


2차년도 주요 결과물

(과제명) 대규모 분산 에너지 저장장치 인프라의 안전한 자율운영
및 성능 평가를 위한 지능형 SW 프레임워크 개발
(과제번호) 2021-0-00077

- 결과물명 : 에너지 저장 장치 인프라 기반 온프레미스 데이터 수집·저장 시스템(SW)
- 작성일자 : 2022년 12월 1일

과학기술정보통신부 SW컴퓨팅산업원천기술개발사업
“2차년도 주요 결과물”로 제출합니다.

수행기관	성명/직위	확인
한국전자기술연구원	최효섭/책임연구원	

정보통신기획평가원장 귀하

사 용 권 한

본 문서에 대한 서명은 한국전자기술연구원 내부에서 본 문서에 대하여
수행 및 유지관리의 책임이 있음을 인정하는 것임.

본 문서는 작성, 검토, 승인하여 승인된 원본을 보관한다.

작성자 :	윤태일	일자 :	2022. 12. 01
-------	-----	------	--------------

검토자 :	김창우	일자 :	2022. 12. 02
-------	-----	------	--------------


승인자 :	최효섭	일자 :	2022. 12. 03
-------	-----	------	--------------

제 · 개정 이력

버전	변경일자	제·개정 내용	작성자
1.0	2022-12-01	최초 등록	윤태일

목 차

1. 개요	-----	1
2. 표준 데이터 모델링 기반 에너지 데이터 수집 시스템 개발	-----	3
3. 온프레미스 기반 수집 데이터의 저장 시스템 설계 및 개발	-----	4
4. OSS기반 에너지 저장장치 데이터 수집·저장 운영 관리 모니터링 시스템 설계	--	5

	에너지 저장장치 인프라 기반 온프레미스 데이터 수집·저장 시스템 (SW)	
	프로젝트	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW 프레임워크 개발

1. 개요

□ 목적

- 본 명세서의 목적은 대규모 분산에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 안전SW 프레임워크를 개발하기 위해 에너지 저장장치 인프라 기반 온프레미스 데이터 수집·저장 시스템을 구축하고 유지보수 및 운영관리를 위한 모니터링 시스템을 설계한다.

□ 범위

- 본 설계서는 에너지 저장장치 인프라 기반의 에너지 데이터 수집·저장 시스템에서 표준 데이터 모델링 기반의 에너지 데이터 수집 시스템 개발과 OSS(OpenSource SW) 기반 에너지 저장장치 데이터 수집·저장 운영관리 모니터링 시스템 설계를 중심으로 설명한다.

