**Local Setup**

**1. Preparing The Server**

There is problem in creating tables and inserting data to local DynamoDB using AWS CLI and thus this will be problem to creating and populating index in ElasticSearch because LogStash depends on the input from the DynamoDB tables.

To overcome this problem, the AWS CLI and Logstash need to be excluded from docker-compose.yml so the YAML configuration will contain the servers only.

The above tasks (populating DynamoDB tables and ElasticSearch index) will be done in manually.

A new YAML config file, named as local.yml is created so the original docker-compose.yml left intact.

File: **local.yml**

**version:** '2'  
**services:**  
 **informix:**  
 **image:** "appiriodevops/informix:1.2"  
 **ports:**  
 - "2021:2021"  
 **dynamodb:**   
 **image:** "tray/dynamodb-local"  
 **ports:**   
 - "7777:7777"  
 **command:** "-inMemory -port 7777"  
 **kafka:**  
 **image:** spotify/kafka  
 **environment:**  
 **ADVERTISED\_HOST:** $DOCKER\_IP  
 **ADVERTISED\_PORT:** 9092  
 **ports:**  
 - "2181:2181"  
 - "9092:9092"  
  **elasticsearch:**  
 **image:** "elasticsearch:2.3"  
 **ports:**  
 - "9200:9200"  
 - "9300:9300"  
 **redis:**  
 **image:** "redis:3.0.7"  
 **ports:**  
 - "6379:6379"

- Edit the env.sh found in ap-member-microservice/local to match your environment, mainly the Docker IP

- Open command line terminal and go to ap-member-microservice/local directory

**$** source env.sh

**$** docker-compose -f local.yml up

Now the required servers should be up and ready.

**2. Populating DynamoDB database**

- Grab the AWS CLI from <https://aws.amazon.com/cli/>

- Install AWS CLI and make sure the aws command in to system PATH

- Create .aws directory under your home directory

- Copy config and credential files in ap-member-microservice/local/aws-cli to .aws directory

- Still in ap-member-microservice/local/aws-cli, in command line terminal

**$** source ../env.sh

**$** ./init-dynamodb.sh

- Check if tables is created

**$** aws dynamodb list-tables --endpoint-url http://$IP:7777

- If there is an error, try to remove $PWD part from the inserting JSON line <file://$PWD/><file>.json to [file://<file>.json](file://member-profile.json/) as the JSON files are in same directory as init-dynamodb.sh

**3. Populating ElasticSearch index with LogStash**

- Get the Logstash v2.3.1 from <https://www.elastic.co/downloads/past-releases>

- Extract and configure so logstash command can be found in system PATH

- Create the ES index. Open command line terminal, in ap-member-microservice/local/logstash

**$** source ../env.sh

**$** ./mapping.sh

- Update hostname in ap-member-microservice/local/conf/regions.xml to match the Docker IP.

- Copy logstash.lib.sh from ap-member-microservice/local/logstash to bin directory of extracted logstash.

- Make sure this following line in logstash.lib.sh refers to correct path of regions.xml

JAVA\_OPTS="$JAVA\_OPTS -Dcom.amazonaws.regions.RegionUtils.fileOverride=regions.xml"

- For convenience, regions.xml is placed in the same directory as the logstash conf file.

- Install Jinja2 python module if it has not been installed before.

**$** pip install Jinja2

**$** pip install j2cli

- Open command line terminal, in ap-member-microservice/local/conf

**$** source ../env.sh

**$** j2 members-es-conf.j2 > members\_es.conf

- Make sure that generated members\_es.conf refer to correct path of informix JDBC jar. For convenience, this jar is already placed in same directory.

