

# MOS Cloud 운용 가이드

## Installation and Operation

2023.03.29

# Contents

1. MOS Cloud Installation
2. MOS Cloud Setting





# Part 01

## MOS Cloud Installation

# Installation

- /opt/install 디렉토리의 01~15 설치 쉘 스크립트를 순서대로 실행합니다.
- 해당 디렉토리에 있는 install\_all.sh 스크립트로도 설치 가능하나, 설치 시 발생하는 오류 확인을 위해 순서대로 실행하는 것을 권장합니다.
- 실행 예

```
root@ubtmos:/opt/install# ./01_install.sh
```

- 설치 쉘 스크립트

```
root@ubtmos:/opt/install# ls
01_install.sh          05_install_rabbitmq-c.sh  09_install_Lynis.sh    13_install_certbot.sh
02_install_rabbitmq.sh 06_install_python.sh     10_install_mariadb.sh  14_install_systemctl_service.sh
03_install_cjson.sh    07_install_python_package.sh 11_install_grafana.sh  15_install_default_config.sh
04_install_open62541.sh 08_install_influx.sh      12_install_nginx.sh    configs
```

- 시스템 Timezone 설정
  - sudo ln -sf /usr/share/zoneinfo/Asia/Seoul /etc/localtime

# Installation

## ▪ 주요 패키지 버전 확인

- MariaDB : 10.5.9
- InfluxDB : 2.6.1
- Grafana : 8.3.2
- Python : 3.8.10

```
root@ubtmos:/opt/install# mariadb --version
mariadb Ver 15.1 Distrib 10.5.9-MariaDB, for debian-linux-gnu (x86_64) using readline 5.2
root@ubtmos:/opt/install# influxd version
InfluxDB v2.6.1 (git: 9dcf880fe0) build_date: 2022-12-29T15:53:07Z
root@ubtmos:/opt/install# influx version
Influx CLI 2.6.1 (git: 61c5b4d) build_date: 2022-12-29T15:41:09Z
```

```
root@ubtmos:/opt/install# grafana-cli -v
Grafana CLI version 8.3.2
root@ubtmos:/opt/install# grafana-server -v
Version 8.3.2 (commit: afb9e8e5f, branch: HEAD)
```

```
root@ubtmos:/opt/install# python3 --version
Python 3.8.10
```

# Part 02

## MOS Cloud Setting

# Settings

## ■ OPCUA 데이터 수집 설정

- AAS로부터 변환한 syscfg.json, engineering.csv, nodeset.xml 파일을 /opt/cfg/ 디렉토리에 위치

```
root@ubtmos:/opt/cfg# ls -al
total 188
drwxr-xr-x 4 root root 4096 Mar 29 05:54 .
drwxr-xr-x 7 root root 4096 Mar 29 04:23 ..
drwxr-xr-x 2 root root 4096 Apr 12 2021 aas
-rw-r--r-- 1 root root 13168 Mar 29 05:54 engineering.csv
-rw-r--r-- 1 root root 19178 Mar 29 05:50 glances.conf
-rw-r--r-- 1 root root 100 Mar 27 07:57 monitoring.cfg
-rwxr-xr-x 1 root root 118467 Mar 29 05:54 nodeset.xml
-rw-r--r-- 1 root root 1002 Mar 27 06:04 opcua_cert.der
-rw----- 1 root root 1190 Mar 27 06:04 opcua_key.der
-rw-r--r-- 1 root root 359 May 26 2021 regi.json
-rw-r--r-- 1 root root 0 Mar 27 07:57 routing.csv
drwxr-xr-x 2 root root 4096 Mar 27 07:57 security
-rwxr-xr-x 1 root root 486 Mar 29 05:54 syscfg.json
```

- 정확한 정보 입력 후 systemctl restart gather 명령어로 데이터 수집 에이전트 재실행
- 아래 명령어로 데이터 수집 에이전트(gather), 데이터 저장 에이전트(itsdb) 로그 확인
  - tail -f /opt/log/gather\_\*\*\*\*\*.log
  - tail -f /opt/log/itsdb\_\*\*\*\*\*.log

```
root@ubtmos:/opt/cfg# tail -f /opt/log/gather_20230329.log
2023/03/29 05:58:20:930 [1] gathering process (CPS) opcua = 250, amqp = 250
2023/03/29 05:58:25:930 [1] gathering process (CPS) opcua = 250, amqp = 250
2023/03/29 05:58:30:930 [1] gathering process (CPS) opcua = 250, amqp = 250
2023/03/29 05:58:35:930 [1] gathering process (CPS) opcua = 250, amqp = 250
2023/03/29 05:58:40:930 [1] gathering process (CPS) opcua = 250, amqp = 250
2023/03/29 05:58:45:931 [1] gathering process (CPS) opcua = 250, amqp = 250
2023/03/29 05:58:50:931 [1] gathering process (CPS) opcua = 250, amqp = 250
```

# Settings

## ■ MariaDB 기본 설정 (1)

- mysql\_secure\_installation 명령어 실행
- password : mos 로 설정

```
root@ubtmos:/opt# mysql_secure_installation
Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
... Success!

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

Remove anonymous users? [Y/n] n
... skipping.

Disallow root login remotely? [Y/n] n
... skipping.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reload privilege tables now? [Y/n] y
... Success!
```



