

□ Superb AI

카프카를 활용한 실시간 데이터 동기화 파이프라인 (feat. CDC)

2022.08.25, 유용환

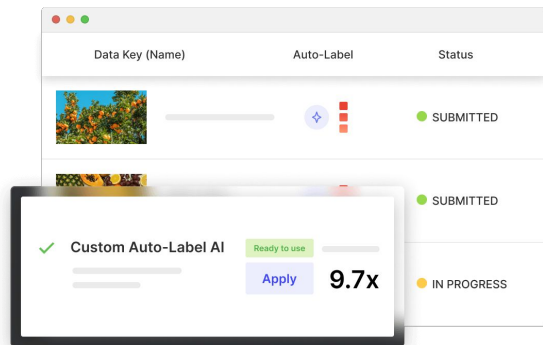
01.

Introduction

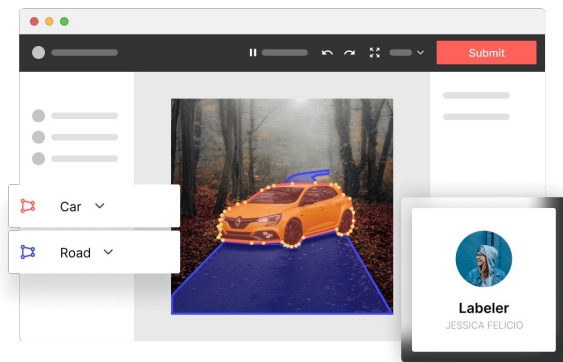
Superb AI Suite

컴퓨터 비전 분야(이미지, 비디오, 라이다 등)의 AI 학습을 위한 데이터 솔루션

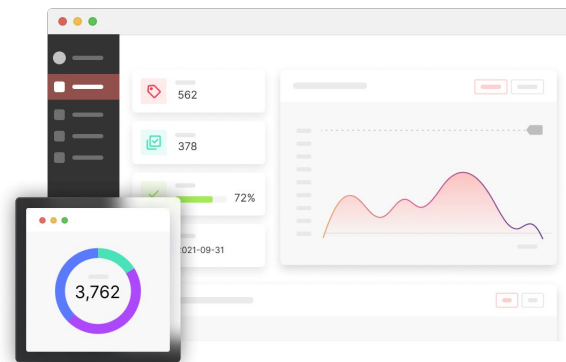
Automated Labeling



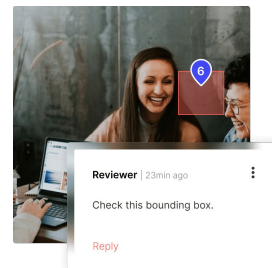
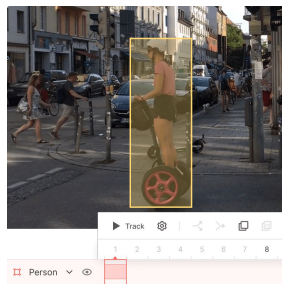
Manual Labeling



Label Analytics

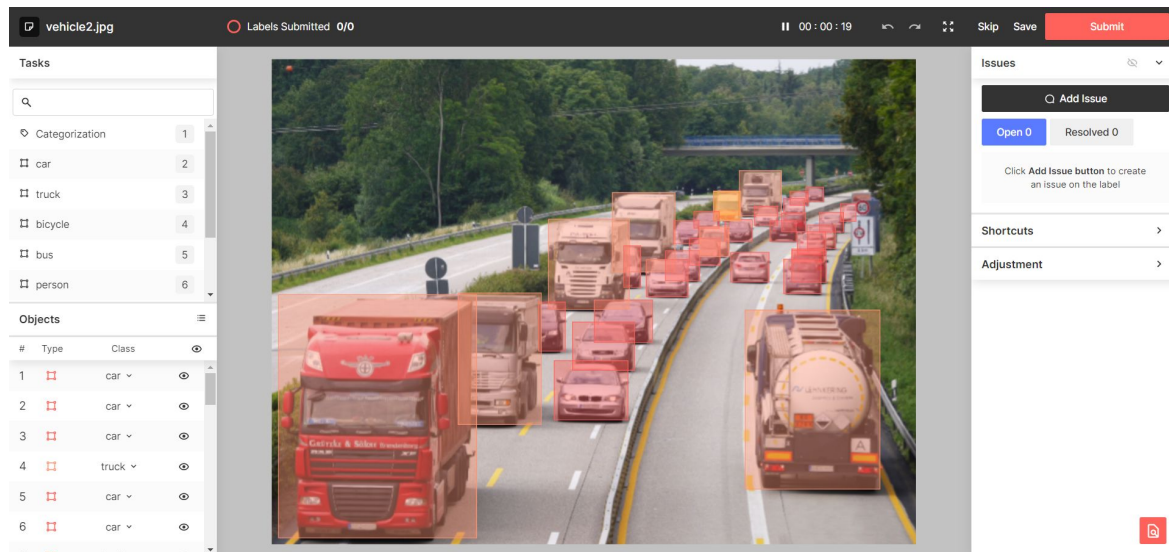


Data Key (Name)	Auto-Label	Status
Bus_1.jpg		SUBMITTED
Bus_2.jpg		SUBMITTED
Bus_3.jpg		IN PROGRESS
Bus_4.jpg		IN PROGRESS