□ 시스템 개요

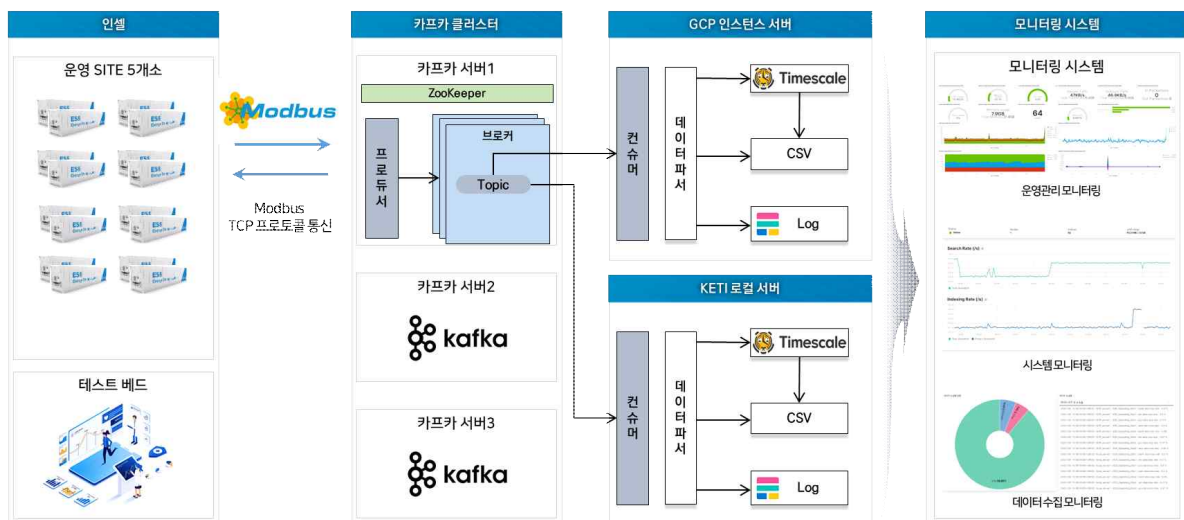


그림 1 에너지 수집·저장 시스템 구조

- 에너지 저장장치 인프라 기반의 에너지 데이터 수집·저장 시스템은 실제 운영사이트로부터 수집되는 ESS 데이터를 Modbus TCP 프로토콜 통신을 통해 퍼블릭 클라우드에 구축된 데이터 처리시스템을 통해 처리하고, 처리한 데이터를 퍼블릭 클라우드와 온프레미스 서버에 구축된 데이터베이스와 스토리지에 저장하는 시스템이다. 데이터 처리과정에서 생성된 로그를 통해 웹 서비스에서 데이터 상태 모니터링을 제공한다.

□ 관련 계획 및 표준

o 본 설계서는 아래 계획 및 표준을 참고한다.

구분	식별자	세부 내용	설명
설계서	ISO/IEC 9126	9126-1 (품질 모델) 9126-2 (외부 품질) 9126-3 (내부 품질) 9126-4 (사용 품질)	품질 특성 및 측정기준을 제시 소프트웨어의 기능성, 신뢰성, 사용성, 효율성, 유지보수 용이성, 이식성
	ISO/IEC 14598	14598-1 (개요) 14598-2 (계획과 관리) 14598-3 (개발자용 프로세스) 14598-4 (구매자용 프로세스) 14598-5 (평가자용 프로세스) 14598-6 (평가 모듈)	ISO 9126에 따른 제품 평가 표준: 반복성, 공정성, 객관성, 재생산성
	ISO/IEC 12119	소프트웨어 패키지 -제품설명서 -사용자문서 -프로그램과 데이터	패키지 SW 품질 요구사항 및 테스트

