

기업연계 프로젝트

산업재산권 데이터 수집 및 분석

프로젝트: 커

차 례

1. 프로젝트 개요
2. 팀 협업 도구 선정
3. 데이터 파이프라인 아키텍처
4. 핵심 구현 Task
5. 프로젝트 결과
6. 향후 개선 필요 사항
7. Q&A

프로젝트 개요

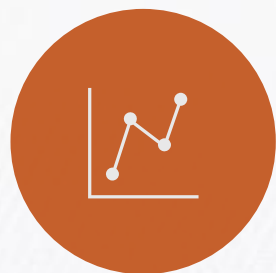
팀 소개

프로젝트:커



박우열

API 호출
데이터 전처리
DB 적재



강상우

시스템 모니터링
네트워크 최적화



조현익

API 호출
데이터 적재
대시보드 구현



정혜인

회의록 작성
커뮤니케이션
자료 작성

프로젝트 개요

프로젝트 목적

법인과 대학이 보유한 산업재산권 데이터 수집 · 분석 · 활용

주요 Tasks

1. KIPRIS Plus에서 API를 통해 제공하는 데이터 수집

- XML 형식 데이터를 MySQL DB에 적재

2. 일일 단위 데이터 현행화

- 기업 고객에 대한 특허고객번호
- 수집 된 산업재산권 데이터에 대한 법적상태 변동 사항

3. 산업재산권 분석 대시보드 구현



팀 협업 도구 선정

협업 도구 선정의 필요성

비대면으로 진행되는 프로젝트로
팀원 간 정보 공유 및 의사 소통 수단이 매우 중요하다고 판단

선정 협업 도구



- 실시간 공동 작업 가능
- 다이어그램, 플로우 차트, 와이어프레임 등 도식 작성 용이

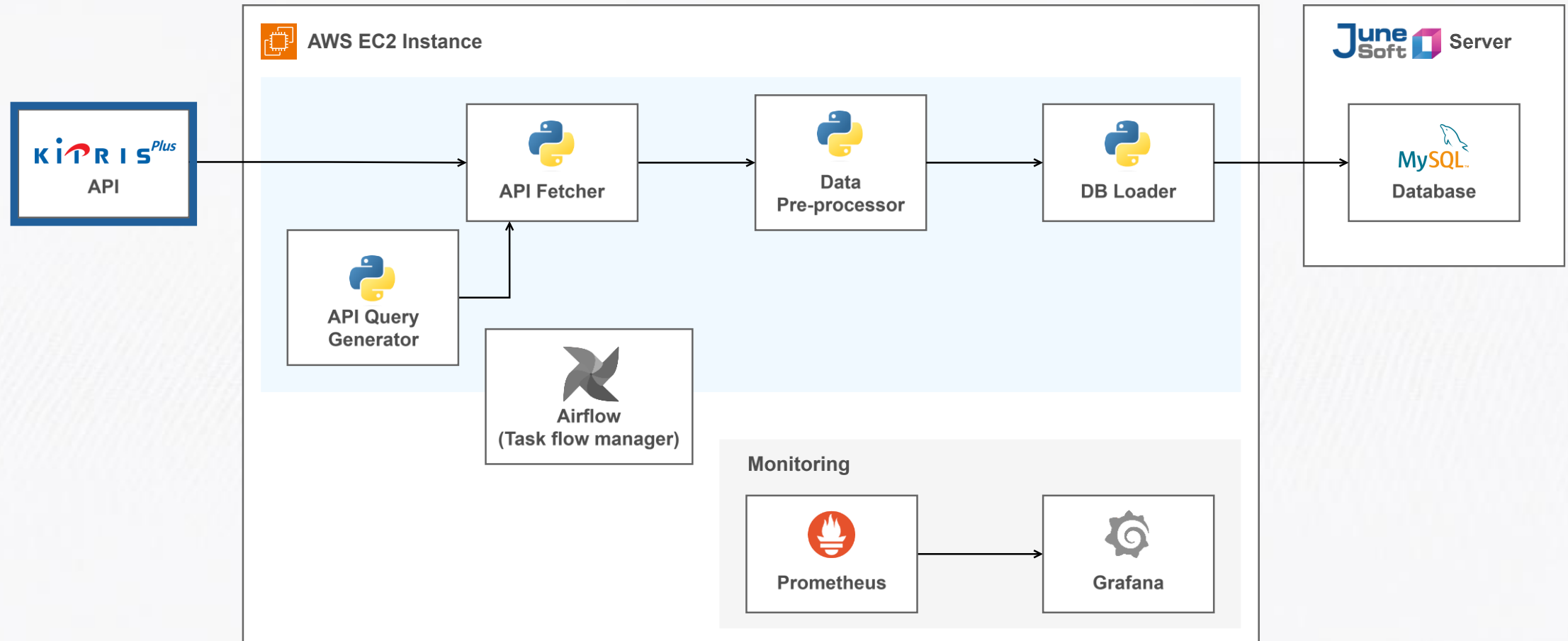


- 프로젝트 문서화 및 테스트 데이터 정리에 용이



- 음성 회의, 화면 공유, 텍스트 기반 대화에 특화

데이터 파이프라인 아키텍처




핵심 구현 Task

KIPRIS Plus API 명세 파악

KIPRIS API 출력 데이터 형식 확인 XML

특허정보활용서비스 API 명세 확인 후 적합 API 선정


주요국 개방 정보

1. 특허 · 실용 공개공보

IP정보	특허·실용 공개공보	KIPO구분	한국공보 > 특허/실용
주기	매일	범위	1983 ~ 현재
형식	SGML, XML, PDF, JPG, TXT	크기	TXT : 37.00GB SGML/PDF : 530.00GB XML/PDF : 3,670.00GB XML/PDF(ST.96) : 4,000.00GB
제공방법	저장매체, FTP, Web download, Web service(API)	가격	당해 연도 (3,273,463원) 과거분 전체 (18,004,045원)
출처	https://plus.kipris.or.kr/portal/data/service/DBIL_000000000000002/view.do?menuNo=210000&kppBCode=&kppMCode=&kppSCode=&subTab=&entYn=N&clckKeyword=		

2024 IP5개방데이터 가이드북

특허·실용 공개·등록공보

1. 개요

한국 특허/실용신안 공개 및 등록공보의 서지정보, 대표도, 전문정보 등을 API(SOAP/REST방식)으로 제공합니다.

2. 세부 정보 내용

[API 서비스]

① 검색: 일반검색 및 항목별검색 기능을 통해 특허·실용신안 공보 정보를 XML 형태로 제공합니다.

② 서지정보: 출원번호, 등록일자, 발명의 명칭, 등록상태 등의 정보를 XML 형태로 제공합니다.

③ 도면/전문: 공개 및 공고 전문, 대표도면의 다운로드 경로를 XML 형태로 제공합니다.

메타정보

분류체계	국내 IP데이터 > 한국공보 > 특허/실용		
제공기관	특허청	업데이트 주기	일 단위
데이터 설명 등록일자	2014-10-20	데이터 설명 갱신일자	2020-10-27
태그	등록공보, 특허, 실용신안, 공개공보, 실용, 특허·실용 공개등록공보, 공개공보		

서비스 유형

SOAP

REST

신청하기

일반검색

항목별검색

서지정보

도면/전문

부가기능

1. 전체검색

2. 자유검색

3. 출원번호 검색

특허정보활용서비스 API 명세

핵심 구현 Task

KIPRIS Plus API 명세 파악

Postman을 이용한
API 요청 및 응답 테스트 수행

Collection과 환경 변수 기능 제공
다수의 API와 API 인증 토큰을 체계적으로 관리 가능

데이터와 Header 분석 용이

KIPRIS Plus API 명세에 누락된
필수 input params 파악

상표 정보 전체 검색 API의 상표 종류 파라미터 누락

The screenshot shows the Postman interface with a GET request configured for the URL `http://plus.kipris.or.kr/kipo-api/kipi/designInfoSearchService/getAc...`. The request parameters are listed in the 'Params' tab:

Param	Value	Description
pageNo	1	페이지번호
numOfRows	500	페이지당건수(기본 : 30, 최...
sortSpec		정렬기준
descSort		정렬방식(asc방식 : false, ...
ServiceKey	{{june_accesskey}}	
ServiceKey	{{private_key}}	

The response is shown in the 'Body' tab as a 200 OK status with a response time of 672 ms and a size of 171.34 KB. The response body is in XML format:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response>
  <header>
    <requestMsgID></requestMsgID>
    <responseTime>2024-11-17 17:55:54.5554</responseTime>
    <responseMsgID></responseMsgID>
    <successYN>Y</successYN>
    <resultCode>00</resultCode>
  
```

Annotations on the image highlight the 'HTTP method' (GET), 'Request parameter', 'Status code / response time', and 'Response body'.

핵심 구현 Task

API 제한 사항 파악 – Mock API 구현

API Quota 제한

초당 50회 이상 호출 발생 시 API key 차단 조치

데이터 요청 시 **초당 최대 요청 횟수** 준수 필요

요청 횟수 제어 알고리즘 적용이 요구 됨

개발 환경에서의 요청 속도 제한
로직 **테스트 환경** 구축

상표 정보 전체 검색 API의 상표 종류 파라미터 누락

질문	API 호출 횟수 제한
답변	<ul style="list-style-type: none"> o (공통) 서버 및 네트워크 부하 해소를 위해 회원 계정 기준 초당 50건 이상 API 호출 발생 시 서비스 이용이 제한됩니다. * 초당 API 호출 횟수를 50회 미만으로 설정 부탁드립니다. o (무료 사용자) 월 1천 건 이하로 API 호출이 가능하며, 상품별로 API 호출 시 전체 합산 횟수가 1천 건을 넘게 되면 서비스 이용이 제한됩니다. * 이용 제한은 매월 1일 초기화되며, 익월 다시 서비스 이용이 가능합니다.

KIPRIS Plus Open API 개발 가이드

X	Headers	Payload	Preview	Response	Initiator	Timing
1	<response>					
2	<header>					
3	<requestMsgID/>					
4	<responseTime>2024-11-10 14:15:22</responseTime>					
5	<responseMsgID/>					
6	<successYN>N</successYN><resultCode>					
7	10</resultCode>					
8	<resultMsg>Blocked users.</resultMsg>					
9	</header>					
10	</response>					
11						

API key 차단 시, 응답 예시

핵심 구현 Task

API 제한 사항 파악 – Mock API 구현

FastAPI를 이용한 Mock API Server 구현

- 비동기 요청 지원 Web framework
- 실제 API와 유사한 환경을 효율적으로 시뮬레이션 가능

작동 방식

1. 요청 수신 시, 현재 시각과
deque에 저장된 요청 timestamp 비교
2. Sliding window 크기를 초과한 요청은
삭제 / 요청 거부
3. 허용된 요청인 경우, 처리 후 응답



핵심 구현 Task

API 제한 사항 파악 – Mock API 구현

작동 방식

· 정상 요청의 경우

클라이언트 – 정상 Dummy XML data 응답 수신

```
<response>
  <header>
    <requestMsgID/>
    <responseTime>1732167569.8265939</responseTime>
    <responseMsgID/>
    <successYN>Y</successYN>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL SERVICE.</resultMsg>
  </header>
  <body>
    <items>
      <item>
```

서버 – 200 정상 응답 로그 기록