# Settings

## ■ MariaDB 기본 설정 (2)

- mysql -u root -p 명령어 실행
- DB 진입 후 아래 명령어 순서대로 입력  
create schema grafana;  
source /opt/install/smartfactory.sql  
show databases;

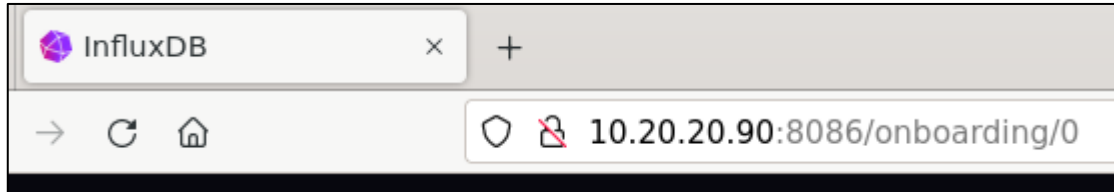
```
MariaDB [smartfactory]> show databases;  
+-----+  
| Database |  
+-----+  
| grafana |  
| information_schema |  
| mysql |  
| performance_schema |  
| smartfactory |  
+-----+  
5 rows in set (0.001 sec)
```

```
grant all privileges on smartfactory.* to 'smartfactory'@'localhost' identified by 'sfPassword123!@#';  
flush privileges;  
exit;
```

# Settings

## ■ InfluxDB 기본 설정

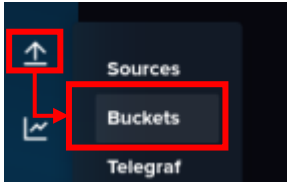
- 웹 브라우저를 실행하여 8086 포트 접속하여 기본 설정 정보 입력

A screenshot of the 'Setup Initial User' form in the InfluxDB onboarding interface. The form has a dark background with white text. It includes fields for Username, Password, Confirm Password, Initial Organization Name, and Initial Bucket Name. The Username field contains 'mos\_influx'. The Password and Confirm Password fields are masked with dots. The Initial Organization Name field contains 'mos\_org'. The Initial Bucket Name field contains 'mos'. There is a small information icon next to the Initial Organization Name and Initial Bucket Name labels. Below the form, there is a note: 'You will be able to create additional Users, Buckets and Organizations later'.

