소개

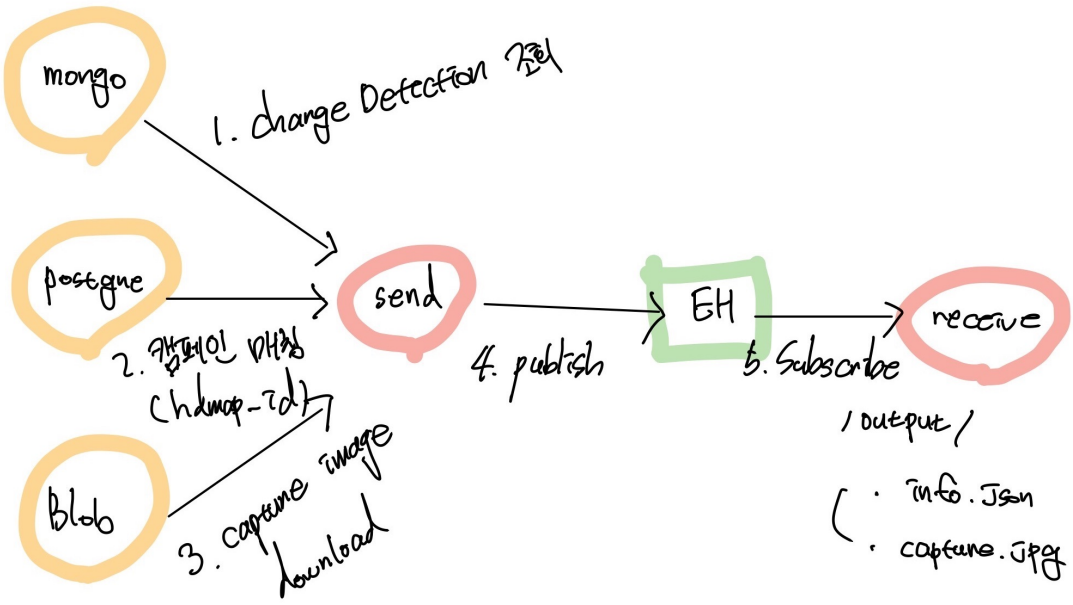
crowd sourcing 기반으로 수집된 ROD 데이터를 클러스터링 하고, HD Map Update 알고리즘을 통해 Change Detection 결과를 Azure Eventhub(Pub/Sub) 를 통해 공유하기 위한 인터페이스를 개발한다.

서울시 CITS 전체 구간을 3주동안 차량 1대로 10회 반복 주행하여 수집한 ROD를 클러스터링 하였고, 이를 통해 HDMap Update 알고리즘을 실행한 Add/Delete 후보군 중에 Confidence 값이 높은 후보군을 대상으로 NZERO 에게 제공한다. 이 결과 14건의 검증된 Add Candidate 가 생성되었고, Delete 의 경우 3주 동안의 짧은 기간이 생성되지 않았다. 향후, 서울시 버스 1700 여대의 ROD 데이터를 통해 오랜 시간동안 데이터를 누적한다면 더 많은 Change Detection 결과물이 생성될 것으로 기대한다.

Features

- Azure Eventhub 를 통해 Change Detecion 정보를 NZERO와 공유하기 위한 인터페이스 제공
- CITS 3주치 테스트 차량 1대 데이터만 가공 (관측횟수 <= 10건)
- add candidate 14건에 대한 사전 검증 및 NZERO 제공 (street view 이미지로 검증 완료)
- 실데이터 수집시 파라미터 변경 (관측횟수 60건) 및 알고리즘 적용 예정

API Flow



1. change detection 조회 mongodb 저장된 add/delete candidate 정보를 주기적으로(1d) 조회
2. 캠페인 매핑 (**Option**) add/delete candidate 정보와 매칭되는 (/w hdmap_id) 캠페인 데이터 수신
3. Capture Image Download (**Option**) 캠페인 데이터 안에 포함된 차량에서 캡쳐하여 올린 사진 이미지를 blob 에서 다운로드
4. Publish send module(send_candidate.py) 에서 change detecion 속성 정보(info.json) 와 캡쳐 사진 (capture.jpg) 이미지를 protobuf 시리얼라이징을 통해 EH(EventHub) 로 Publish
5. Subscribe receive module(recv_candidate.py) 에서 데이터 수신후 local /output 경로명에 각 change detecion 속성 정보(info.json) 와 캡쳐 사진 저장 (하단 디렉토리 구조 참고)