```
INFO: 123.123.123.123:12345 - "GET /mock_api HTTP/1.1" 200 OK
```

핵심 구현 Task

API 제한 사항 파악 – Mock API 구현

작동 방식

· 최대 요청 수 초과 요청의 경우

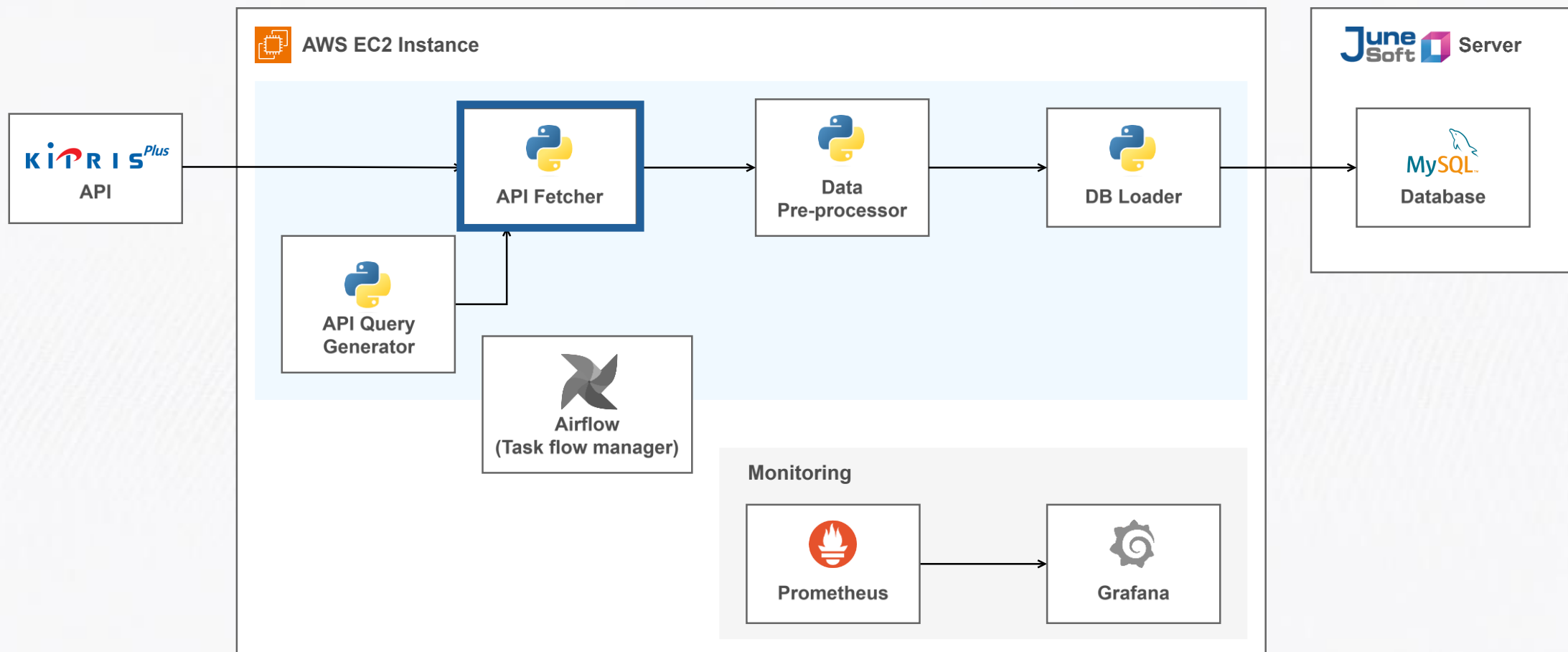
클라이언트 – 차단 메시지 Dummy XML data 수신

X	Headers	Payload	Preview	Response	Initiator	Timing
1				<response>		
2				<header>		
3				<requestMsgID/>		
4				<responseTime>2024-11-10 14:15:22</responseTime>		
5				<responseMsgID/>		
6				<successYN>N</successYN><resultCode>		
7				10</resultCode>		
8				<resultMsg>Blocked users.</resultMsg>		
9				</header>		
10				</response>		
11						

서버 – 429 Too Many Requests 응답 로그 기록

INFO: 123.123.123.123:12345 - "GET /mock_api HTTP/1.1" 429 Too Many Requests

데이터 파이프라인 아키텍처



핵심 구현 Task

API 요청 모듈 – 비동기 방식 적용

도입 이유

· 동기 방식 대비 빠른 속도

구분	기업				대학		
종류	특허고객번호	특허/ 실용신안	디자인	상표	특허/ 실용신안	디자인	상표
요청 수	13,019	11,600	11,602	11,605	755	390	391
동기	32분 32초	58분	150분	116분	3분 46초	5분	4분
비동기	11분 2초	9분 50초	24분 5초	43분 57초	1분 25초	51초	1분 30초
대비	2.94배	5.89배	6.22배	2.63배	2.65배	5.88배	2.66배

I/O 처리 효율성 향상에 기인

핵심 구현 Task

API 요청 모듈 – Token bucket 알고리즘 적용

작동 방식

1. 버킷에 일정 간격으로 토큰을 생성
간격 조정으로 토큰 생성 속도 조절 가능
2. 요청 처리 시, 버킷에서 토큰 획득
버킷에 토큰이 없는 경우, 토큰 생성까지 대기 또는 처리 거부
3. 버킷의 최대 크기로 토큰의 과다 생성 방지

장점

구현이 직관적이고 계산이 간단