2. 표준 데이터 모델링 기반 에너지 데이터 수집 시스템 개발

□ 공통데이터 규격 기반 데이터 수집 시스템 개발

o 카프카 데이터 파이프라인 시스템 구축

- 카프카 토픽 구분을 통한 운영사이트 데이터 구분
- 카프카 데이터 확인을 위한 AKHQ 모니터링 툴 구축

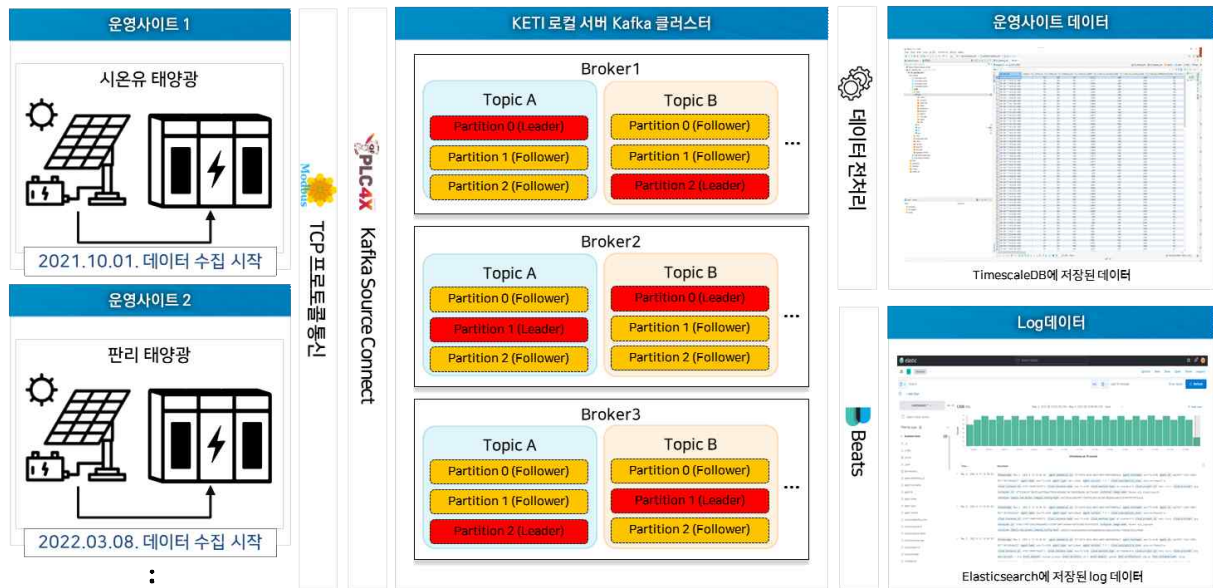


그림 2 카프카 클러스터 구조

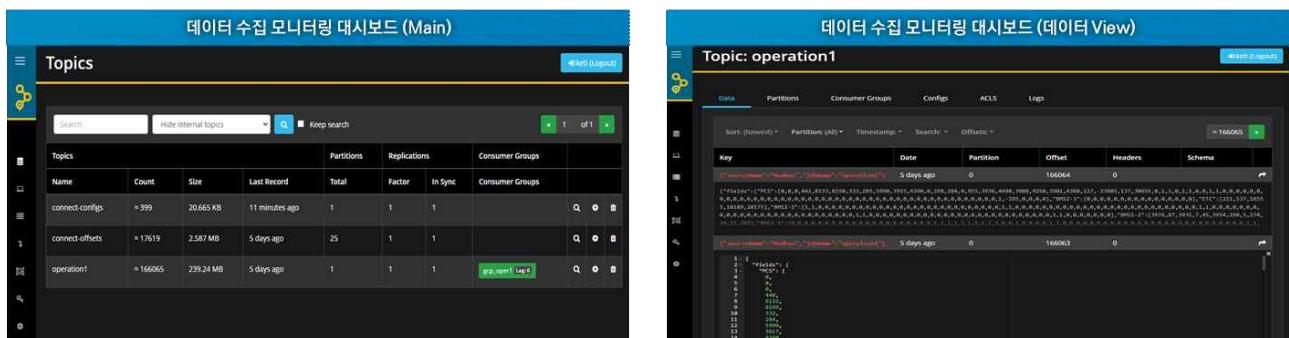


그림 3 AKHQ를 이용한 카프카 데이터 수집 모니터링

o 카프카 프로듀서 개발을 통한 데이터 저장 시스템 개발

- Modbus TCP 프로토콜과 결합한 카프카 프로듀서 개발
- 운영사이트별 데이터 처리 크기에 따른 카프카 프로듀서 구분

3. 온프레미스 기반 수집 데이터의 저장 시스템 설계 및 개발

□ 온프레미스 기반 데이터베이스 서버 설계 및 구축

○ 서버 용도에 따른 서버 구분

- 온프레미스 서버의 경우 데이터 축적의 목적에 맞춰 ESS 실험데이터 및 EV 배터리 데이터 등 다양한 데이터 저장할 수 있게 데이터베이스 설계 및 구축
- 클라우드 서버는 데이터 분석 및 웹서비스 관련된 특이치 저장