- Extract logstash-filter-skills.zip and install the plugin locally

**$** logstash-plugin install –-local <extracted\_path>/logstash-filter-skills-2.0.3.gem

- Install other necessary plugins

**$** logstash-plugin install logstash-input-dynamodb

**$** logstash-plugin install logstash-filter-dynamodb

**$** logstash-plugin install logstash-input-jdbc

**$** logstash-plugin install logstash-filter-jdbc

- Run the logstash using generated config file, in ap-member-microservice/local/conf

**$** logstash –-debug -f members\_es.conf

**4. Running the API service**

- Open command line terminal, in ap-member-microservice/service

**$** mvn clean compile package

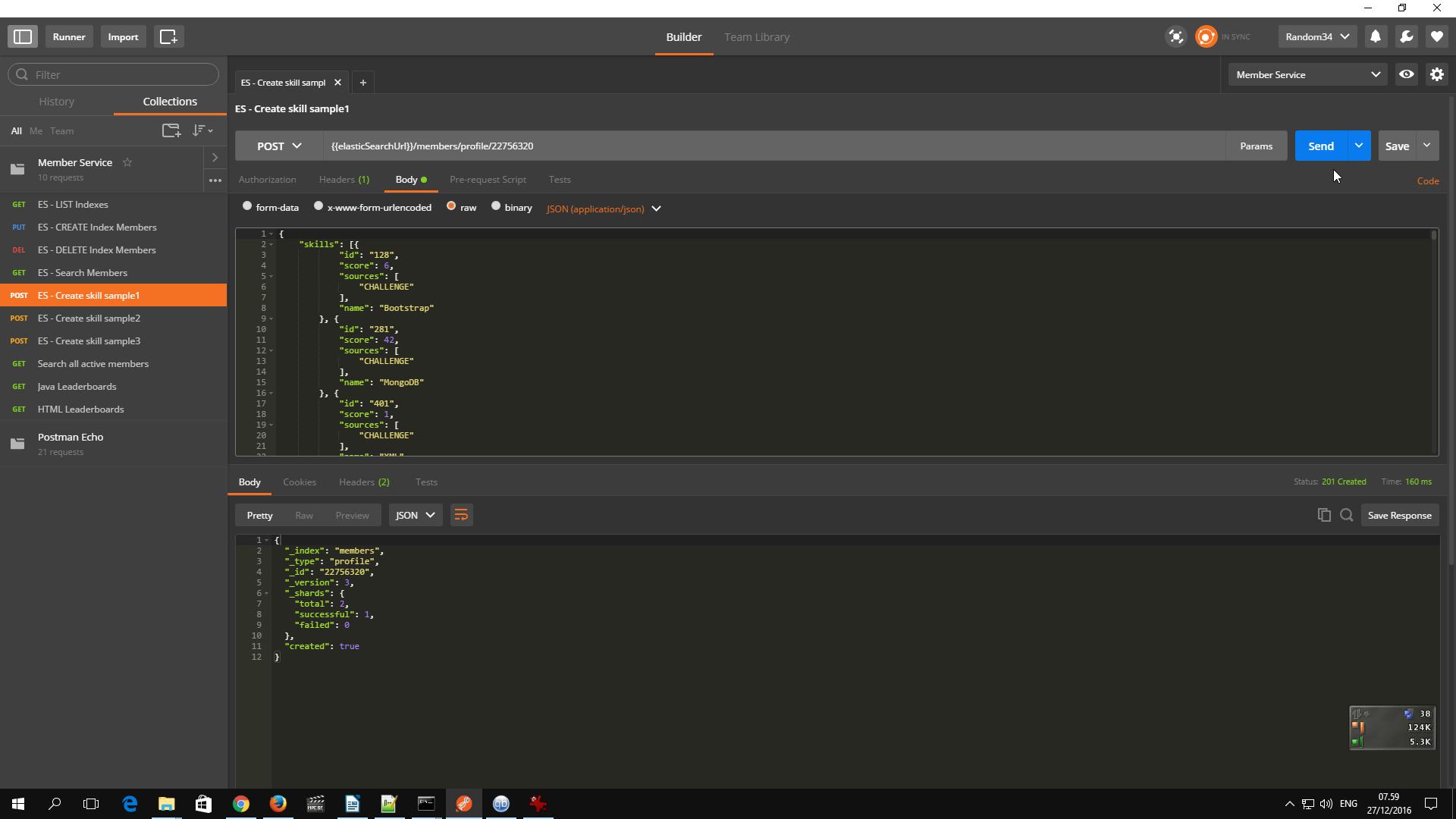
**$** java -jar target/member-microservice-\*.jar server src/main/resources/member-service.yaml