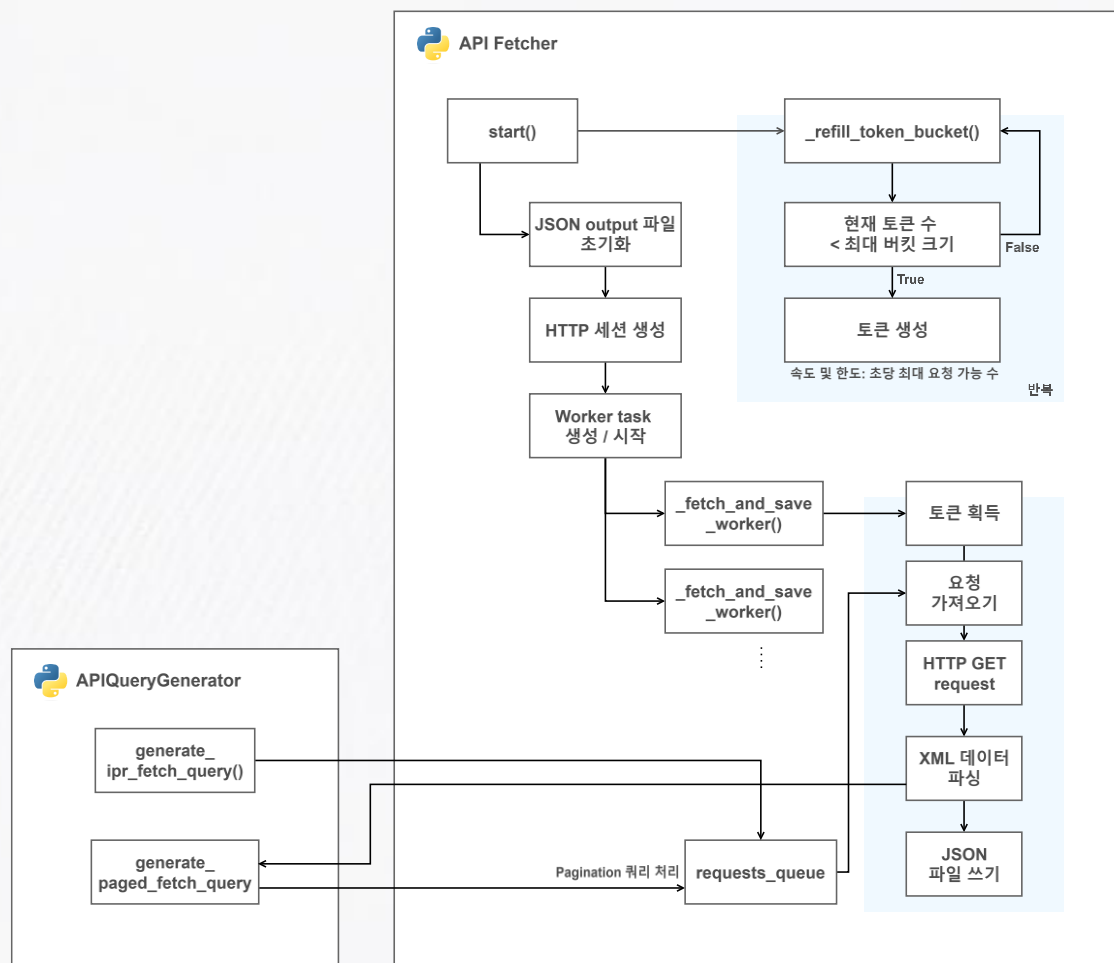
단점

순간적으로 다수의 요청도 허용하는 알고리즘

→ 동시 요청 시, 워커 별 별도 지연 필요



API 데이터 수집 모듈 개념도



핵심 구현 Task

API 요청 모듈 – Raw data 파일 적재

작동 방식

1. 버킷에 일정 간격으로 토큰을 생성
간격 조정으로 토큰 생성 속도 조절 가능
2. 요청 처리 시, 버킷에서 토큰 획득
버킷에 토큰이 없는 경우, 토큰 생성까지 대기 또는 처리 거부
3. 버킷의 최대 크기로 토큰의 과다 생성 방지

장점

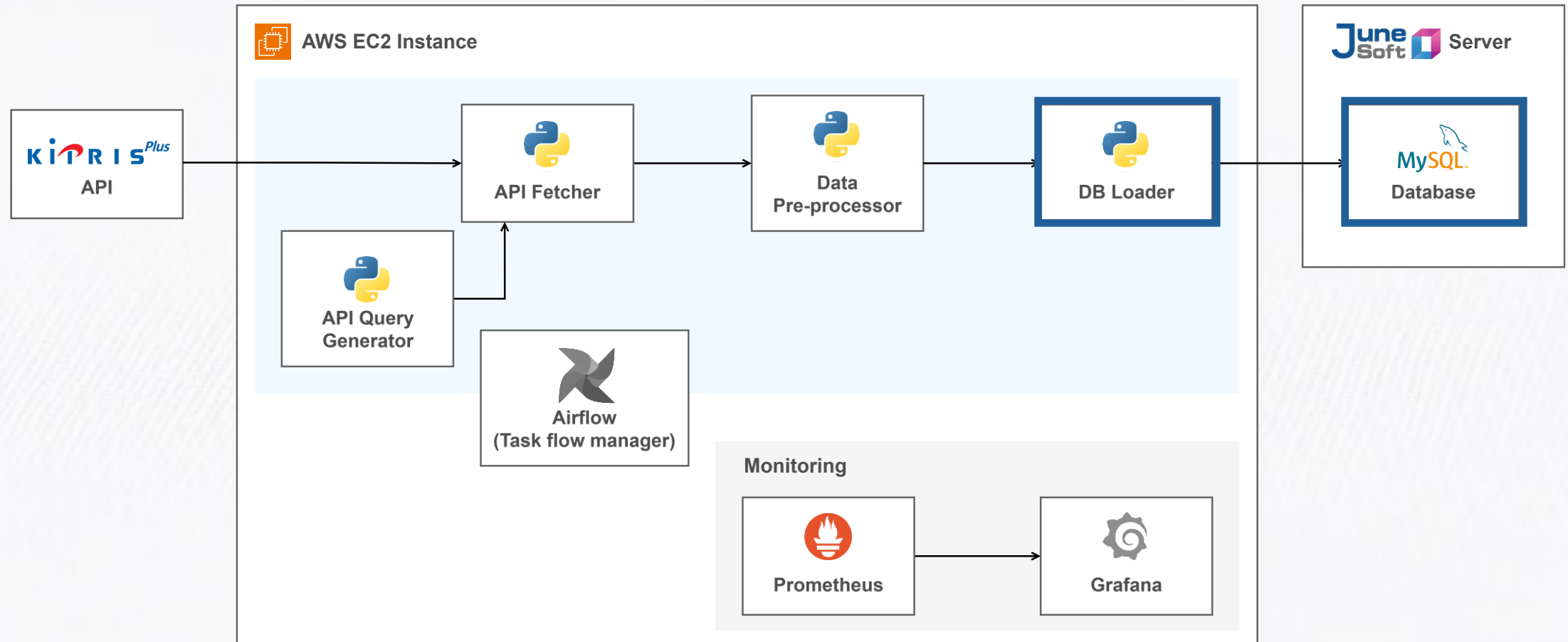
구현이 직관적이고 계산이 간단

단점

순간적으로 다수의 요청도 허용하는 알고리즘

→ 동시 요청 시, 워커 별 별도 지연 필요

데이터 파이프라인 아키텍처



핵심 구현 Task

MySQL DB 적재

기업 제공 reference schema 기반으로 구성

Default value 지정 및 Trigger 적용

- 데이터 생성 일시, 수정 일시
- 조사 연월

Unique key가 없는 테이블의 경우,
2개 이상 컬럼을 조합한
Multi column Index 생성

tb24_100_bizinfo company_seq biz_no corp_no realname_di biz_type head_flag zipcode addr1 addr2 ceo_birth_year company_name found_date ceo_name ceo_mobile biz_tel_no biz_fax_no biz_email standard_code products staff_name staff_dept staff_mobile staff_email kipo_no metro_center_code ipstone_flag ipstone_year ipstone_center_code login_ip login_time company_type del_flag small_biz_type position write_time modify_time department branch_type phone_flag email_flag restriction_flag restriction_period restriction_period_end temporary_key temporary_key_start temporary_key_end tech_field_desc biz_type_desc main_product