Database

Protobuf

campaign.proto

```
// SKT campaign data definitions
syntax = "proto2";
package campaign;

message Image {
    required string type = 1;   //"jpeg" , "yuv422" , "yuv420"
    optional bytes  image_data = 2;
    optional string blob_container = 3;
    optional string blob_dir = 4;
    optional string blob_file_nm = 5;
}

// ****************************************
// [[[[ Definiton of Campaign ]]]]
// ****************************************
message CampaignPacket {
    required string ver = 1;
    required string type = 2; // add or delete
    required string hdmap_id = 3; // skt hdmap unique_id
    optional int32 dl_cnt = 4; // observed discrete LM count
    optional float observe_rate = 5; // observe_cnt / travel_cnt
    required string category = 6; // LM category
    required int32 attribute = 7; // LM attribute
    required float x = 8; // Landmark coordinate x
    required float y = 9; // Landmark coordinate y
    optional float z = 10; // Landmark coordinate z
    optional float heading = 11; // Landmark heading
    optional Image image = 12; // campaign image (option)
}
// [END messages]
```

Protobuf Compile

```
protoc -I=./ --python_out=./ ./campaign.proto
```

컴파일 이후 campaign_pb2.py 파일 생성 되고, recv/send_candidate.py 파일에서 API 통신하기 위해 import 하여 사용

MongoDB (Azure CosmosDB)

DB Strcuture

- candidate
  - add_candidate
  - del_candidate

PostgreSQL (Azure PostgreSQL)

DB Structure

- hdmap
  - campaign_scenario
  - campaign_blob_info

add/delete 후보군의 hdmap_id 를 이용하여 현재 진행중인 캠페인 데이터(캡처 이미지)를 맵핑하여 가져오는 쿼리문

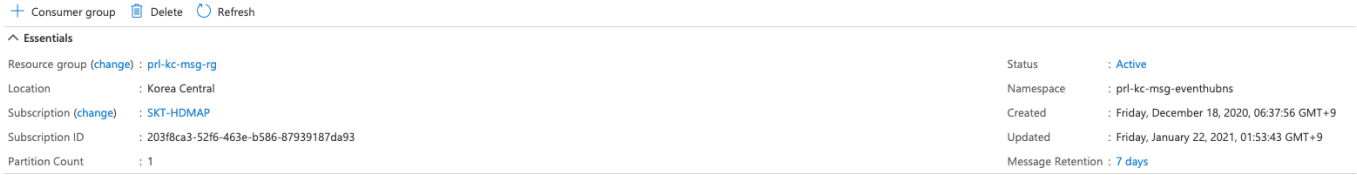
```
sql = '''
    select a.obj_key as hdmap_id, b.*
    from campaign_scenario a, campaign_blob_info b
    where a.obj_key = '{hdmap_id}' and a.campaign_id = b.campaign_id;
    '''.format(hdmap_id=hdmap_id)
```

Blob Strcuture

- Blob (prldrodsa) file strcuture: prldrodsa/campaign/device_id/campaign_id/capture.jpg