**5. Verifying Leaderboards Endpoint**

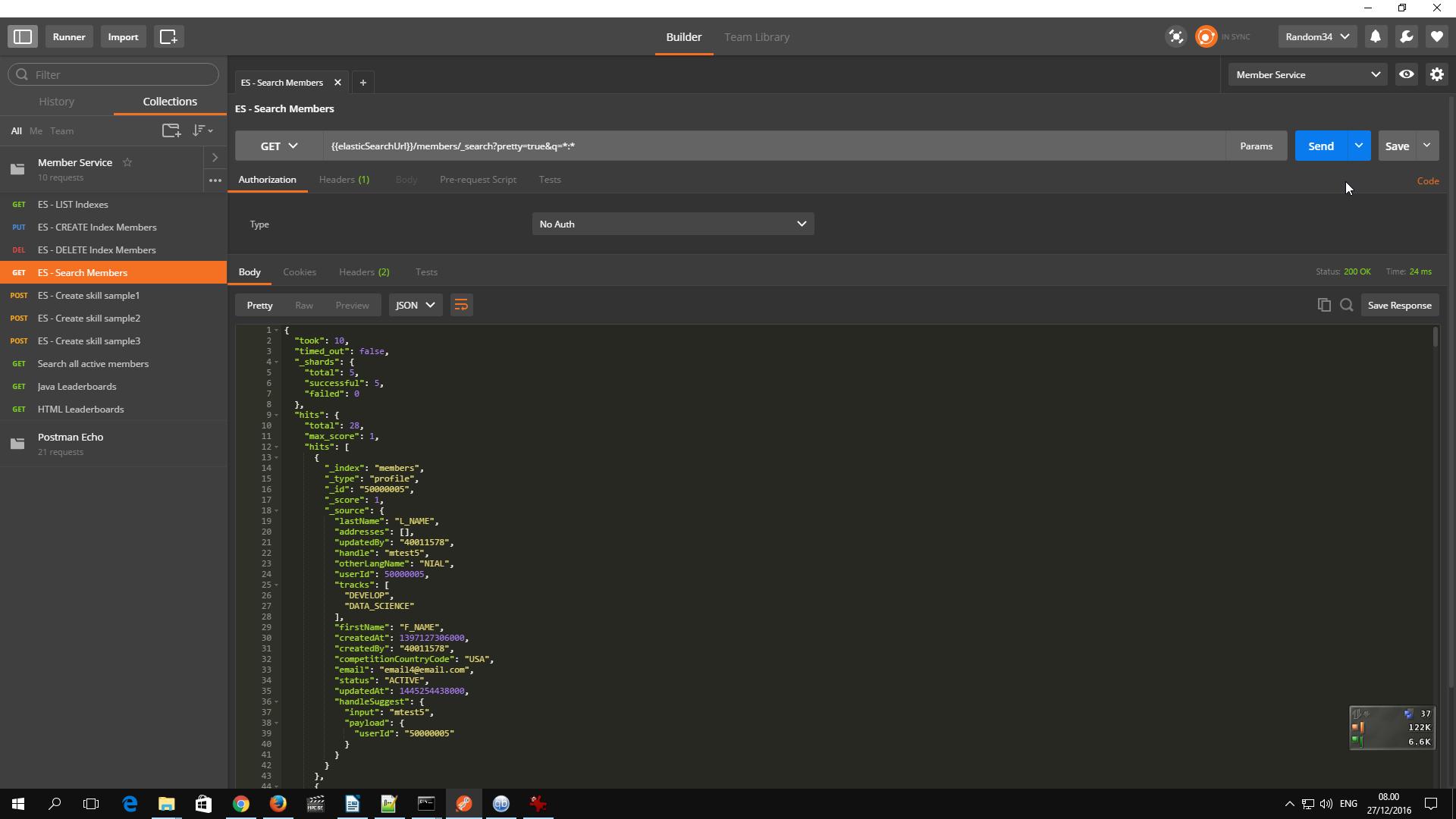
- Open Postman, import JSON collection and environment files from submission/postman directory.

- In “Member Service” Collection, there are some requests to insert necessary documents for verifying the new Leaderboards endpoint.

- Fires the “ES – Create Skill Sample <n>” to insert the sample document.



- Check the newly inserted document in ES



- Fires the leaderboards request