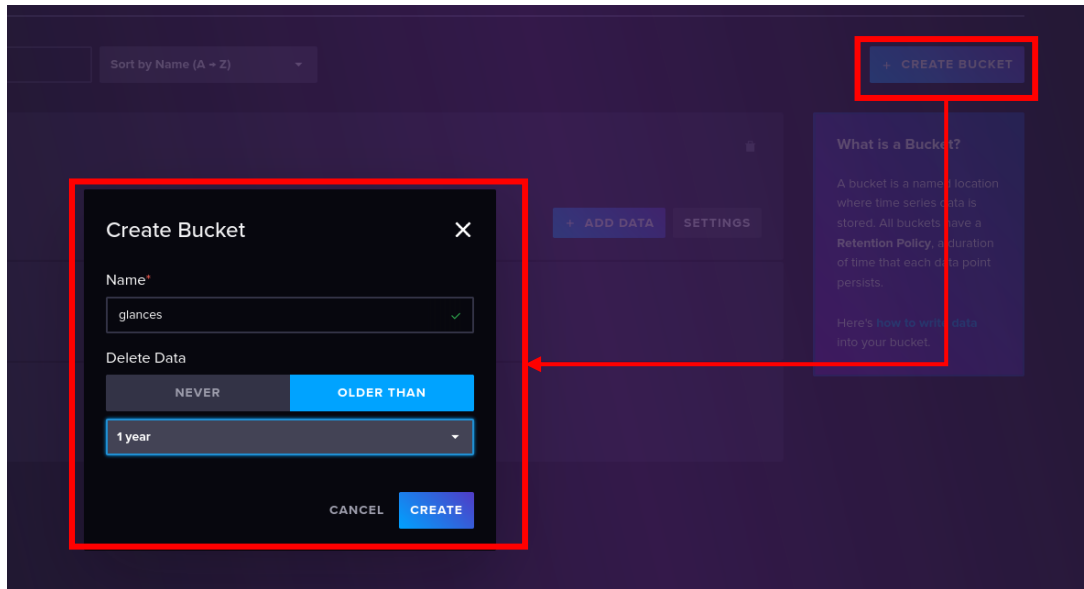
# Settings

## ■ InfluxDB 기본 설정

- 좌측 메뉴 버튼 – Buckets 항목 선택



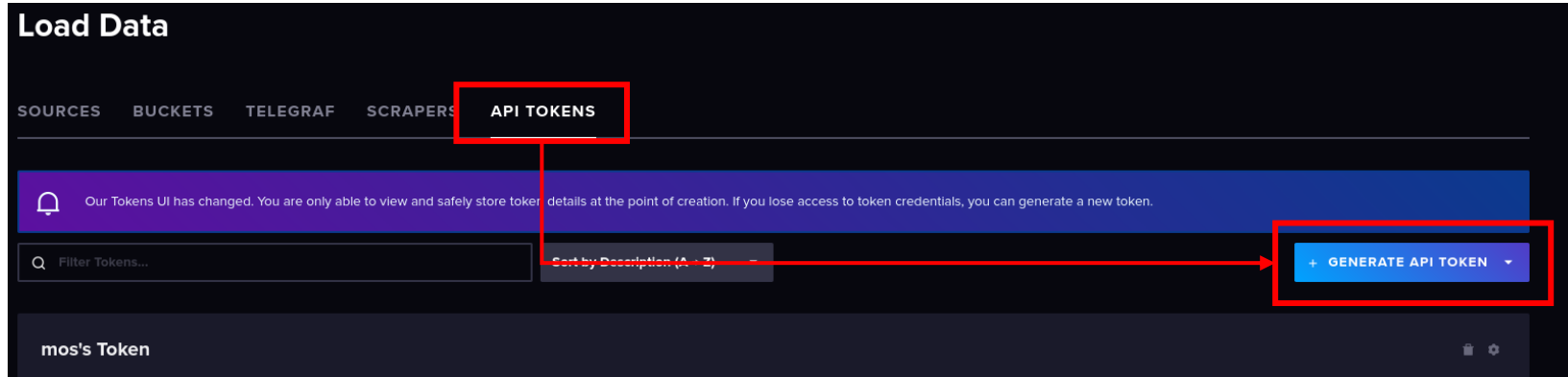
- 우측 상단 CREATE BUCKET 버튼을 눌러 “glances” Bucket 생성



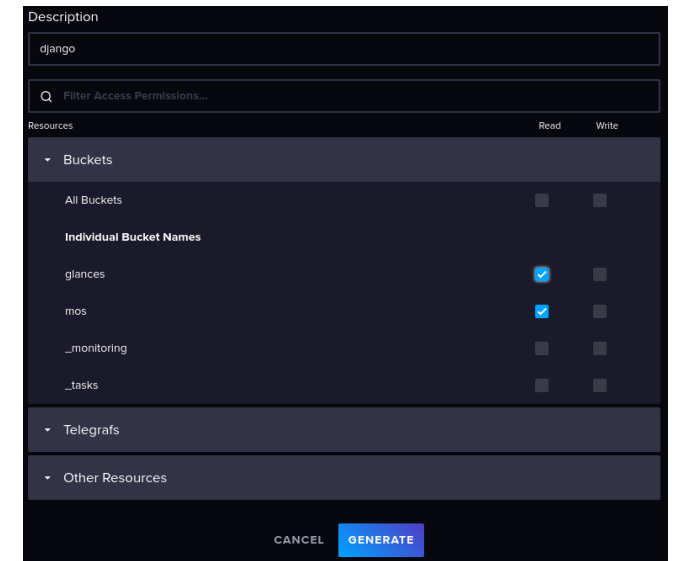
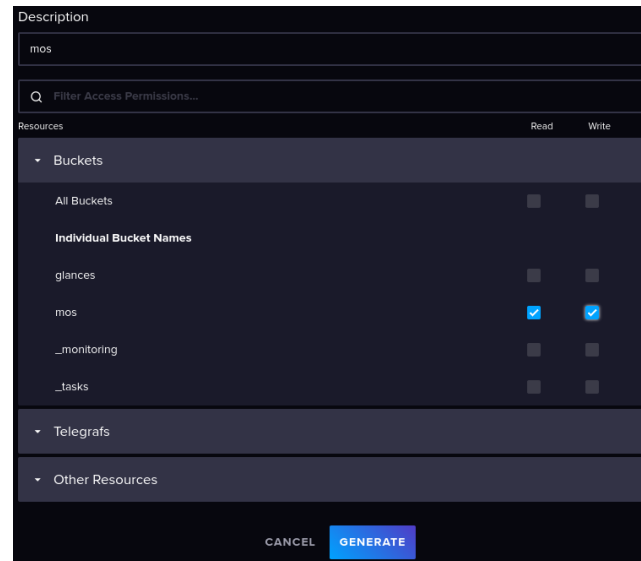
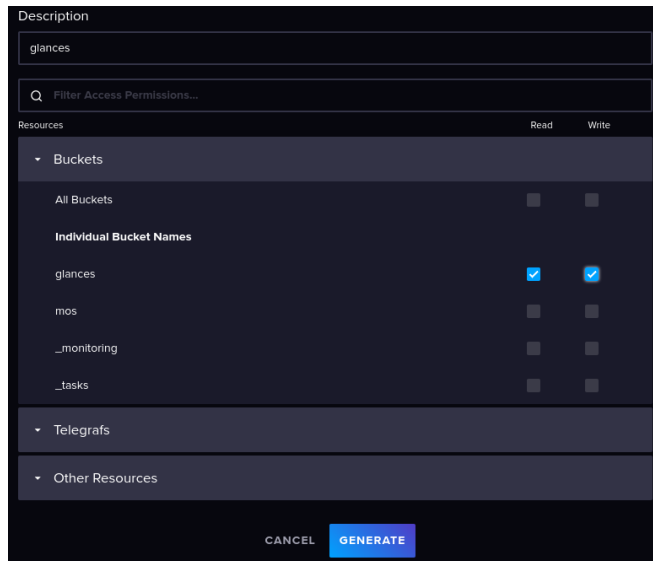
# Settings

