



AAS기반 Cloud 솔루션 운용 가이드

Installation and Operation



네스트필드(주)

Contents

1. Cloud Solution Installation
2. Cloud Solution Setting

Part 01

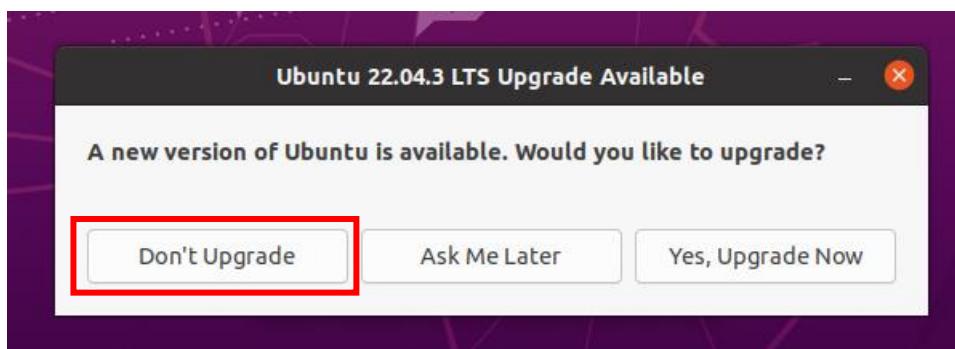
Cloud Solution Installation



네스트필드(주)

Installation

- 모든 설치 및 동작 과정, 이후 솔루션 운용은 root 계정을 통해 진행됩니다. 아래 명령어를 통해 root 권한 접속이 가능합니다.
 - `sudo su`
- Github CloudSolution 레포지토리의 두 압축파일을 /opt 디렉토리에 배치합니다.
 - `cd /opt` (만약 해당 디렉토리가 없다면 `mkdir /opt` 명령어로 디렉토리를 생성해주세요.)
- 두 압축파일은 아래 명령어를 통해 다운로드할 수 있습니다.
 - `wget https://github.com/kosmo-nestfield/Cloud_Solution/raw/main/Cloud%20Ver2.0/CloudSolution_apps.tar.gz`
 - `wget https://github.com/kosmo-nestfield/Cloud_Solution/raw/main/Cloud%20Ver2.0/CloudSolution_install.tar.gz`
- 아래 명령어를 통해 두 압축파일의 압축을 해제합니다.
 - `tar zxf ./CloudSolution_apps.tar.gz`
 - `tar zxf ./CloudSolution_install.tar.gz`
- 아래 문구와 같이 OS 업그레이드 창 출력 시 좌측 'Don't Upgrade' 항목을 선택해주세요. 본 매뉴얼은 Ubuntu 20.04 버전에 최적화되어있습니다.



Installation

- /opt/install 디렉토리의 01~15 설치 쉘 스크립트를 순서대로 실행합니다.

```
- cd install  
- ./01_install.sh ~ ./15_install_default_config.sh
```

(해당 디렉토리에 있는 install_all.sh 스크립트로도 설치 가능하나, 설치 시 발생되는 오류 확인을 위해 순서대로 실행하는 것을 권장합니다.)

```
root@ubuntu:/opt/install# ls  
01_install.sh          07_install_rabbitmq-c.sh  13_install_certbot.sh      libopen62541_cloud.tar.gz  
02_install_python.sh    08_install_influx.sh     14_install_systemctl_service.sh restart_services.sh  
03_install_python_package.sh 09_install_Lynis.sh   15_install_default_config.sh service  
04_install_rabbitmq.sh   10_install_mariadb.sh  configs                      smartfactory.sql  
05_install_cjson.sh     11_install_grafana.sh  get-pip.py  
06_install_open62541.sh  12_install_nginx.sh   install_all.sh
```

- 01~15 설치 스크립트를 모두 실행하였다면 ‘restart_services.sh’ 스크립트를 실행해주세요.

```
- ./restart_services.sh
```

- 시스템 Timezone 설정

```
- sudo ln -sf /usr/share/zoneinfo/Asia/Seoul /etc/localtime
```

- Timzone 설정 확인

```
- timedatectl
```

```
nestfield@ubuntu:~$ timedatectl  
          Local time: [REDACTED] KST  
          Universal time: [REDACTED] UTC  
             RTC time: [REDACTED]  
        Time zone: Asia/Seoul (KST, +0900)  
System clock synchronized: yes  
          NTP service: active  
    RTC in local TZ: no
```

Part 02

Cloud Solution Setting

Settings

▪ MariaDB 기본 설정 (1)

- mysql_secure_installation

- password : cloud 로 설정

- 순서대로 [기존계정비밀번호] > y > y > cloud > cloud > n > n > y > y 입력하시면 됩니다.