Superb AI Suite

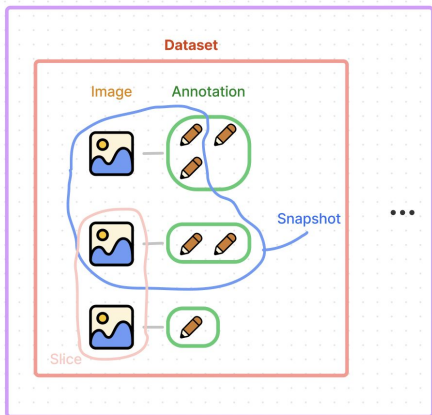
다음 마일스톤: AI 데이터셋의 효율적인 관리



<https://docs.superb-ai.com/docs/label-mode-overview>

Superb AI Suite

(신규 기능) 이미지 검색



Client 1

```
import sdk

dataset = sdk.create_dataset(...)

image = dataset.upload_image(...)

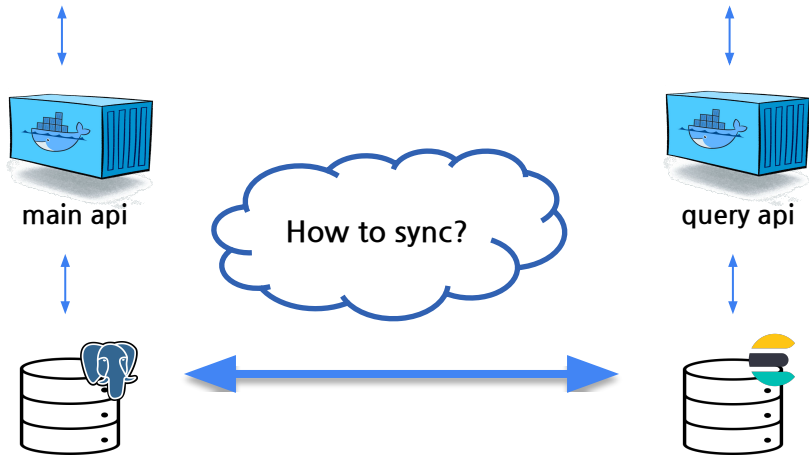
image.add_metadata(...)

image.add_annotations(...)
```

Client 2

find all images where...

- width \geq 300, height \geq 200
- metadata.timeofday = night
- annotation.car $>$ 0, annotation.metadata.occluded = true
- ...



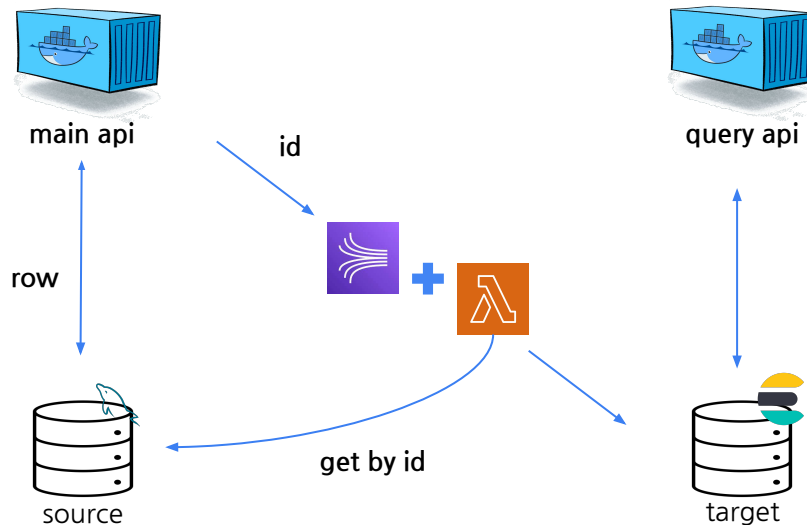
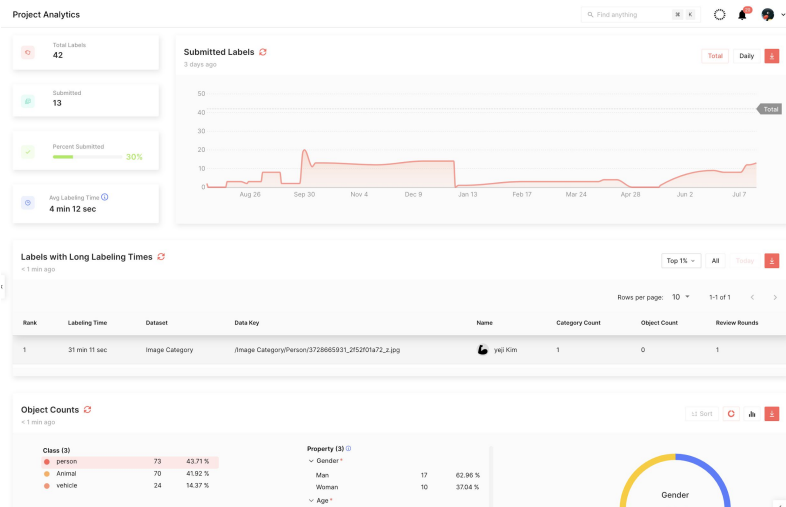
02.