covid_product applicant_no	tb24_200_corp_applicant applicant_no applicant corp_no biz_no write_time modify_time ref_desc	tb24_300_corp_ipr_reg ipr_seq applicant_no biz_no ipr_code applicant inventor agent main_ipc appl_no appl_date open_no open_date reg_no reg_date pub_no pub_date int_appl_no int_appl_date int_open_no int_open_date legal_status_desc exam_flag exam_date claim_cnt img_url abstract title write_time modify_time survey_year survey_month	tb24_310_ipc_cpc ipc_seq ipr_seq appl_no ipc_cpc ipc_cpc_code	tb24_210_univ_applicant applicant_no applicant corp_no biz_no ref_desc write_time	tb24_400_univ_ipr_reg ipr_seq applicant_no biz_no ipr_code applicant inventor agent main_ipc appl_no appl_date open_no open_date reg_no reg_date pub_no pub_date int_appl_no int_appl_date int_open_no int_open_date legal_status_desc exam_flag exam_date claim_cnt img_url abstract title write_time modify_time survey_year survey_month	tb24_410_ipc_cpc ipc_seq ipr_seq appl_no ipc_cpc ipc_cpc_code
		tb24_320_priority priority_seq ipr_seq applicant_no priority_nation priority_no priority_date			tb24_420_priority priority_seq ipr_seq applicant_no priority_nation priority_no priority_date	

프로젝트 결과

Backfile Raw data 파일 생성