※ 다른 내용을 입력할 경우 Grafana를 포함한 웹 서비스가 정상적으로 동작하지 않을 수 있습니다.

> 웹 서비스 오류시에 mysql_secure_installation 명령어 입력 후 다시 설정하는 것이 도움될 수 있습니다.

```
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

※ 복사 붙여넣기하시는 것을 권장합니다.

```
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 127.0.0.1:8086 주소 입력 및 접속 후 ‘Quick Start’ 버튼 클릭 후 기본 설정 정보 입력
- Bucket과 Organization Name은 향후 데이터 수집 설정에서 사용되므로 반드시 기억

The screenshot shows the 'Setup Initial User' page of the InfluxDB web interface. At the top, there is a browser header with the InfluxDB logo and address bar showing '127.0.0.1:8086'. The main form has the following fields:

- Username:** cloudsolution
- Password:** (redacted)
- Confirm Password:** (redacted)
- Initial Organization Name:** ⓘ An organization is a workspace for a group of users.
- Initial Bucket Name:** ⓘ A bucket is where your time series data is stored with a retention policy.

Settings

▪ InfluxDB 기본 설정

- 좌측 메뉴 버튼 – Buckets 항목 선택



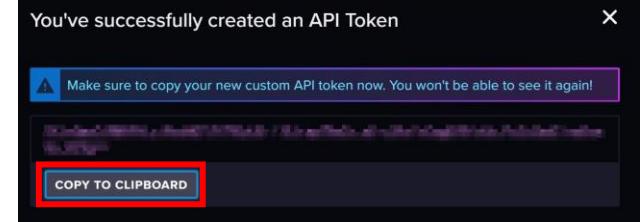
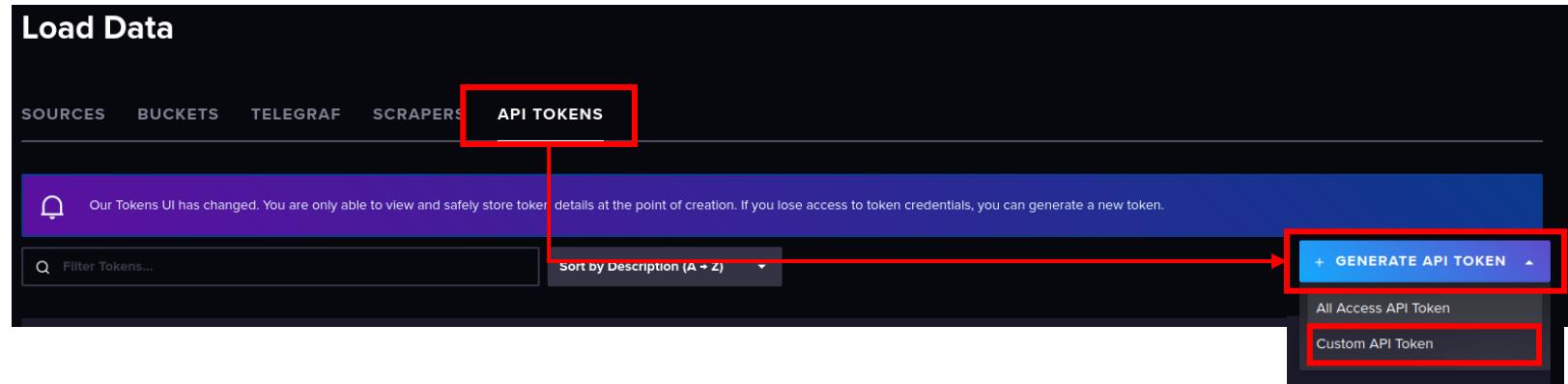
- 우측 상단 CREATE BUCKET 버튼을 눌러 “glances” Bucket 생성
- 데이터 삭제 기간 1year 로 설정