Base Architecture

Previous Approach

소스 애플리케이션에서 양쪽 DB를 업데이트

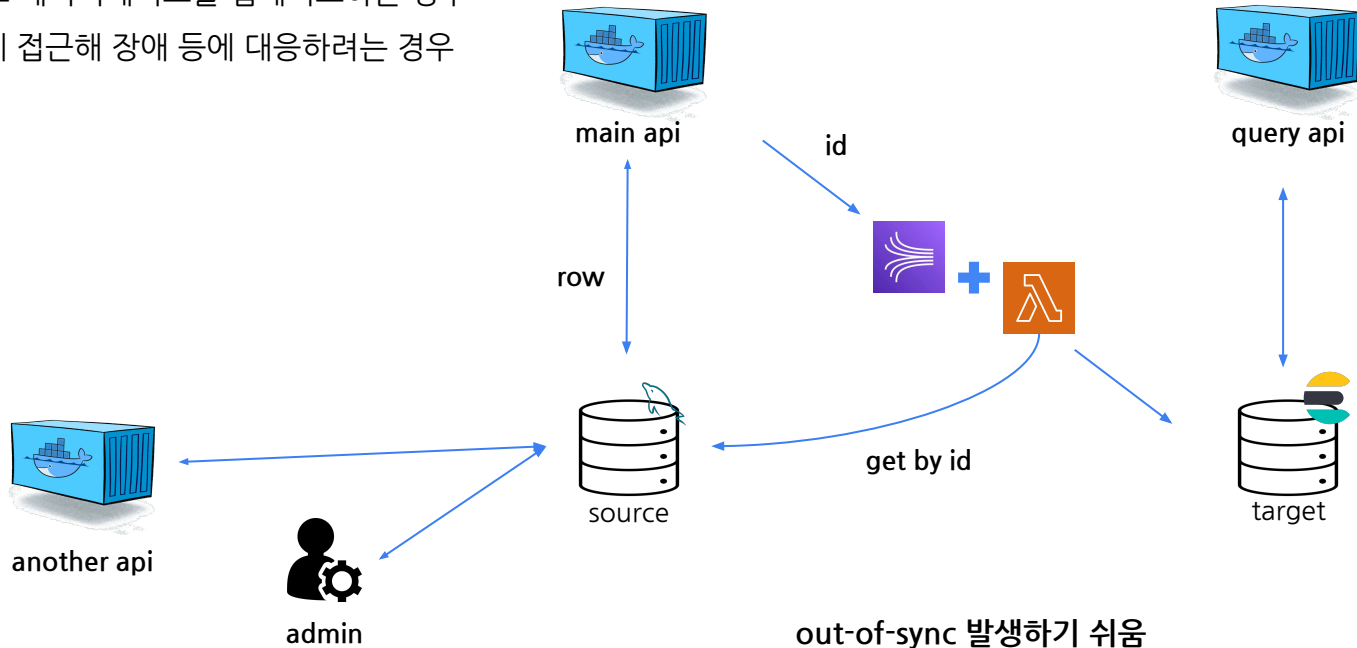
Label Analytics



Previous Approach

문제점: 소스 애플리케이션에 대한 종속성

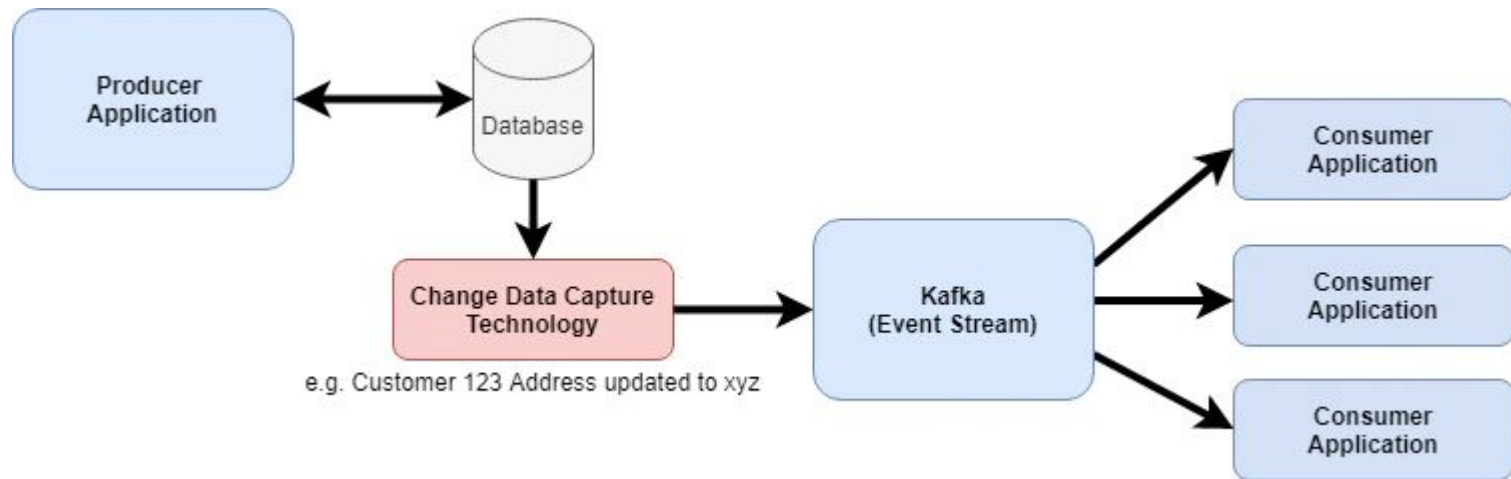
- 다른 api 서버가 소스 데이터베이스를 업데이트하는 경우
- 직접 데이터베이스에 접근해 장애 등에 대응하려는 경우



