가 더 나은 모습을 보인다. 지표 수집은 필요로 하는 key에 대한 데이터 위치를 찾고 이를 가져가기만 하면 되는데, InfluxDB가 metric value도 db로 관리하는 것은 오버헤드에 가깝기 때문이다. 대신, InfluxDB의 이런 특징은 지표 모니터링 보단 이벤트 로깅에서 더 강점을 보인다.

# Open Source Software Installation & Application Guide

**nipa** 공개SW역량프라자



이 저작물은 크리에이티브 커먼즈 [저작자표시-비영리-동일조건 변경허락 2.0 대한민국 라이선스]에 따라 이용하실 수 있습니다.