The screenshot shows the InfluxDB interface. On the left, a dark-themed sidebar lists 'Sources' and 'Buckets'. The 'Buckets' item is highlighted with a red box and a red arrow points to it from the top-left. On the right, a 'Create Bucket' dialog box is open over a list of buckets. The dialog has a red border and contains fields for 'Name*' (set to 'glances'), 'Delete Data' (set to '1 year'), and 'CREATE' and 'CANCEL' buttons. To the right of the dialog is a tooltip with a red border and a red arrow pointing to its title 'What is a Bucket?'. The tooltip defines a bucket as a named location for storing time series data and mentions a retention policy. It also links to 'how to write data'.

Settings

▪ InfluxDB 기본 설정

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



생성된 토큰은 반드시 기록/저장

- glances, 앞서 생성한 기본 Bucket, django 각각 토큰 생성 (총 3가지, Read/Write 권한 모두 필요)

Settings

▪ Glances 설정

- glances.conf 파일의 influxdb2 항목 아래 그림처럼 수정

```
- vi /opt/cfg/glances.conf or nano /opt/cfg/glances.conf
```

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

- org는 앞서 생성한 Organization 항목, bucket은 glances 입력 후
- Token 항목은 이전에 생성하고 저장한 glances의 API Token을 입력하시면 됩니다.
- `systemctl restart glances` 명령어로 서비스 재시작

Settings

■ itsdb 설정

- itsdb.service 파일의 TSDB_ORG, TSDB_BUCKET, TSDB_TOKEN 항목 수정

```
vi /usr/lib/systemd/system/itsdb.service or nano /usr/lib/systemd/system/itsdb.service
```

```
[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
```