Eventhub

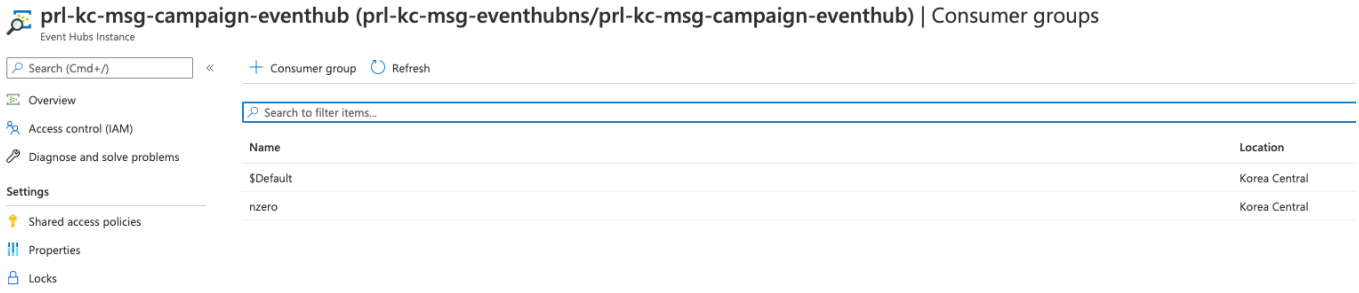
prl-kc-msg-campaign-eventhub 기본정보



- path: prl-kc-msg-eventhubns/prl-kc-msg-campaign-eventhub
- resource group: prl-kc-msg-rg
- namepsace: prl-kc-msg-eventhubns
- message retention: 7days (**NZERO 에서 7d 이내에 데이터 수신 필요**)

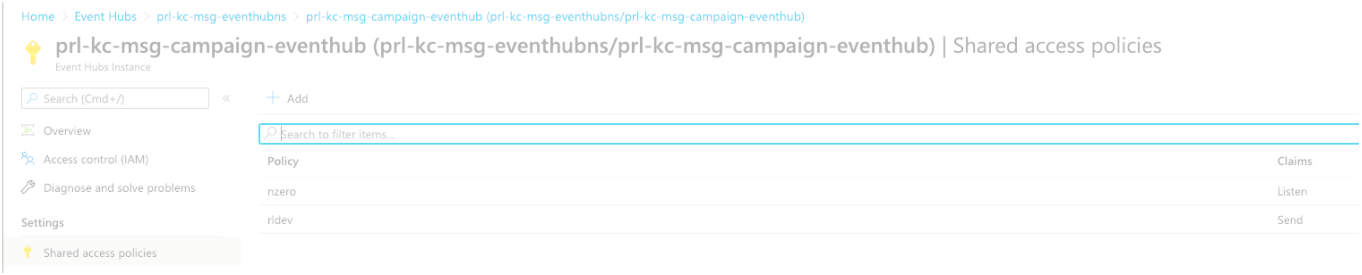
Consumer Group

- consumer_group: nzero



SAP (Shared Access Policies)

- nzero (listen)
- rldev (send)



사용법

테스트 환경 설정 (공통)

1. python install (python version: 3.6.8) 가능하면 virtualenv 환경에서 실행할 것을 추천 pyenv 이용한 virtualenv 설치방법

2. python 가상환경 activation

```
pyenv activate {your project nane}
```

3. install python package

```
pip install -r requirements.txt
```

4. verify whether installation is ok

```
pip freeze
```

Publish

실행방법

1. Azure Credential 환경변수 정의 (중요 정보이므로 .gitignore 를 통해 repo 관리에서 제외하고, 별도 관리자에게 공유 예정)

```
source skt_secret.sh
```

- 주요 접속정보
  - eventhub
  - blob
  - postgresql
  - mongodb

2. Change Detection 전송

```
python send_candidate.py

2021-01-24 22:36:01,007 - send_candidate.py - INFO - mongodb connected
2021-01-24 22:36:01,476 - send_candidate.py - INFO -
****************************
2021-01-24 22:36:01,476 - send_candidate.py - INFO - start to sending add
candidate
2021-01-24 22:36:01,476 - send_candidate.py - INFO -
****************************
2021-01-24 22:36:01,495 - send_candidate.py - INFO -
****************************
2021-01-24 22:36:01,495 - send_candidate.py - INFO - start to sending del
candidate
2021-01-24 22:36:01,495 - send_candidate.py - INFO -
****************************
2021-01-24 22:36:02,305 - send_candidate.py - INFO - ver: "0.1"
type: "del"
hdmap_id: "557631910F02N000123"
observe_rate: 0.23999999463558197
category: "signal"
attribute: 502
x: 320825.3125
y: 4158853.5
z: 62.584999084472656
heading: -1.0
```