```
-rw-rw-r-- 1 ubuntu ubuntu 2.2M Nov 22 05:59 applicant_no_20241120_corp_values.json
-rw-rw-r-- 1 ubuntu ubuntu 460 Nov 22 06:11 ipc_cpc_20241121_corp_values.json
-rw-rw-r-- 1 ubuntu ubuntu 288 Nov 22 06:10 ipc_cpc_20241121_univ_values.json
-rw-rw-r-- 1 ubuntu ubuntu 44M Nov 22 06:10 ipr_reg_20241121_corp_values.json
-rw-rw-r-- 1 ubuntu ubuntu 4.5M Nov 22 06:10 ipr_reg_20241121_univ_values.json
-rw-rw-r-- 1 ubuntu ubuntu 53 Nov 22 06:10 priority_20241121_corp_values.json
-rw-rw-r-- 1 ubuntu ubuntu 53 Nov 22 06:10 priority_20241121_univ_values.json
```

Preprocessed data 파일 생성

```
-rw-rw-r-- 1 ubuntu ubuntu 2.2M Nov 22 05:59 applicant_no_20241120_corp_values.json
-rw-rw-r-- 1 ubuntu ubuntu 25M Nov 22 07:06 ipc_cpc_20241121_corp_values.json
-rw-rw-r-- 1 ubuntu ubuntu 57M Nov 22 07:06 ipc_cpc_20241121_univ_values.json
-rw-rw-r-- 1 ubuntu ubuntu 191M Nov 22 07:06 ipr_reg_20241121_corp_values.json
-rw-rw-r-- 1 ubuntu ubuntu 329M Nov 22 07:06 ipr_reg_20241121_univ_values.json
-rw-rw-r-- 1 ubuntu ubuntu 4.7K Nov 22 07:06 priority_20241121_corp_values.json
-rw-rw-r-- 1 ubuntu ubuntu 49 Nov 22 07:06 priority_20241121_univ_values.json
```

프로젝트 결과

MySQL DB 적재

각 테이블에 대한 신규 데이터 Upsert 수행 결과

