Steps for PGR live indexing

Promotion Steps

- Add https://github.com/egovernments/configs/blob/master/egov-indexer/pgr-services.yml to indexer.

For PGR module a new index needs to be created. Please refer to the indexer configuration here https://github.com/egovernments/configs/blob/master/egov-indexer/pgr-services.yml .

- Index for complaints (index name “pgr-services”).

- **Index for complaints (“pgr-services“)**
    - It will index complaints.
    - The records will be picked up from the following kafka topics:-
        - save-pgr-request:- The records on this topic are newly created complaints.
        - update-pgr-request:- The records on this topic are for updating existing complaints.
        - pgr-services-legacyIndex:- Records on this topic are for reindexing old complaints.
    - Every time a new complaint is created, new record is pushed to elastic search index. If existing complaint is updated, then the existing complaint on the index is updated. So this index stores latest information for every serviceRequestId.

Updates from previous indexing structure

- All PIIs are masked in the index.
- All the fields belonging to service and workflow information are now separated out in their respective object wrappers.
- The structure of ward, tenantData, slaHours and department fields remains the same.

Impact on Dashboard

- Total count of complaints will be total number of records on pgr-services index.
- As the fields are now segregated into service and workflow objects, respective fields can be fetched by accessing them from within the aforementioned wrapper objects.

- Application statuses have been changed from PGR v1 to v2 and corresponding changes are made in the dashboard queries to get application status wise charts.

PGR Services: Tenant and date range wise re-indexing Utility:

- Need to have criteria based re-indexing because in any production environments we would not be doing or needing entire data reindexing.
- It would be for either for one tenant or for certain criteria like all complaints within a date range.
- Added a plainsearch API endpoints for multiple tenantIds and date range wise reindexing.
- fromDate and toDate are to be converted to respective unix timestamps.
- Query parameters are tenantId(can be single or multiple), fromDate and toDate which are to be converted to respective unix timestamps.
- Basically Unix timestamp is a method to track time as running total of seconds. The count starts from Jan 1st 1970, which is considered as Unix Epoch. Unix timestamp is the number of seconds elapsed between a given date and unix epoch.

The part of pushing records to elastic search pgr indexes is done through Kafka-connect. The commands to create connector for indexing are as follows:-

Public Grievance Redressal Indexing (to index pgr-services):-

```
curl -X POST \
  http://kafka-connect.kafka-cluster:8083/connectors/ \
  H 'Content-Type: application/json' \
  H 'Cookie: SESSIONID=f1349448-761e-4ebc-a8bb-f6799e756185' \
  H 'Postman-Token: adabf0e8-0599-4ac9-a591-920586ff4d50' \
  H 'cache-control: no-cache' \
  d '{
  "name": "cms-case-es-sink9121",
  "config": {
    "connector.class": "io.confluent.connect.elasticsearch.ElasticsearchSinkConnector",
    "connection.url": "http://elasticsearch-data-v1.es-cluster:9200",
    "type.name": "general",
    "topics": "pgr-services",
    "key.ignore": "false",
    "schema.ignore": true,
    "value.converter.schemas.enable": false,
    "key.converter": "org.apache.kafka.connect.storage.StringConverter",
    "value.converter": "org.apache.kafka.connect.json.JsonConverter",
    "transforms": "TopicNameRouter",
    "transforms.TopicNameRouter.type":
"org.apache.kafka.connect.transforms.RegexRouter",
```

```
    "transforms.TopicNameRouter.regex": ".*",
    "transforms.TopicNameRouter.replacement": "pgr-services",
    "batch.size": 10,
    "max.buffered.records": 500,
    "flush.timeout.ms": 600000,
    "retry.backoff.ms": 5000,
    "read.timout.ms": 10000,
    "linger.ms": 100,
    "max.in.flight.requests": 2,
    "errors.log.enable": true,
    "errors.deadletterqueue.topic.name": "pgr-services-es-failed",
    "tasks.max": 1
  }
}'
```