Change Detection 전송 조건

Add Candidate

- status = 'I' (HDMap Update 알고리즘을 적용하여, Confidence 값이 일정 임계치 이상인 후보군에 대해 I(Insert) 상태로 업데이트 된 후보군들)
- trsfer_chk = 0 (한번도 NZERO 에 전송이 안된 후보군들 대상으로만 전송)
- query 조건

```
    cursor = db.add_candidate.find({"status": 'I', "trsfer_chk": {"$lt":
'1'}})
```

Delete Candidate

- 주행횟수(travel_cnt) > 30
- 관측율(observe_rate) = 관측횟수 (observe_cnt) / 주행회수 (travel_cnt) < 30% 이하인 경우. (파라미터 튜닝 필요)
- query 조건

```
cursor = db.del_candidate.aggregate([
      {
        "$match": {
          "travel_cnt": {"$gt": 30}
        }
      },
      {
        "$addFields": {
            "observe_rate":{"$divide": ["$observe_cnt", "$travel_cnt"]}
        }
      }
      ])
```

Subscribe

Azure Eventhub Pub/Sub 구조를 통해 SKT 가 Change Detecion(Add/Del Candidate) 정보를 제공하고, NZERO 에서 Subscribe 하여 데이터를 수신하는 구조 입니다. 현재 Change Detecion 정보를 이미 Azure Eventhub 에 Publish 했고, 아래와 같은 방법을 통해 테스트 가능합니다.

1. Azure Credential 환경변수 정의 (중요 정보이므로 .gitignore 를 통해 repo 관리에서 제외하였고, NZERO에 안전한 채널을 통해 전달)

```
source nzero_secret.sh
```

2. Change Detecion 수신

- activate virtualenv

```
pyenv activate {your virtualenv}
```

- usage

eventhub message retention: 7days 이므로 **NZERO 에서 7d 이내에 데이터 수신 필요**

```
# 사용법
python recv_candidate.py <date>

# 현재 시점부터 구독
# 현재 시점에 Publish 하는 데이터가 없으므로 아무것도 받지 않는 대기 상태
python recv_candidate.py

# 현재 날짜 기준으로 < 7d 이내, 특정 날짜부터 구독 (재실행시 overwrite)
# 즉, 현재날짜 기준 7일 이내이고, 2021/1/26 일부터 Publish 된 데이터를 구독하여 테스트 가능. (현재 해당 날짜에 데이터 publish 되어 있어 테스트 가능함.)
python recv_candidate.py 2021/1/26
```

3. /output 구조

위 실행을 통해 /output 폴더에 Change Detecion 정보들을 확인할 수 있다. 서울시 CITS 전체 구간을 3주 동안 차량 1대로 10회 반복 주행하여 수집한 ROD를 클러스터링 하였고, HDMap Update 알고리즘을 통해 총 14건의 Add Candidate 생성되었다. Lnadmark에 대한 속성 정보는 info.json 을 통해 확인 가능하고, 검지를 위해 capture.jpg(option) street view 이미지를 활용할 수 있다.