New Approach

CDC (change data capture)

- DB 변화를 추적하여 액션을 취한다



New Approach

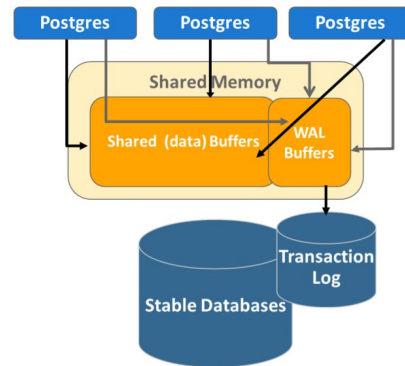
CDC (change data capture)

- 어떻게 DB 변화를 감지할 것인가?

1. Database 쿼리 (ex, LogStash agent로 폴링하기)
 - performance 이슈, reliability 이슈 (ex, delete 캡처 불가)
2. Database 트리거 (ex, Postgres CREATE TRIGGER)
 - performance 이슈, trigger 추가를 위한 database 설정 필요
3. Transaction Logs (ex, Postgres WAL Logs)

Write Ahead Logging (WAL)

- Backend write data to WAL buffers
- Flush WAL buffers periodically (WAL writer), on commit, or when buffers are full
- Group commit



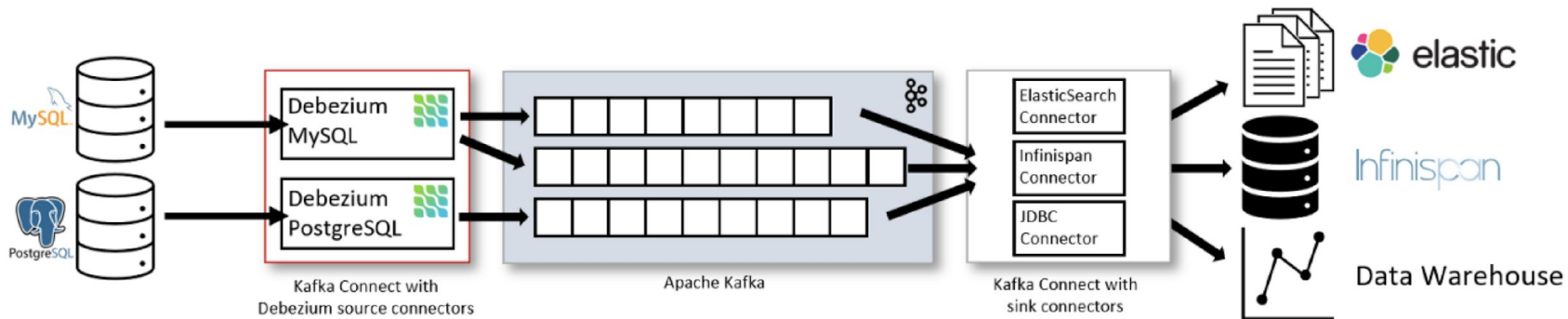
© EnterpriseDB® Corporation. All rights reserved.

https://pgadminedb.readthedocs.io/en/latest/module_02/#write-ahead-logging-wal

Architecture

CDC with Debezium (Kafka Source Connector)

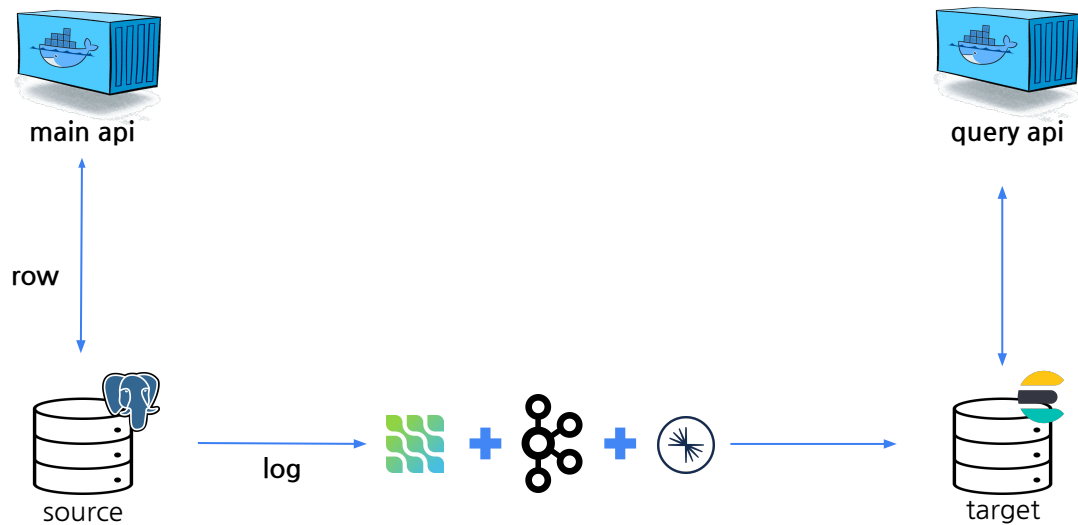
- 소스 DB 커넥터(ex, debezium source connector)를 통해 Transaction Log를 메시지 브로커(ex, Kafka)로 스트리밍
- 타겟 DB 커넥터(ex, elasticsearch sink connector)통해 메시지 브로커에 쌓인 데이터를 다른 DB에 실시간으로 동기화