## ■ InfluxDB 기본 설정

- 상단 API TOKENS – GENERATE API TOKEN 버튼을 클릭하여 TOKEN 생성



- Glances, MOS, Django 각각 토큰 생성 (총 3가지, 생성된 토큰은 반드시 기록/저장)





# Settings

## ■ Glances 설정

- cd /opt/cfg 명령어로 디렉토리 이동
- glances.conf 파일의 influxdb2 항목 수정

```
[influxdb2]
# Configuration for the --export influxdb2 option
# https://influxdb.com/
host=localhost
port=8086
protocol=http
org=
bucket=
token=
```

```
[influxdb2]
# Configuration for the --export influxdb2 option
# https://influxdb.com/
host=localhost
port=8086
protocol=http
org=mos_org
bucket=glances
token=1MvT_GD4HMoc1s1k1KokcPsggShXCp6-M47MvwdpANwvp8w4saBNHDqCGh85GbrmD8LE2yp0pJLCxwrLQfxi-A==
# One fix will be added for all measurement name
```

- systemctl restart glances 명령어로 서비스 재시작

# Settings

## ■ itsdb 설정

- /usr/lib/systemd/system/itsdb.service 파일 수정
- TSDB\_ORG, TSDB\_BUCKET, TSDB\_TOKEN 항목 입력

```
[Unit]
Description=influxdb tsdb service

[Service]
Environment="AMQP_USER=app_itsdb"
Environment="AMQP_PWD=ait24680!"
Environment="TSDB_ORG="
Environment="TSDB_BUCKET="
Environment="TSDB_TOKEN="

Type=simple
ExecStart=/opt/bin/itsdb.sh
Restart=on-failure

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

```
[Unit]
Description=influxdb tsdb service

[Service]
Environment="AMQP_USER=app_itsdb"
Environment="AMQP_PWD=ait24680!"
Environment="TSDB_ORG=mos_org"
Environment="TSDB_BUCKET=mos"
Environment="TSDB_TOKEN=bT384roPKnfTia1QE7vrk0I20h07DgOkNLvtVfk-SM-qIFPL-THucWtDA7499o-b12yWUMZeRGGiJqA-ggJvBA=="