- File Structure

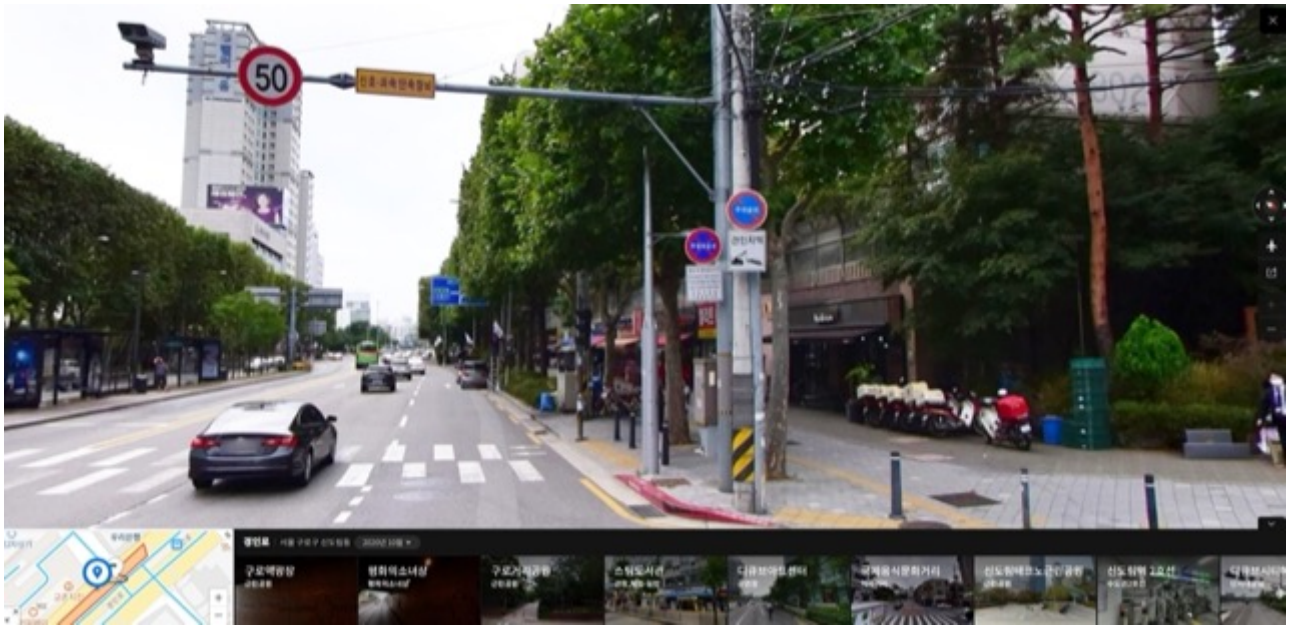
```
add
├── 557631708F01N003563
│   ├── capture.jpg
│   └── info.json
├── 557631906F01N007442
│   ├── capture.jpg
│   └── info.json
├── 557631906F01N007443
│   ├── capture.jpg
│   └── info.json
├── 557631913F01N004903
│   ├── capture.jpg
│   └── info.json
├── 557631913F01N004904
│   ├── capture.jpg
│   └── info.json
├── 557631928F01N005951
│   ├── capture.jpg
│   └── info.json
├── 557631928F01N005952
│   ├── capture.jpg
│   └── info.json
├── 557631928F01N005953
│   ├── capture.jpg
```

```
│   └── info.json
├── 557631929F01N003858
│   ├── capture.jpg
│   └── info.json
├── 557631930F01N001097
│   └── info.json
├── 557631933F01N003868
│   ├── capture.jpg
│   └── info.json
├── 557632273F01N002362
│   ├── capture.jpg
│   └── info.json
├── 557632273F01N002363
│   ├── capture.jpg
│   └── info.json
└── 557632273F01N002364
    ├── capture.jpg
    └── info.json
```

- info.json (output/add) sample

```
{
  "ver": "0.1",                        # version
  "type": "add",                       # event type (add or delete)
  "hdmapId": "557631708F01N003563",    # SKT HDMap ID
  "dlCnt": 11,                         # 관측된 Landmark 수
  "category": "sign",                  # Landmark 카테고리
  "attribute": 215,                    # Landmark 속성
  "x": 312917.94,                      # UTM_52S East
  "y": 4153088.2,                      # UTM_52S North
  "z": 36.1615,                        # UTM_52S Altitude
  "heading": 232.0                     # Landmark heading
}
```

- capture image (street view image)



- info.json (output/del) sample CITS 3주 데이터를 통해서는 del candidate 는 생성되지 않았지만, 생성시 아래와 같은 format 으로 구성

```
{
  "ver": "0.1",
  "type": "del",
  "hdmapId": "557631910F01N000387",
  "observeRate": 0.12,                # 관측율 = (관측횟수 / 주행횟수)
  "category": "sign",
  "attribute": 399,
  "x": 320323.28,
  "y": 4159710.2,
  "z": 65.507,
  "heading": -1.
}
```