- Org와 Bucket은 앞서 생성한 항목 이름을 입력해주세요
- 마찬가지로 token항목은 이전에 생성하고 저장한 기본 Bucket의 API Token을 입력하시면 됩니다.
- `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)

```
root@ubuntu:/opt/apps/django/SmartFactory# python manage.py createsuperuser
You have 19 unapplied migration(s). Your project may not work properly until
you run 'python manage.py migrate' to apply them.
User ID: admin
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

- 아래 명령어로 웹 서비스 재시작

- `systemctl restart django`
- `systemctl restart nginx`

- 웹브라우저를 실행하여 AASX Package Browser 실행 확인 (127.0.0.1주소 입력)



Settings

■ AAS Web Service 기본 설정

- `python manage.py createsuperuser` 입력 시 아래와 같이 Django 패키지 관련 오류 메시지가 출력되는 경우

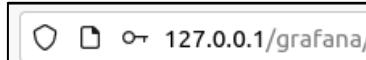
```
File "/usr/local/lib/python3.8/site-packages/django/urls/resolvers.py", line 708, in urlconf_module
    return import_module(self.urlconf_name)
File "/usr/local/lib/python3.8/importlib/_init_.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
File "<frozen importlib._bootstrap>", line 1014, in _gcd_import
File "<frozen importlib._bootstrap>", line 991, in _find_and_load
File "<frozen importlib._bootstrap>", line 975, in _find_and_load_unlocked
File "<frozen importlib._bootstrap>", line 671, in _load_unlocked
File "<frozen importlib._bootstrap_external>", line 848, in exec_module
File "<frozen importlib._bootstrap>", line 219, in _call_with_frames_removed
File "/opt/apps/django/SmartFactory/config/urls.py", line 2, in <module>
    from django.conf.urls import url
ImportError: cannot import name 'url' from 'django.conf.urls' (/usr/local/lib/python3.8/site-packages/django/conf/urls/_init_.py)
```

- 아래 명령어를 순서대로 입력 후 명령어 재실행
- `python -m pip uninstall django`
- `python -m pip install django==3.2.14`

Settings

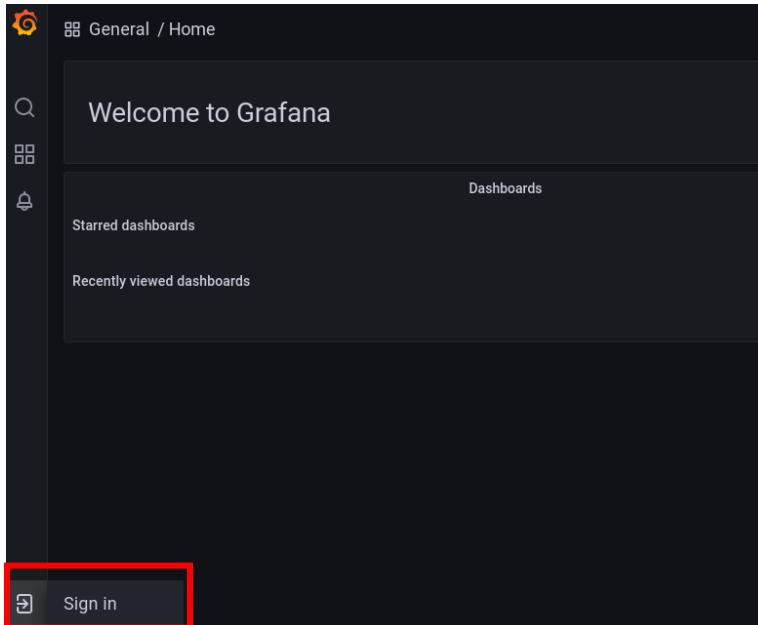
■ AAS Web Dashboard 설정

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