Type=simple
ExecStart=/opt/bin/itsdb.sh
Restart=on-failure

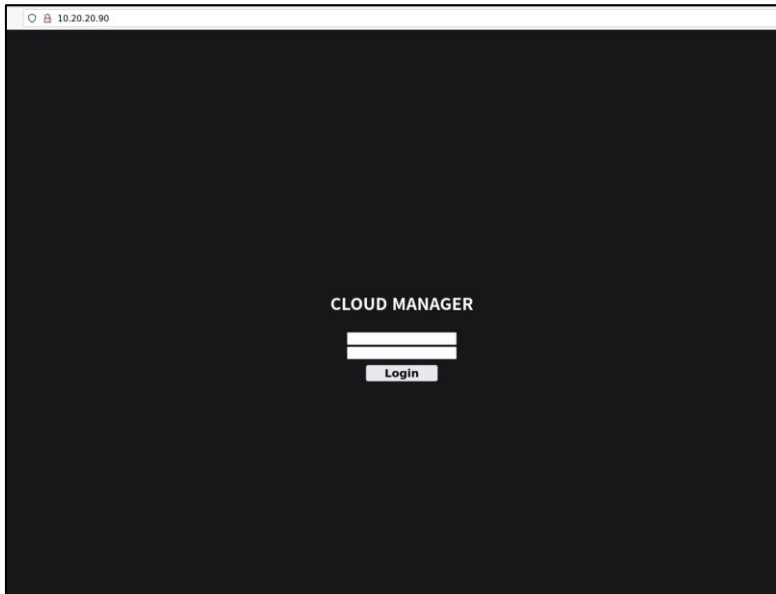
[Install]
WantedBy=multi-user.target
```

- systemctl daemon-reload 명령어로 설정사항 반영
- systemctl restart itsdb 명령어로 프로세스 재실행

# Settings

## ▪ AAS Web Service 기본 설정

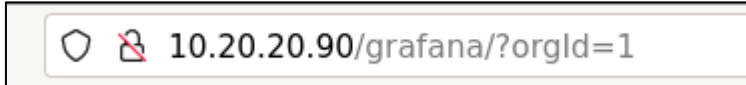
- cd /opt/apps/django/SmartFactory/ 명령어로 설정파일 디렉토리 진입, 아래 명령어를 입력하여 환경변수 설정
  - export TSDB\_ORG=1
  - export TSDB\_TOKEN=1
  - export TSDB\_BUCKET=1
- python manage.py createsuperuser 명령어로 웹서비스 관리자 계정 생성 (기본 ID : admin, PW : admin 으로 생성)
- 아래 명령어로 웹 서비스 재시작
  - systemctl restart django
  - systemctl restart nginx
- 브라우저를 실행하여 AASX Package Browser 실행 확인



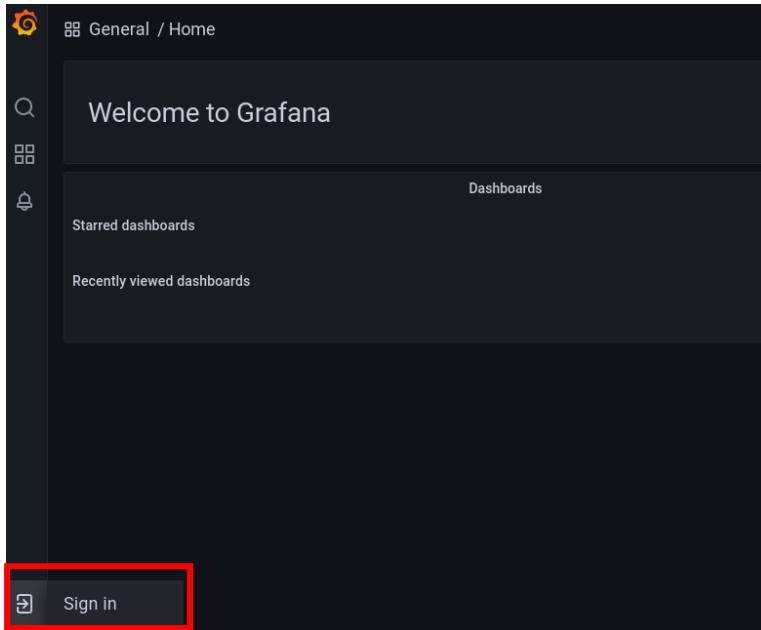
# Settings

## ■ AAS Web Dashboard 설정

- 웹브라우저에서 [웹서비스 URL]/grafana/ 입력하여 대시보드 서비스 페이지 이동



- 좌측 하단 Sign in 버튼으로 로그인 (기본 ID : admin , PW : admin)

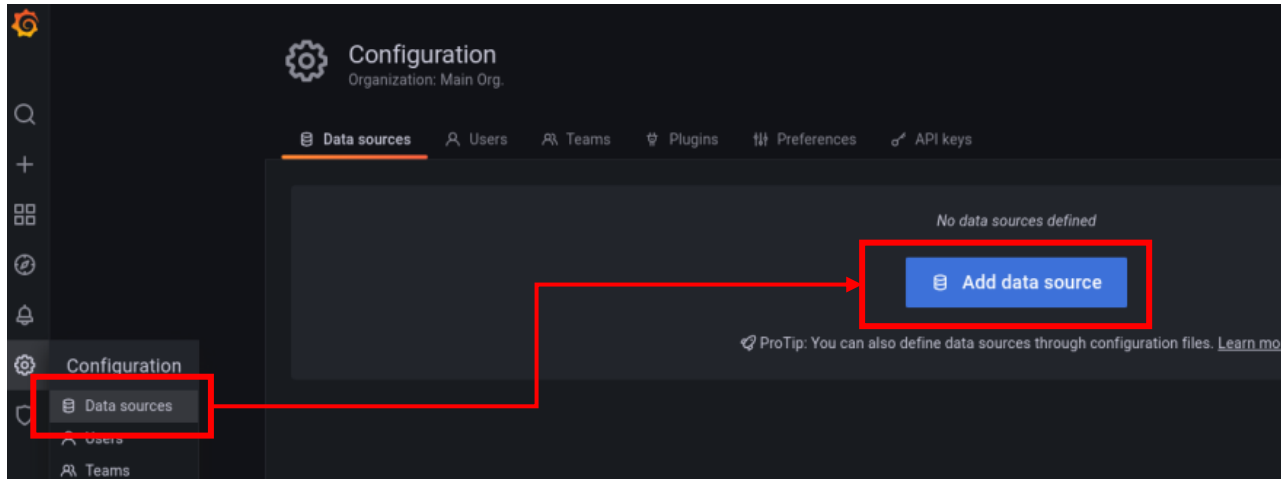




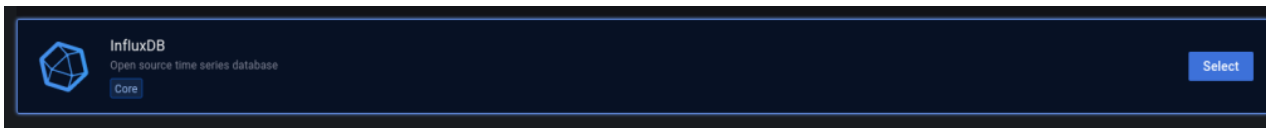