○ 실험데이터 수집

- 실험데이터 날짜 및 실험기관 구분에 따른 데이터베이스 구조 설계

□ 카프카 컨슈머를 통한 데이터 저장 시스템 설계 및 개발

○ 서버별 컨슈머 구축을 통한 데이터 관리

- 클라우드 서버, 온프레미스 서버별 컨슈머 구분을 통해 데이터 저장 및 에러 추적
- 운영사이트 별 데이터 처리 컨슈머 구현을 통한 빠른 속도의 저장시스템 구축

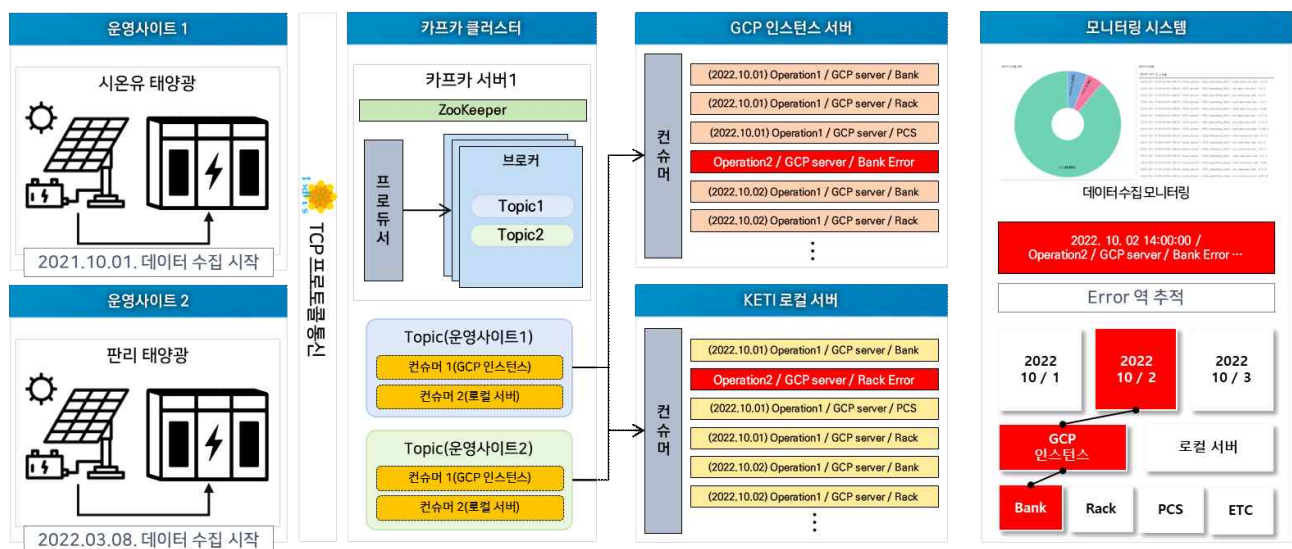


그림 4 서버별 컨슈머 구분에 따른 데이터 저장 및 에러 추적

4. OSS기반 에너지 저장장치 데이터 수집 · 저장 운영 관리 모니터링 시스템 설계

□ 데이터 수집 모니터링 시스템 설계

o Docker를 통한 ELK Stack(Elastic search, Log stash, Kibana) 구축

- Docker-compose를 통한 ELK Stack 구축
- Elastic search : 데이터 검색엔진
- Log stash : 데이터 집계, 변환, 저장
- Kibana : 데이터 분석 시각화 도구