※ AASX Package Browser (http://127.0.0.1)에 먼저 로그인이 되어있어야 접속 가능

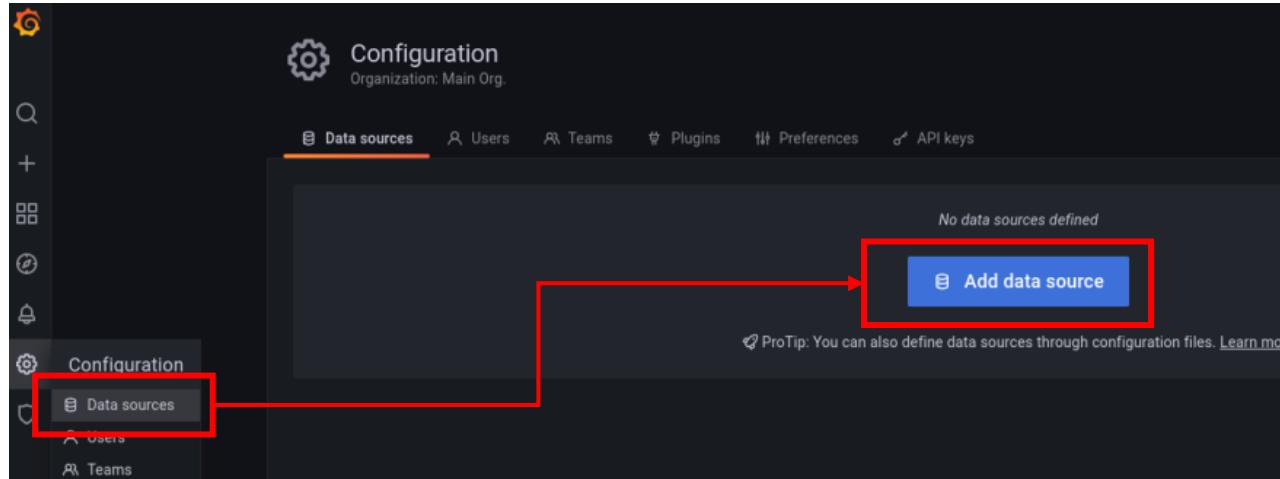
- 정상적으로 접속이 되지 않는 경우 `sudo systemctl restart grafana-server` 명령어로 서버 재실행
- 좌측 하단 Sign in 버튼으로 로그인 (기본 ID : admin , PW : admin)



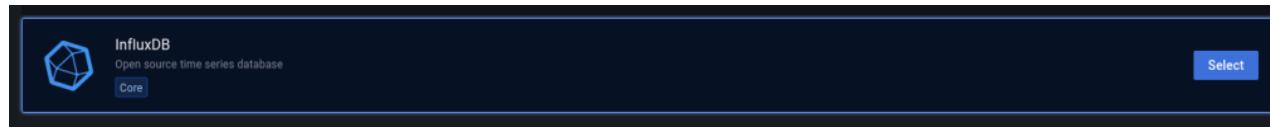
Settings

▪ AAS Web Dashboard 설정 (2)

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



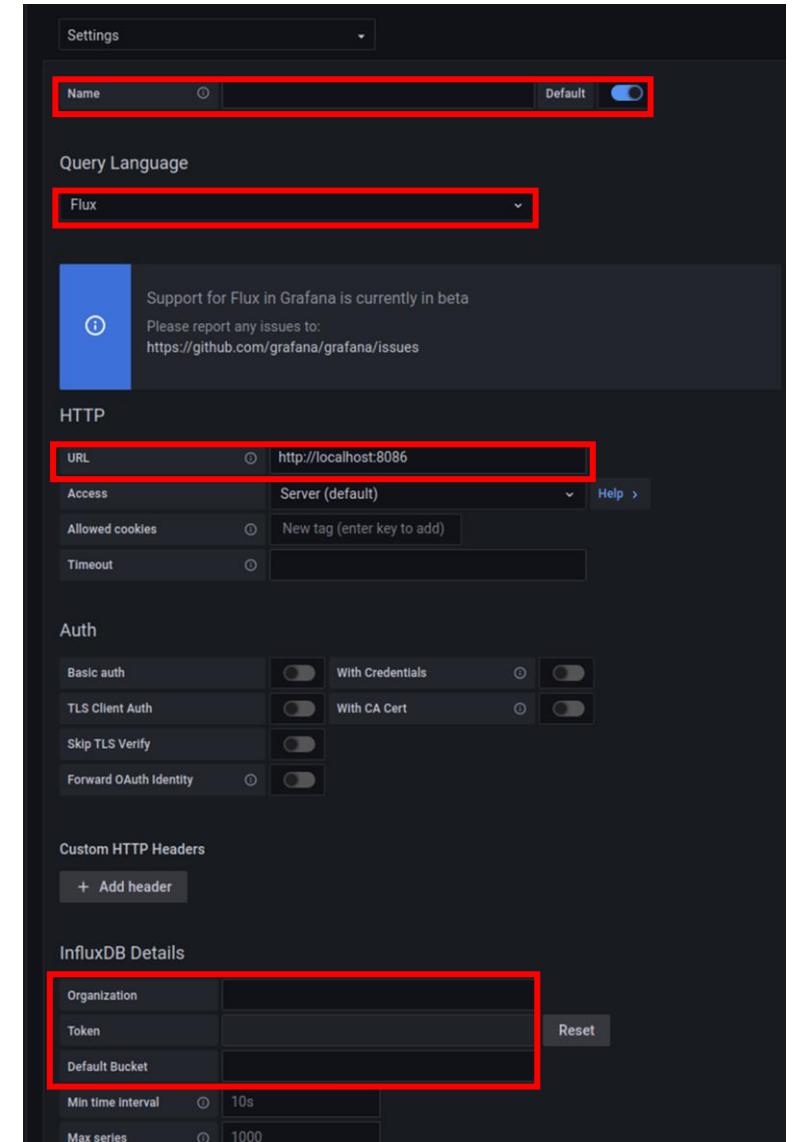
- InfluxDB 선택



Settings

■ AAS Web Dashboard 설정 (3)

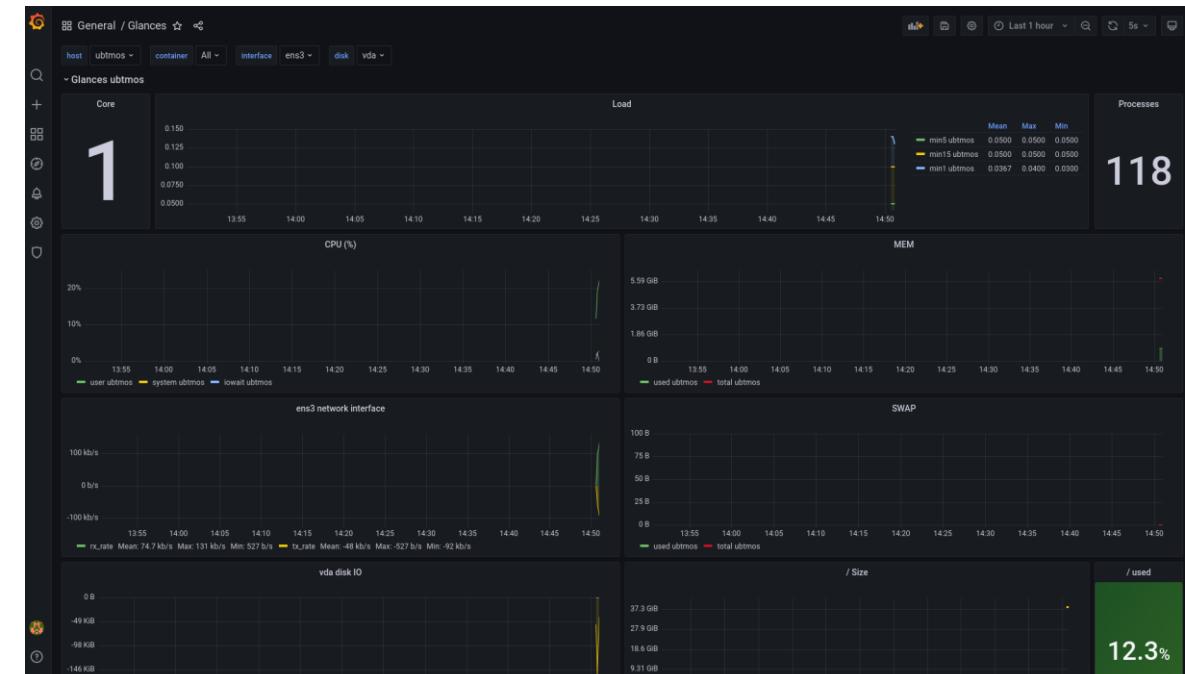
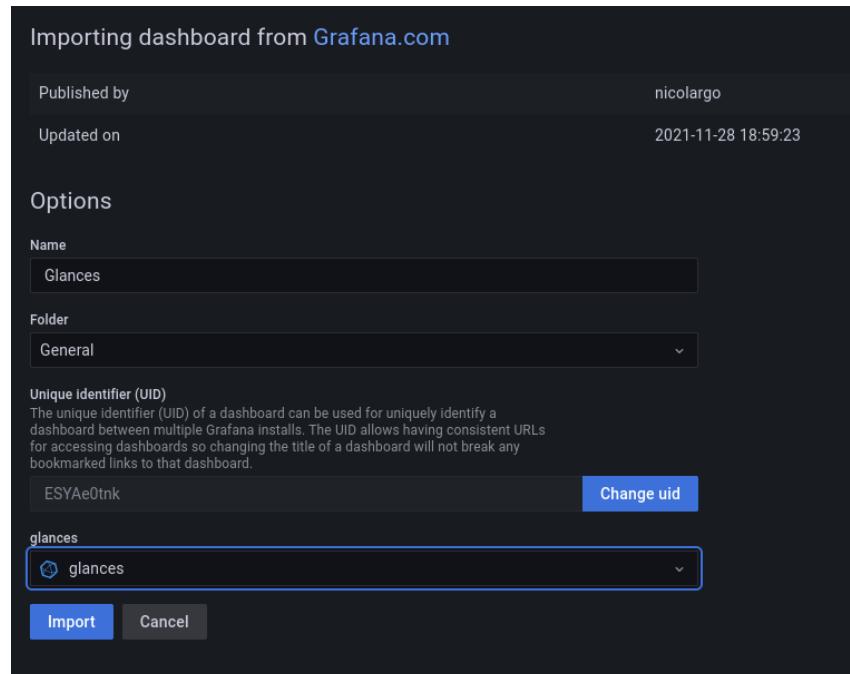
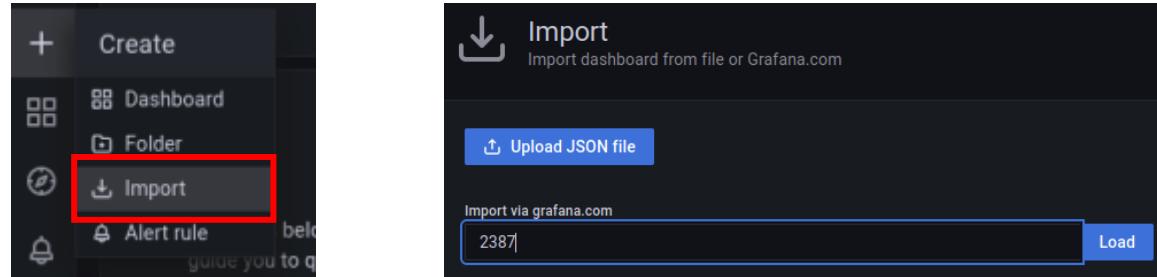
- Query Language : Flux 선택
- 우측 화면과 같이 세부 정보 입력
 - URL (http://localhost:8086 그대로 입력)
 - DB Organization (앞서 생성한 Organization)
 - Token (저장해둔 glances or 기본Bucket 의 API Token)
 - Default Bucket (glances or 기본Bucket)
- [기본 Bucket, Glances] 각각 연동



Settings

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

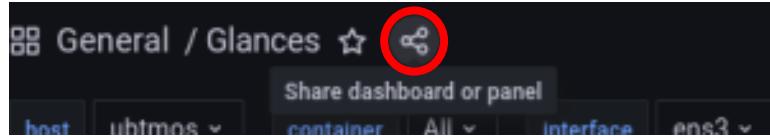