https://pgadminedb.readthedocs.io/en/latest/module_02/#write-ahead-logging-wal

Architecture

CDC with Debezium (Kafka Source Connector)

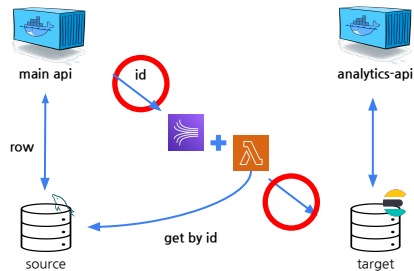


Architecture

CDC with Debezium (Kafka Source Connector)

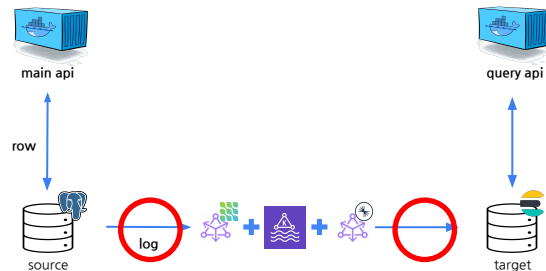


Trade-Offs



기존 아키텍처

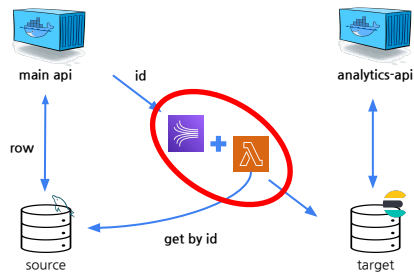
- 완전 관리형 서비스인 kinesis 사용하여 운영 비용 절감
- 기존 스택(람다)으로 자유로운 데이터 조작 구현 가능
- 소스 애플리케이션에 대한 강한 의존성(강한 커플링)
- 람다 핸들러에 DB 커넥션 등 인프라 관련 로직 필요



신규 아키텍처

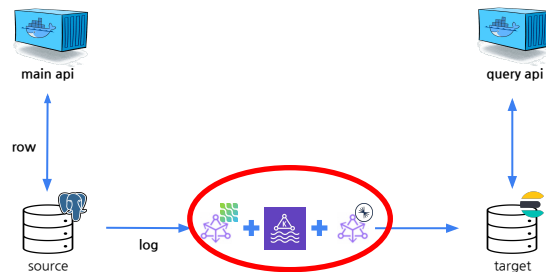
- DB를 source of truth로 하여 애플리케이션 의존성 제거
- 오픈소스 DB 커넥터(low code)로 인프라 관련 로직 대체
- kafka 인프라에 대한 높은 운영 비용 (ex, 브로커, 파티션)
- 새로운 스택(kafka connect) 도입으로 인한 구현의 제약

Trade-Offs



기존 아키텍처

- 완전 관리형 서비스인 kinesis 사용하여 운영 비용 절감
- 기존 스택(람다)으로 자유로운 데이터 조작 구현 가능
- 소스 애플리케이션에 대한 강한 의존성(강한 커플링)
- 람다 핸들러에 DB 커넥션 등 인프라 관련 로직 필요