Docker를 이용한 ELK Stack 구축

```
# 파일 다운로드
$ git clone https://github.com/ksundong/docker-elk-kor.git
$ cd docker-elk

# 설정파일 수정 및 보안을 위한 xpack 설정
-----/docker-elk/elasticsearch/config/elasticsearch.yml-----
## Default Elasticsearch configuration from Elasticsearch base image.
## https://github.com/elastic/elasticsearch/blob/master/distribution/docker/src/docker/config/elasticsearch.yml
#
cluster.name: "docker-cluster"
network.host: 0.0.0.0

## X-Pack settings
## see https://www.elastic.co/guide/en/elasticsearch/reference/current/setup-xpack.html
#
discovery.type: single-node

xpack.license.self_generated.type: trial
xpack.security.enabled: true
xpack.monitoring.collection.enabled: true
-----

# docker-elk/kibana/config/kibana.yml 대시보드 kibana 설정
-----docker-elk/kibana/config/kibana.yml -----
server.name: kibana
server.host: 0.0.0.0
elasticsearch.hosts: [ "http://elasticsearch:9200" ]
#monitoring.ui.container.elasticsearch.enabled: true
xpack.monitoring.ui.container.elasticsearch.enabled: true
## X-Pack security credentials
elasticsearch.username: elastic
elasticsearch.password: 비밀번호쓰기
-----

# logstash 설정
-----docker-elk/logstash/config/logstash.yml-----
## Default Logstash configuration from Logstash base image.
```



```
## https://github.com/elastic/logstash/blob/master/docker/data/logstash/config/logstash-full.yml
```

```
#
```

```
http.host: "0.0.0.0"
```

```
xpack.monitoring.elasticsearch.hosts: [ "http://elasticsearch:9200" ]
```

```
## X-Pack security credentials
```

```
#
```

```
xpack.monitoring.enabled: true
```

```
xpack.monitoring.elasticsearch.username: elastic
```

```
xpack.monitoring.elasticsearch.password: 비밀번호쓰기
```

```
-----docker-elk/logstash/pipeline/logstash.conf-----
```

```
input {
```

```
  tcp {
```

```
    port => 5000
```

```
  }
```

```
}
```

```
## Add your filters / logstash plugins configuration here
```

```
output {
```

```
  elasticsearch {
```

```
    hosts => "elasticsearch:9200"
```

```
    user => "elastic"
```

```
    password => "비밀번호쓰기"
```

```
  }
```

```
}
```

```
# Docker-compose 파일 수정 # 메모리 초과 방지를 위해 ES_JAVA_OPTS 수정
```

```
version: '3.2'
```

```
services:
```

```
  elasticsearch:
```

```
    build:
```

```
      context: elasticsearch/
```

```
      args:
```

```
        ELK_VERSION: $ELK_VERSION
```

```
    volumes:
```

```
      - type: bind
```

```
        source: ./elasticsearch/config/elasticsearch.yml
```

```
        target: /usr/share/elasticsearch/config/elasticsearch.yml
```

```
        read_only: true
```

```
      - type: volume
```

```
        source: elasticsearch
```

```
        target: /usr/share/elasticsearch/data
```

```
    ports:
```

```
      - "9200:9200"
```

```
      - "9300:9300"
```

```
    environment:
```

```
      ES_JAVA_OPTS: "-Xmx256m -Xms256m"
```

ELASTIC_PASSWORD: <비밀번호>

Use single node discovery in order to disable production mode and avoid bootstrap checks

see <https://www.elastic.co/guide/en/elasticsearch/reference/current/bootstrap-checks.html>

discovery.type: single-node

networks:

- elk

logstash:

build:

context: logstash/

args:

ELK_VERSION: \$ELK_VERSION

volumes:

- type: bind
 - source: ./logstash/config/logstash.yml
 - target: /usr/share/logstash/config/logstash.yml
 - read_only: true
- type: bind
 - source: ./logstash/pipeline
 - target: /usr/share/logstash/pipeline
 - read_only: true

ports:

- "5044:5044"
- "5000:5000/tcp"
- "5000:5000/udp"
- "9600:9600"

environment:

LS_JAVA_OPTS: "-Xmx256m -Xms256m"

networks:

- elk

depends_on:

- elasticsearch

kibana:

build:

context: kibana/

args:

ELK_VERSION: \$ELK_VERSION

volumes:

- type: bind
 - source: ./kibana/config/kibana.yml
 - target: /usr/share/kibana/config/kibana.yml
 - read_only: true

ports:

- "5601:5601"

networks:

- elk

depends_on:

- elasticsearch

networks:

elk:

driver: bridge

volumes:

elasticsearch:

Docker-compose 실행

\$ Docker-compose up -d

- o 경량 데이터 수집기 Beats를 통한 데이터 수집
 - Filebeat를 통한 로그데이터 수집

Filebeat를 통한 로그데이터 수집 및 모니터링

filebeat 설치

\$ curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.2.0-amd64.deb

/etc/filebeat/filebeat.yml 수정

#===== Filebeat inputs =====

filebeat.inputs:

Each - is an input. Most options can be set at the input level, so

you can use different inputs for various configurations.

Below are the input specific configurations.

- type: log

Change to true to enable this input configuration.

enabled: true

Paths that should be crawled and fetched. Glob based paths.

paths:

- /home/user/test/log/*.json

json.keys_under_root: true

json.overwrite_keys: true

json.add_error_key: true

json.expand_keys: true

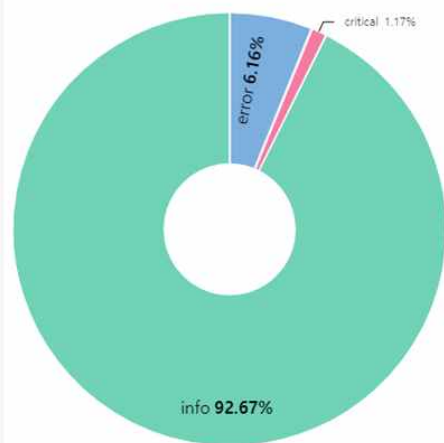
#filebeat setup 및 실행

\$ filebeat setup -e

\$ service filebeat start

[illegible]

데이터 손실률 현황



데이터 손실률

데이터 위치 및 손실률	
2022-11-09 00:00:00+09:00 / GCP_server1 / ESS_Operating_Site1 / bank data loss rate : 0.01...	
2022-11-09 00:00:00+09:00 / GCP_server1 / ESS_Operating_Site1 / etc data loss rate : 0.01 %	
2022-11-09 00:00:00+09:00 / GCP_server1 / ESS_Operating_Site1 / pcs data loss rate : 0.01 %	
2022-11-09 00:00:00+09:00 / GCP_server1 / ESS_Operating_Site1 / rack data loss rate : 0.01 %	
2022-11-09 00:00:00+09:00 / GCP_server1 / ESS_Operating_Site2 / bank data loss rate : 0.24...	
2022-11-09 00:00:00+09:00 / GCP_server1 / ESS_Operating_Site2 / etc data loss rate : 0.32 %	
2022-11-09 00:00:00+09:00 / GCP_server1 / ESS_Operating_Site2 / pcs data loss rate : 0.32 %	
2022-11-09 00:00:00+09:00 / GCP_server1 / ESS_Operating_Site2 / rack data loss rate : 0.23 %	
2022-11-09 00:00:00+09:00 / local_server1 / ESS_Operating_Site1 / bank data loss rate : 0.01...	
2022-11-09 00:00:00+09:00 / local_server1 / ESS_Operating_Site1 / etc data loss rate : 0.01 %	
2022-11-09 00:00:00+09:00 / local_server1 / ESS_Operating_Site1 / pcs data loss rate : 0.01 %	
2022-11-09 00:00:00+09:00 / local_server1 / ESS_Operating_Site1 / rack data loss rate : 0.01 %	
2022-11-09 00:00:00+09:00 / local_server1 / ESS_Operating_Site2 / bank data loss rate : 0.24...	
2022-11-09 00:00:00+09:00 / local_server1 / ESS_Operating_Site2 / etc data loss rate : 0.32 %	
2022-11-09 00:00:00+09:00 / local_server1 / ESS_Operating_Site2 / pcs data loss rate : 0.32 %	

그림 5 수집한 로그 데이터 및 로그 모니터링 결과