# Settings

## ■ AAS Web Dashboard 설정 (2)

- 좌측 하단 Configuraton 메뉴에서 Data sources 항목 클릭, Add data source 버튼을 통해 DB 연결



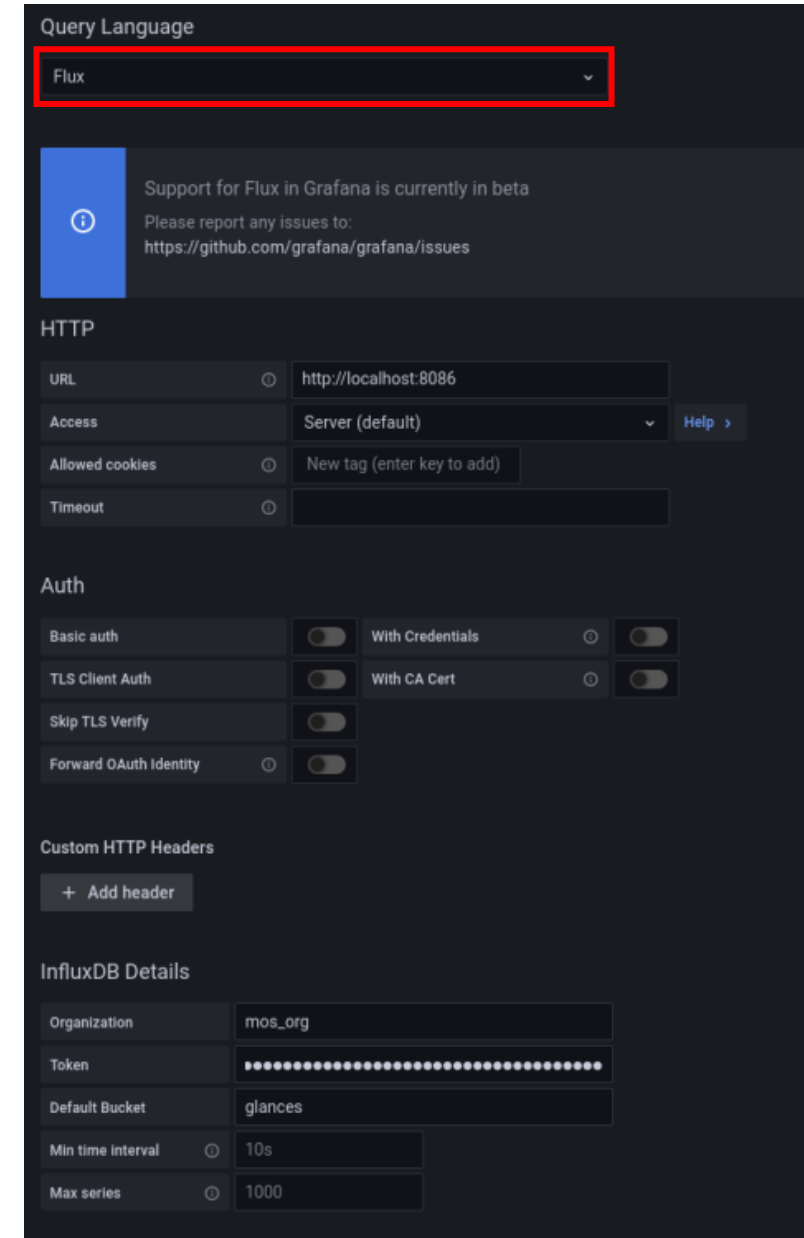
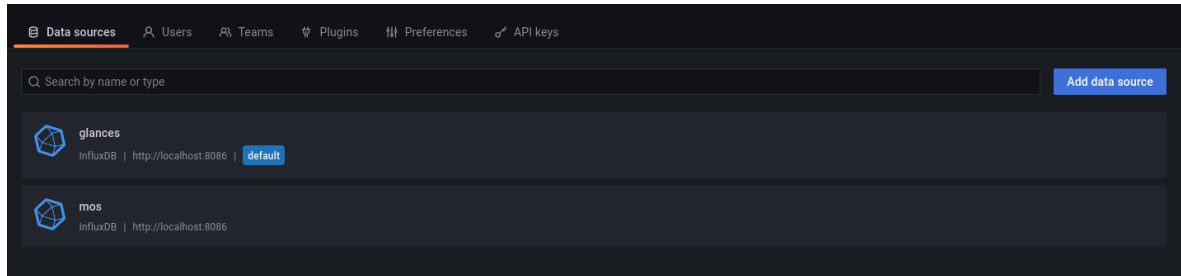
- InfluxDB 선택



# Settings

## ■ AAS Web Dashboard 설정 (3)

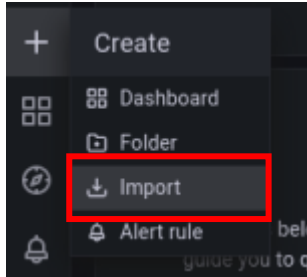
- Query Language : Flux 선택
- 우측 화면과 같이 세부 정보 입력
  - URL
  - DB Organization
  - Token
  - Default Bucket
- MOS, Glances Bucket 각각 연동



# Settings

## ■ AAS Web Dashboard 설정 (4) : 서버 모니터링 대시보드 생성

- 좌측 “+” 메뉴에서 Import 버튼 클릭, “2387” 입력 후 glances 선택하여 대시보드 생성 및 서버 모니터링 대시보드 확인



Importing dashboard from [Grafana.com](#)

Published by: nicolargo  
Updated on: 2021-11-28 18:59:23

**Options**

Name:

Folder:

**Unique Identifier (UID)**  
The unique identifier (UID) of a dashboard can be used to uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

ESYAe0tnk Change uid

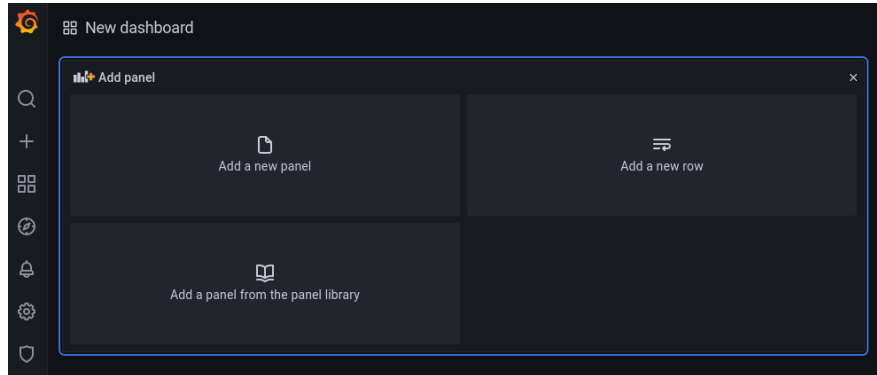
glances

Import Cancel

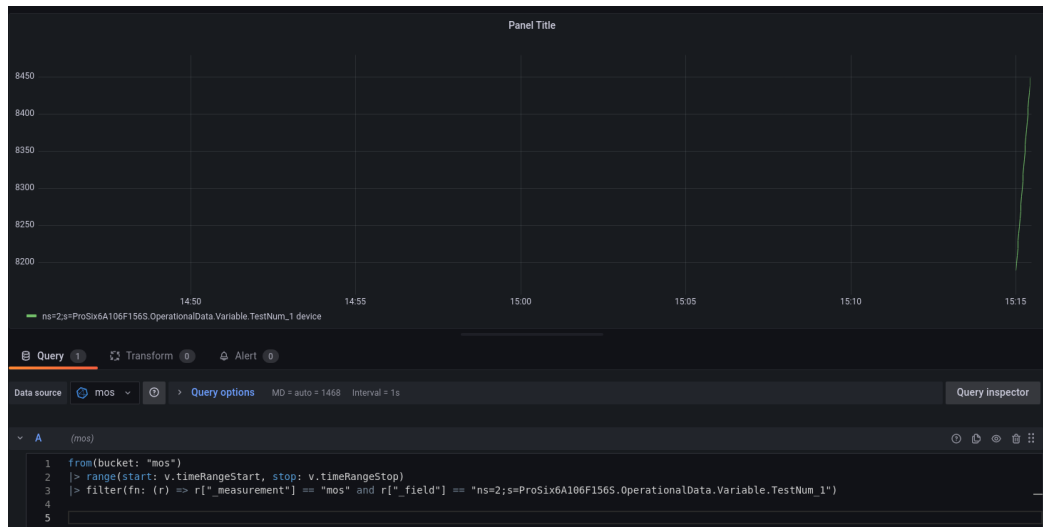