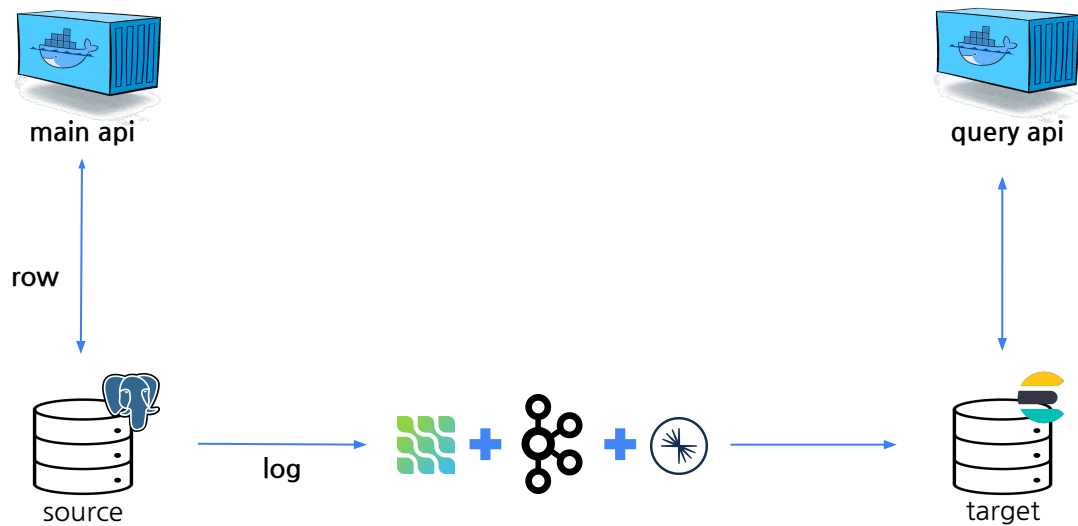
신규 아키텍처

- DB를 source of truth로 하여 애플리케이션 의존성 제거
- 오픈소스 DB 커넥터(low code)로 인프라 관련 로직 대체
- kafka 인프라에 대한 높은 운영 비용 (ex, 브로커, 파티션)
- 새로운 스택(kafka connect) 도입으로 인한 구현의 제약

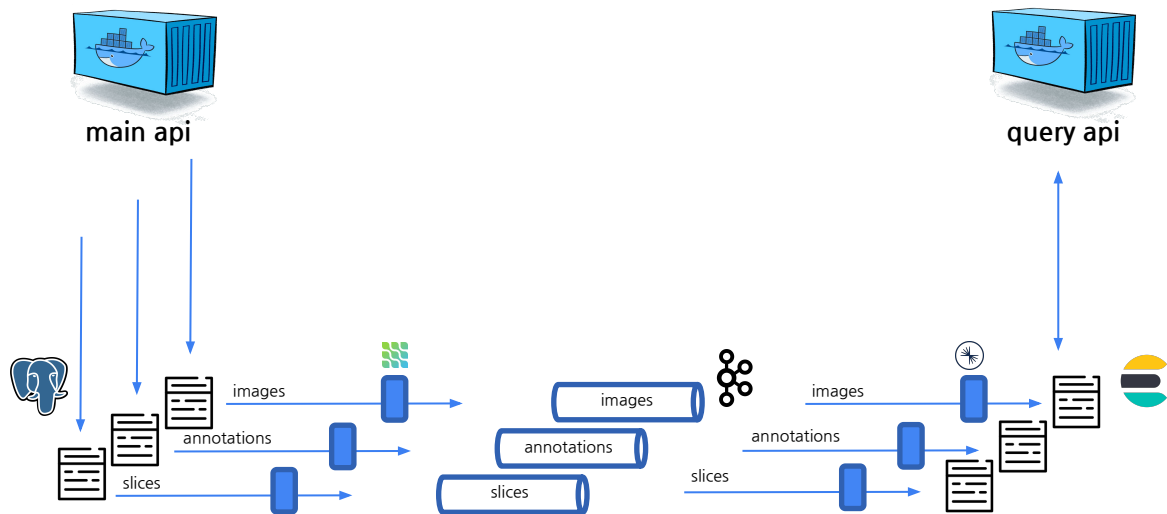
03.

Challenges

JOIN Query

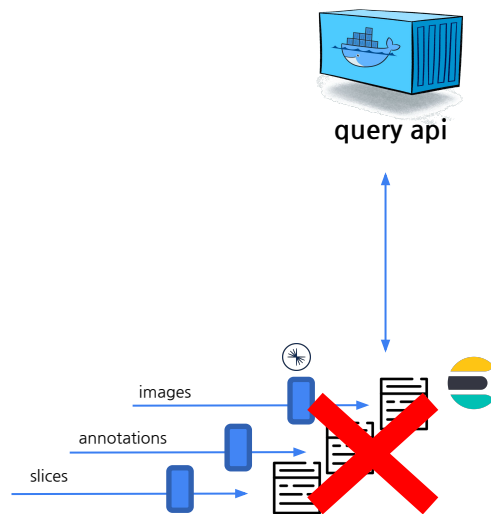


JOIN Query



JOIN Query

```
SELECT *  
FROM image i  
LEFT JOIN annotations a ON i.id = a.image_id  
WHERE i.metadata ->> 'timeofday' = 'daytime'  
AND a.metadata ->> color = 'White'
```



JOIN Query

Elastic Docs > Elasticsearch Guide [8.3] > Query DSL

Joining queries

Performing full SQL-style joins in a distributed system like Elasticsearch is prohibitively expensive. Instead, Elasticsearch offers two forms of join which are designed to scale horizontally.

nested query

Documents may contain fields of type `nested`. These fields are used to index arrays of objects, where each object can be queried (with the `nested` query) as an independent document.

