

2020년 AI 바우처 지원 사업

『지능형 연계 거래량 예측 엔진 구축』

아키텍처 정의서

2020. 12. 28.

어두운이(가) 표시된 사진

자동 생성된 설명

**목 차**

**1. 어플리케이션 아키텍처** **2**

**2. 소프트웨어 구성도** **3**

**3. 하드웨어 구성도** **7**

**4. 네트워크 구성도** **9**

1. 어플리케이션 아키텍처

|  |
| --- |
|  |

2. 소프트웨어 구성도

2.1 Component Diagram for Train/Test Model

|  |
| --- |
|  |

2.1.1 패키지 리스트

|  |  |  |
| --- | --- | --- |
| 패키지 | 모듈 | 설명 |
| ECrossLogTrainerWeb | Dashboard | 데이터셋 및 모델 학습 관리 웹 인터페이스 |
| ECrossLogTrainerAPI | APIService | 시스템 연동(외부/내부) |
| ECrossLogTrainer | TrainingManager | ECrossLog 데이터/모델 관리 도구 |
| ECrossLogParser | DataSetGenerator | ECrossLog 데이터셋 생성기(특성추출) |
| DeepCoreMini | Seq2SeqCNN | 모델 학습기 및 서빙 |

2.1.2 외부시스템 리스트

|  |  |  |
| --- | --- | --- |
| 외부시스템 | 기능 | 설명 |
| eCross | System | eCross System은 내부 로그 데이터베이스를 가지고 있다. 이 로그들을 ECrossLogAPI를 통해 공급한다. |
| ElasticSearch | 데이터셋 저장 | ECrossLogTrainer와 ECrossLogParser에 의해 생성된 데이터셋이 저장되고 조회되는 곳이다. |
| Kibana | 데이터셋 시각화 | 저장된 Dataset에 대한 시각화가 가능하다. |
| PostgreSQL | 모델 정보 저장 | ECrossTrainer와 DeepCoreMini에 의해 생성된 모델 정보가 저장되고 조회되는 곳이다. |
| MinIO | 모델 파일 저장 | ECrossTrainer와 DeepCoreMini에 의해 생성된 모델 파일이 저장되고 조회되는 곳이다. |
| Docker Daemon Server | 모델 서빙 도커 이미지 저장 | 모델 서빙을 위한 도커 이미지가 빌드되고 저장되는 곳이다. |

2.2. Component Diagram for Serving Model

|  |
| --- |
|  |

2.2.1 패키지 리스트

|  |  |  |
| --- | --- | --- |
| 패키지 | 모듈 | 설명 |
| ECrossLogServing | APIService | 시스템 연동(외부/내부) |
| ECrossLogParser | DataSetGenerator | ECrossLog 데이터셋 생성기(특성추출) |
| DeepCoreMini | Seq2SeqCNN | 모델 학습기 및 서빙  (이전에 학습해 놓은 모델을 파일형태로 도커 이미지에 존재한다. 이를 로드하여 모델을 서빙한다.) |

2.2.2 외부시스템 리스트

|  |  |  |
| --- | --- | --- |
| 외부시스템 | 기능 | 설명 |
| eCross | System | eCross는 내부 로그 데이터베이스를 가지고 있다. eCorss는 ECrossLogServing에서 시스템 로그 요청 시 반환해줘야 한다.  역으로 특정한 범위의 예측값을 제공받도록 ECrossLogServing쪽에 요청할 수 있다. (자세한 내용은 아래 Sequence Diagram 참고) |
| Docker Image | 가상환경 구동 | Docker Daemon Server가 있는 서버이면 해당 이미지를 설치없이 구동 가능하다. 여러 모델 서빙을 위해 2개 이상 도커 컨테이너로 생성 및 구동될 수 있다. |

2.3 Sequence Diagram

|  |
| --- |
|  |

3. 하드웨어 구성도

3.1 Train/Test model

|  |
| --- |
|  |

* 시스템 관리자는 eCross를 접근 및 관리할 수 있다.
* 시스템 관리자는 ECrossLogTrainerWeb의 웹 서비스로 접근하여 데이터셋 및 모델을 관리할 수 있다.
* 시스템 관리자는 ECrossLogTrainerAPI의 REST API 서비스로 접근하여 데이터셋 및 모델을 관리할 수 있다.
* ECrossTrainerAPI는 DockerDaemonServer, ElasticSearch, PostgreSQL, MinIO 데이터베이스로 접근할 수 있어야 한다.

3.2 Serving model

|  |
| --- |
|  |

* 시스템 관리자는 eCross를 접근 및 관리할 수 있다.
* ECrossLogTrainerServing은 eCross로 접근하여 해당 기간의 로그를 제공받을 수 있다.
* eCross는 ECrossLogTrainerServing으로 접근하여 해당 기간의 예측 값을 제공받을 수 있다.

3.3 Server Spec

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 구분 | 사양(1식 기준) | | | | OS | 비고 |
| CPU | Memory | Disk | |
| Local | External |
| eCross 서버 | - | - | - | - | - | 기존 서버 사용 |
| DeepCoreMini 서버 | 1.6GHz\*2EA  (16Core) | 64GB | 146GB\*2EA | - | Ubuntu 20.04 | - |
| DB 서버 | 1.6GHz\*2EA  (4Core) | 128GB | 146GB\*2EA | 1TB\*6EA | Ubuntu 20.04 | - |

4. 네트워크 구성도

4.1. Train/Test model

|  |
| --- |
|  |

* ECrossLogTrainerWeb은 내부망을 통해 웹 서비스를 내부망 사용자들에게 제공할 수 있어야 한다.
* ECrossLogTrainerAPI는 내부망을 통해 REST API 서비스를 ECrossLogTrainerWeb 및 내부망 사용자들에게 제공할 수 있어야 한다.
* Kibana는 내부망을 통해 데이터셋 시각화 서비스를 내부망 사용자들에게 제공할 수 있어야 한다.
* DockerDaemonServer, PosrgreSQL 및 MinIO는 ECrossLogAPI에서 접근 가능해야 한다.
* ElasticSearch는 ECrossLogAPI 및 Kibana에서 접근 가능해야 한다.

4.2. Serving model

|  |
| --- |
|  |

* ECrossLogServing은 ecross\_system과 양방향 접근이 가능해야 한다.