- 좌측 "+" 메뉴에서 Import 버튼 클릭, "2387" 입력 후 glances 선택하여 대시보드 생성 및 서버 모니터링 대시보드 확인



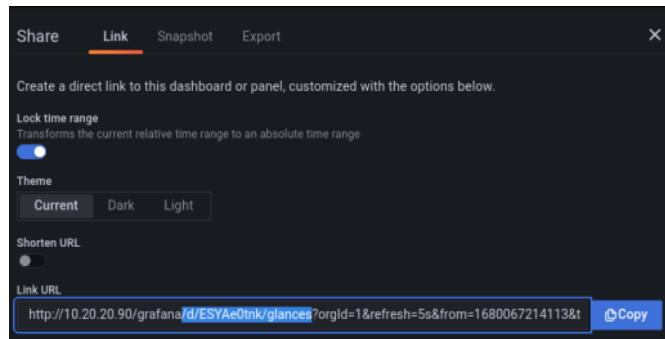
Settings

■ AAS Web Dashboard 설정 (5) : AASX Package Browser 연동

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



- "/d/*********/대시보드이름" 복사



- AASX Package Browser 설정 메뉴(아래 그림 좌상단 클릭)에서 "대쉬보드 설정" 탭으로 이동하여 "클라우드 상태 모니터링" 항목에 붙여넣기

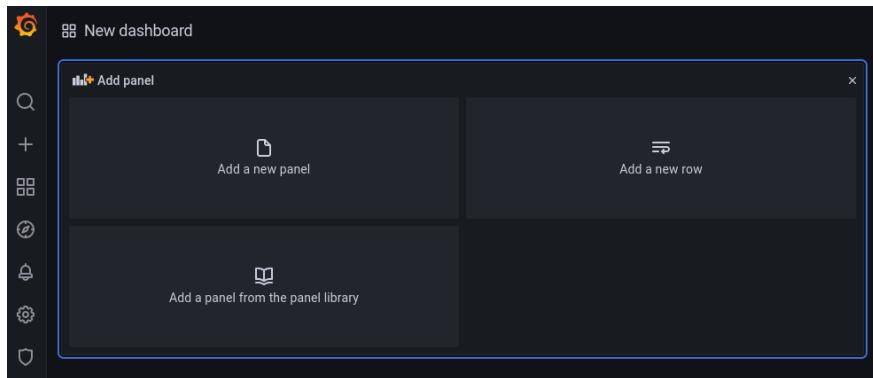
A screenshot of the 'AASX PACKAGE BROWSER' settings page. On the left, a sidebar lists 'AASX PACKAGE BROWSER', '클라우드 상태 모니터링' (highlighted with a red box), '기기 모니터링', '히스토리', and '사용자관리'. On the right, the '대쉬보드 설정' tab is active. Under '클라우드 상태 모니터링', there is a 'URL:' input field containing `/d/00000002/cloud-monitoring`, which is also highlighted with a red box.

Settings

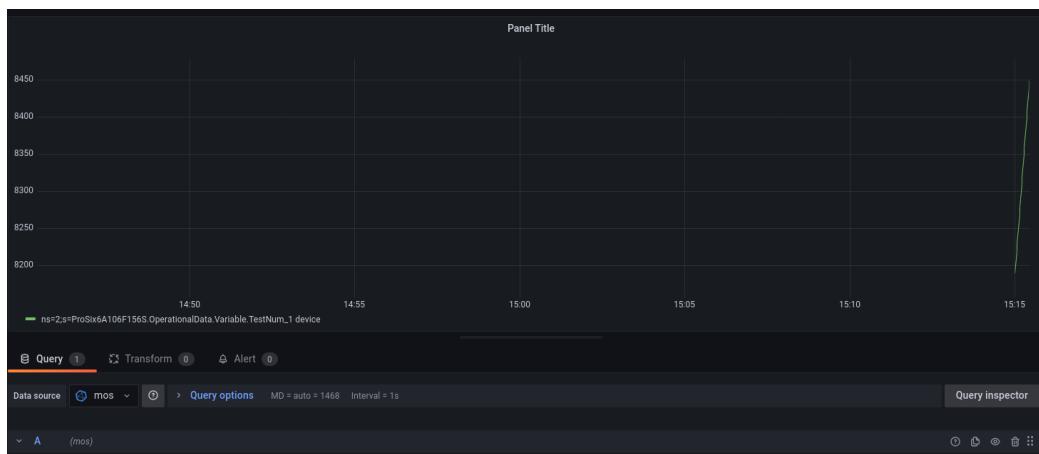
※ 아래 단계는 기능동작 점검 또는 실제 공정 데이터 수집 시 실행하는 단계로, 설치과정에선 생략하여 진행