has_child and has_parent queries

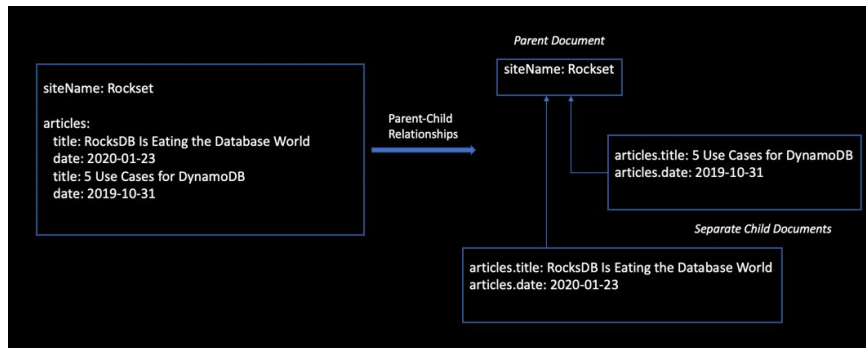
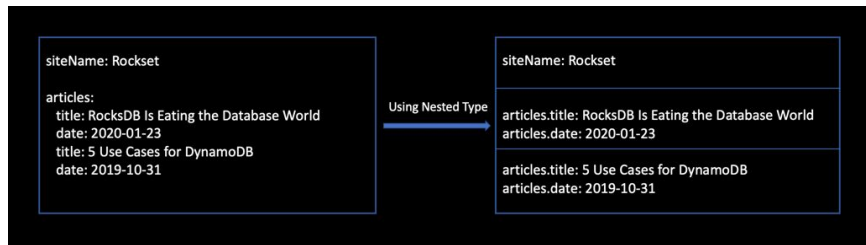
A `join` field relationship can exist between documents within a single index. The `has_child` query returns parent documents whose child documents match the specified query, while the `has_parent` query returns child documents whose parent document matches the specified query.

Also see the `terms-lookup` mechanism in the `terms` query, which allows you to build a `terms` query from values contained in another document.

Notes

Allow expensive queries

Joining queries will not be executed if `search.allow_expensive_queries` is set to false.



<https://www.elastic.co/guide/en/elasticsearch/reference/current/joining-queries.html>

JOIN Query

But in reality...

Document modeling



Documents should be modeled so that search-time operations are as cheap as possible.

In particular, joins should be avoided. **nested** can make queries several times slower and **parent-child** relations can make queries hundreds of times slower. So if the same questions can be answered without joins by denormalizing documents, significant speedups can be expected.

Elastic Cloud) 둘 다 추천하지 않는 구조입니다. 데이터 모델링을 다시 하시는게 좋을 듯합니다

아니; 이렇거면 기능은 왜 만드신 건가요? $\pi\pi$

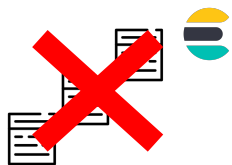


<https://www.elastic.co/guide/en/elasticsearch/reference/current/tune-for-search-speed.html>

JOIN Query

TODO List

1. Document Modeling (인덱스 구조 재설계)



2. Aggregation (그 구조의 데이터를 만들 방법)



Document Modeling

Solution: 역정규화 + 검색용 필드 추가

image

```
{
  "docs": [
    {
      "_source": {
        "id": "im_01G8T6ERR7ZSESAGN8PFM8HGR4",
        "created_at": "2022-07-25T08:09:27.079Z",
        "updated_at": "2022-07-25T08:09:27.079Z",
        "deleted_at": null,
        "key": "string",
        "dataset_id": "ds_01G8T6DKMBRQVCSRZ6DD79VYY",
        "metadata": {},
        "attributes": {}
      }
    }
  ]
}
```

annotation

```
{
  "docs": [
    {
      "_source": {
        "id": "an_01G6QP3XB58VSXAFM011JAMMY",
        "created_at": "2022-06-29T12:14:01.603Z",
        "updated_at": "2022-06-29T12:14:01.603Z",
        "deleted_at": null,
        "image_id": "im_01G6Q9NPHFMS3B49BBV7W0ANM2",
        "dataset_id": "bu_01G4Y1F6242WVSSJGA75C8BGSQ",
        "annotation_type": "bbox",
        "annotation_class": "some-class",
        "metadata": {
          "train": "true"
        },
        "coord": { }
      }
    }
  ]
}
```

```
{
  "docs": [
    {
      "_source": {
        "id": "im_01G8T6ERR7ZSESAGN8PFM8HGR4",
        "created_at": "2022-07-25T08:09:27.079Z",
        "updated_at": "2022-07-25T08:09:27.079Z",
        "deleted_at": null,
        "key": "string",
        "dataset_id": "ds_01G8T6DKMBRQVCSRZ6DD79VYY",
        "metadata": {},
        "attributes": {},
        "annotations": [
          {
            "annotation_class": "some-class",
            "annotation_type": "bbox",
            "metadata": {
              "train": "true"
            }
          }
        ],
        "slices": [
          {
            "slice_id": "sl_01G8T6Z91TPPFMQZTGNMDSDBZS",
            "slice_name": "some-slice"
          }
        ]
      }
    }
  ]
}
```