# Settings

- AAS Web Dashboard 설정 (5) : 기기 모니터링 데이터 대시보드 생성
  - 좌측 “+” 메뉴에서 Dashboard 버튼 클릭, New Dashboard 버튼으로 새 대시보드 생성
  - Add a new panel 항목 클릭하여 신규 패널 생성



- Data Source, DB 쿼리문 입력하여 데이터 그래프 확인



## [DB 쿼리문 예시] ※ 붉은색은 AAS 태그 이름

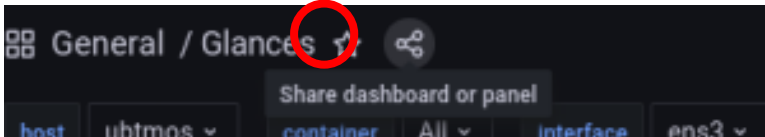
```
from(bucket: "mos")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r["_measurement"] == "mos" and r["_field"] ==
"ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1")
```



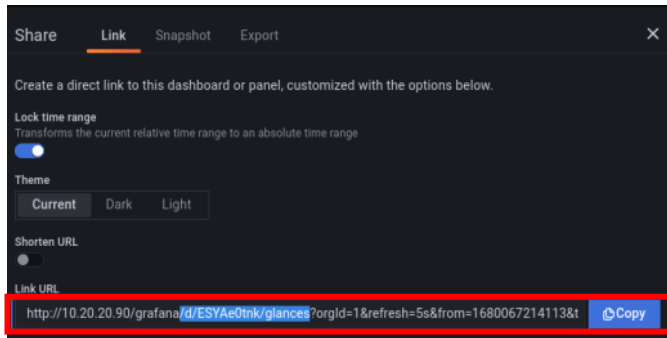
# Settings

## ■ AAS Web Dashboard 설정 (6) : AASX Package Browser 연동

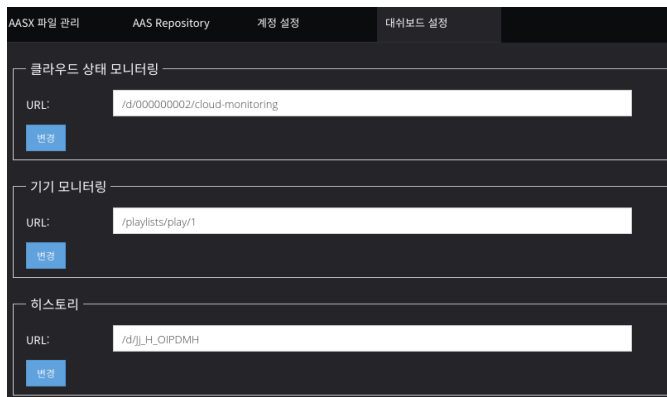
- 대시보드 좌측 상단 “Share dashboard or panel” 버튼 클릭하여 URL 확인



- “/d/\*\*\*\*\*/대시보드이름” 복사



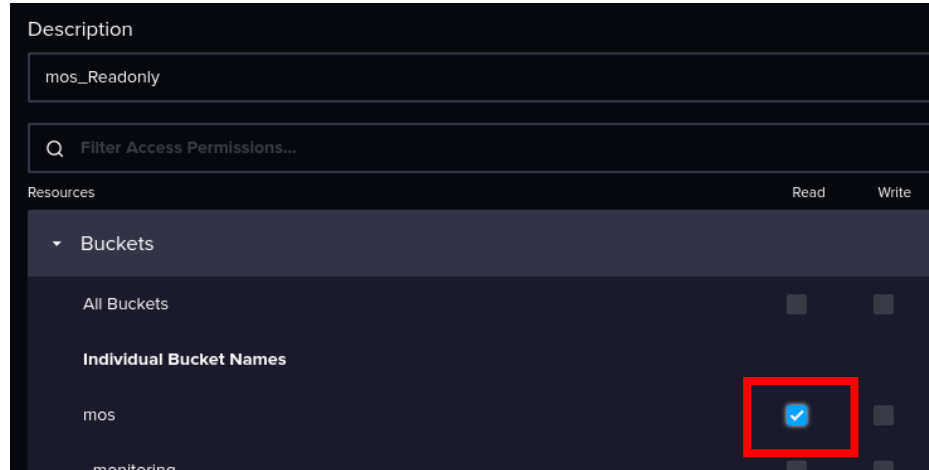
- AASX Package Browser 설정 메뉴에서 “대쉬보드 설정” 탭으로 이동하여 URL 입력



# Settings

## ■ REST API 이용하여 수집된 데이터 접근

- 데이터 저장 Bucket에 읽기 권한만 갖는 토큰 생성



- 아래 명령어를 통해 데이터 접근 확인 (curl 외에도 REST Client 프로그램 이용하여 데이터 접근 가능)