▪ AAS Web Dashboard 설정 (6) : 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

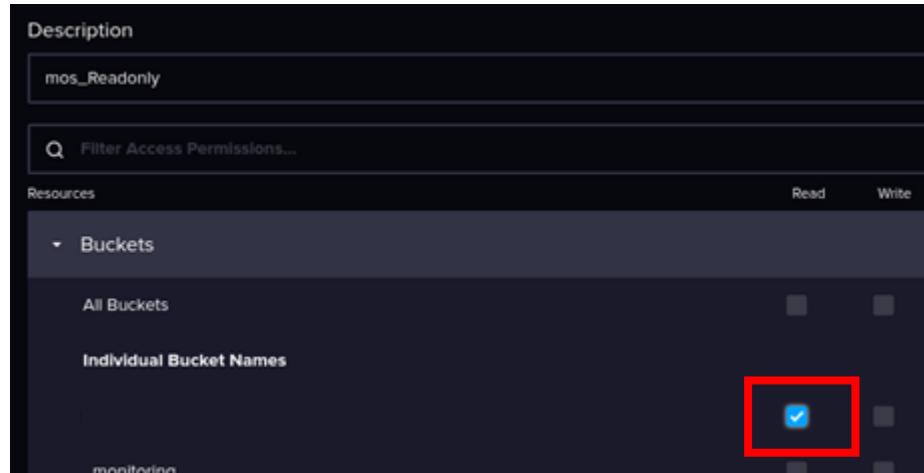
```
from(bucket: "기본Bucket")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r["_measurement"] == "기본Bucket" and r["_field"] ==
"ns=2;s=ProSix6A106F156S.OperationalData.Variable.TestNum_1")
```

Settings

※ 아래 단계는 기능동작 점검 또는 실제 공정 데이터 수집 시 실행하는 단계로, 설치과정에선 생략하여 진행

▪ REST API 이용하여 수집된 데이터 접근

- 데이터 저장 Bucket에 읽기 권한만 갖는 토큰 생성(미리 생성한 django 토큰 이용)



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

```
curl --request POST \
http://127.0.0.1:8086/api/v2/query?org=*** \
--header 'Authorization: Token YywnjLzP8yGg0UX28AXhE8b0WDP0t1_dp_6mzlL514359Sargvf3Blrme_7NDSBxTfNi4tyVi8ZM6vxDZMbLlg=' \
--header 'Accept: application/csv' \
--header 'Content-type: application/vnd.flux' \
--data 'from(bucket: "***") \
> 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')'
```

생성한 토큰 입력
접근할 데이터 기간 입력
접근할 태그 이름 입력, 해당 영역 생략 시 모든 태그 데이터 접근