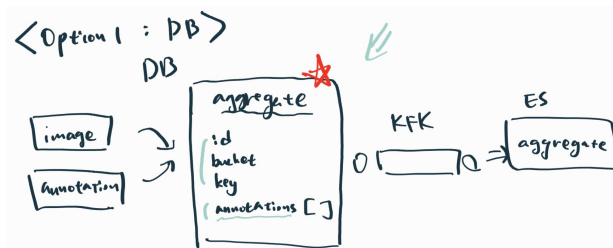
image_enriched

```
{
  "docs": [
    {
      "doc": {
        "_index": "_index",
        "_type": "_doc",
        "_id": "_id",
        "_source": {
          "slices": [
            {
              "slice_name": "some-slice",
              "slice_id": "sl_01G8T6Z91TPPFMQZTGNMDSDBZS"
            }
          ],
          "metadata": { },
          "dataset_id": "ds_01G8T6DKMBRQVCSRZ6DD79VYY",
          "created_at": "2022-07-25T08:09:27.079Z",
          "annotations": [
            {
              "annotation_class": "some-class",
              "metadata": {
                "train": "true"
              },
              "annotation_type": "bbox"
            }
          ],
          "deleted_at": null,
          "updated_at": "2022-07-25T08:09:27.079Z",
          "annotations_search": {
            "count": 1,
            "class_count": {
              "some-class": 1
            },
            "type_count": {
              "bbox": 1
            }
          },
          "attributes": { },
          "id": "im_01G8T6ERR7ZSESAGN8PFM8HGR4",
          "key": "string"
        },
        "_ingest": {
          "timestamp": "2022-07-27T14:35:52.739585122Z"
        }
      }
    }
  ]
}
```

Aggregation

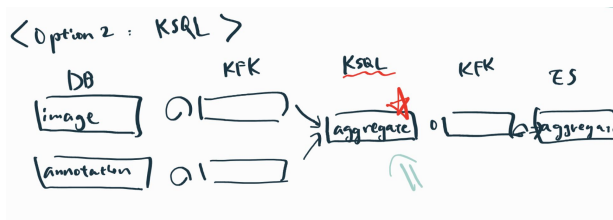
Options

1. CDC on Materialized View



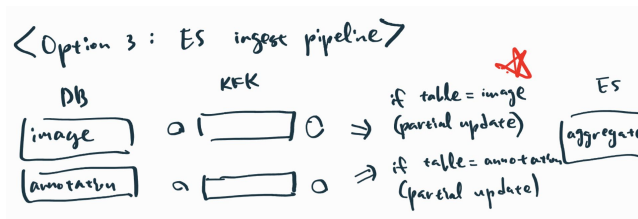
performance issue
not supported in Postgres

2. KTable foreign key join



resource requirement
RocksDB limitation

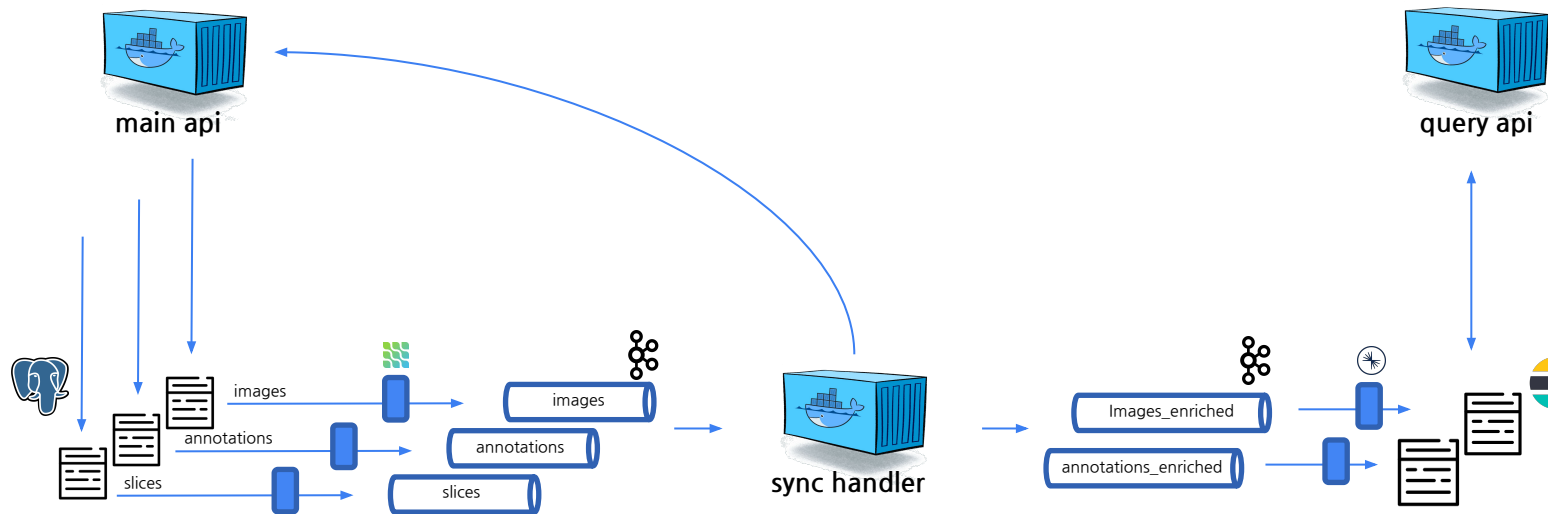
3. Elasticsearch ingest pipeline



reliability issue (partial update)
maintenance issue

Aggregation

Final Architecture



□ Superb AI

THANK YOU.

Global Headquarters
400 Concar Drive, San Mateo, CA, USA

Asia-Pacific Office
14 Teheran-ro 4-gil, Gangnam-gu, Seoul, Korea