```
tb24_200_corp_applicant 테이블에 업서트 중:
100%|██████████████████████████████████████████████████████████████████████████████| 11643/11643 [00:00<00:00, 74713.22rows/s]
총 11643개의 행이 tb24_200_corp_applicant 테이블에 업서트되었습니다.
tb24_300_corp_ipr_reg 테이블에 업서트 중:
100%|██████████████████████████████████████████████████████████████████████████████| 188310/188310 [00:24<00:00, 7596.57rows/s]
총 188310개의 행이 tb24_300_corp_ipr_reg 테이블에 업서트되었습니다.
tb24_400_univ_ipr_reg 테이블에 업서트 중:
100%|██████████████████████████████████████████████████████████████████████████████| 236478/236478 [00:23<00:00, 10066.25rows/s]
총 236478개의 행이 tb24_400_univ_ipr_reg 테이블에 업서트되었습니다.
tb24_310_ipc_cpc 테이블에 업서트 중:
100%|██████████████████████████████████████████████████████████████████████████████| 332062/332062 [00:11<00:00, 30116.64rows/s]
총 332062개의 행이 tb24_310_ipc_cpc 테이블에 업서트되었습니다.
tb24_410_ipc_cpc 테이블에 업서트 중:
100%|██████████████████████████████████████████████████████████████████████████████| 766520/766520 [00:40<00:00, 19070.26rows/s]
총 766520개의 행이 tb24_410_ipc_cpc 테이블에 업서트되었습니다.
tb24_320_priority 테이블에 업서트 중:
100%|██████████████████████████████████████████████████████████████████████████████| 57/57 [00:00<00:00, 791640.16rows/s]
총 57개의 행이 tb24_320_priority 테이블에 업서트되었습니다.
업서트할 데이터가 없습니다.
테스트 완료
```

프로젝트 결과

MySQL DB 적재

DB 내 각 테이블에 대한 Rows 수 출력