```
curl --request POST \W                                     생성한 토큰 입력
http://127.0.0.1:8086/api/v2/query?org=mos \W
--header 'Authorization: Token YywnjLzP8yGg0UX28AXhE8b0WDP0t1_dp_6mzlL514359Sargvf3Blrme_7NDSBxTfNi4tyVI8ZM6vxDZMbLlg==:' \W
--header 'Accept: application/csv' \W
--header 'Content-type: application/vnd.flux' \W
--data 'from(bucket:"mos")
|> range(start: 2023-05-24T05:51:57.000Z, stop: 2023-05-24T17:58:58.000Z) 접근할 데이터 기간 입력
|> filter(fn: (r) => r["_field"] ==
"ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1")' 접근할 태그 이름 입력, 해당 영역 생략 시 모든 태그 데이터 접근
```

# Settings

## ■ REST API 이용하여 수집된 데이터 접근 예시

```
root@ubumosntu:/opt/install# curl --request POST \  
> http://127.0.0.1:8086/api/v2/query?org=mos \  
> --header 'Authorization: Token YywnjLzP8yGg0UX28AXhE8b0WDP0t1_dp_6mzLL514359Sargvf3Blrme_7NDSBxTfNi4tyVI8ZM6vxDZMbLIg==' \  
> --header 'Accept: application/csv' \  
> --header 'Content-type: application/vnd.flux' \  
> --data 'from(bucket:"mos")  
> |> range(start: 2023-05-24T05:51:57.000Z, stop: 2023-05-24T17:58:58.000Z)  
> |> filter(fn: (r) => r["_field"] ==  
> "ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1")'  
,result,table,_start,_stop,_time,_value,_field,_measurement,host  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:22.047691Z,7854,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:23.067729Z,7864,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:24.042703Z,7874,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:25.044562Z,7884,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:26.067654Z,7894,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:27.022201Z,7904,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:28.042622Z,7914,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:29.049876Z,7924,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:30.067581Z,7934,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:31.042525Z,7944,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:32.048798Z,7954,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:33.067487Z,7964,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:34.042445Z,7974,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:35.045452Z,7984,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:36.067401Z,7995,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:37.042367Z,8005,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device  
,_result,0,2023-05-24T05:51:57Z,2023-05-24T17:58:58Z,2023-05-24T07:39:38.047155Z,8015,ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1,mos,device
```

# Settings

## ▪ (option)시계열 데이터베이스(influxDB) 데이터 저장경로 변경 (기존 /var/lib/influxdb → /data/influxdb)

- 데이터 저장 디렉토리 생성 및 권한 설정

```
root@mos:~$ mkdir -p /data/influxdb
```

```
root@mos:~$ sudo chown -R influxdb:influxdb /data/influxdb
```

- InfluxDB 서비스 중지

```
root@mos:~$ systemctl stop influxdb
```

- InfluxDB 데이터 파일 이동

```
root@mos:~$ mv /var/lib/influxdb/* /data/influxdb/
```

- InfluxDB 설정 파일 내 데이터 저장 경로 변경

```
root@mos:~$ vi /etc/influxdb/config.toml
```

```
bolt-path = "/data/influxdb/influxd.bolt"  
engine-path = "/data/influxdb/engine"
```

- InfluxDB 서비스 재가동

```
root@mos:~$ systemctl start influxdb
```

※ Root 디스크 외 별도 디스크에 데이터 저장 시, 데이터 저장 디렉토리 생성 후 마운트 작업 필요