```
SELECT
TABLE_NAME AS `Table`,
TABLE_ROWS AS `Rows`
FROM
information_schema.TABLES
WHERE
TABLE_SCHEMA = 'kipris'
ORDER BY
TABLE_ROWS DESC;
```

출력 결과

```
+-----+-----+
| Table                | Rows |
+-----+-----+
| tb24_310_ipc_cpc      | 324056 |
| tb24_410_ipc_cpc      | 211120 |
| tb24_400_univ_ipr_reg | 168037 |
| tb24_300_corp_ipr_reg | 161849 |
| tb24_100_bizinfo      | 21005  |
| tb24_200_corp_applicant | 12122  |
| tb24_210_univ_applicant | 389    |
| tb24_320_priority     | 23     |
| tb24_420_priority     | 0      |
+-----+-----+
9 rows in set (0.01 sec)
```

향후 개선 필요 사항

미완료 tasks 처리

- 보안 문제로 유실된 task 결과물 재현 필요
 - Airflow DAG
 - Grafana 모니터링 대시보드
 - Streamlit 데이터 분석 대시보드

Storage HA 확보

- 고가용성 확보를 위해 AWS S3 버킷으로 Raw data 보관 필요

로그 분석 및 모니터링 고도화

- ELK stack 도입으로 에러 패턴 분류, 성능 병목 구간 파악
- 애플리케이션 관련 metrics

감사합니다!