Keycloak

http://localhost:8089

admin/admin

Elastic Search Login

http://localhost:9200

elastic/elastic

kibana

http://localhost:5601

elastic/elastic

mongodb

http://localhost:27017

root/root

Docker Setup

#################################

The project has docker images for ElasticSearch, Kibana and MongoDB

It is inside yummyfoodsapp/docker-compose folder.

In case at your side data is not available in MongoDB then there is a schema

yummyfoodsapp/dist/random-data.json which can be used to generate data from online genertors.

OR you can use sample.json to directly insert test records into the mongo.

Kibana & ElasticSearch

###############################

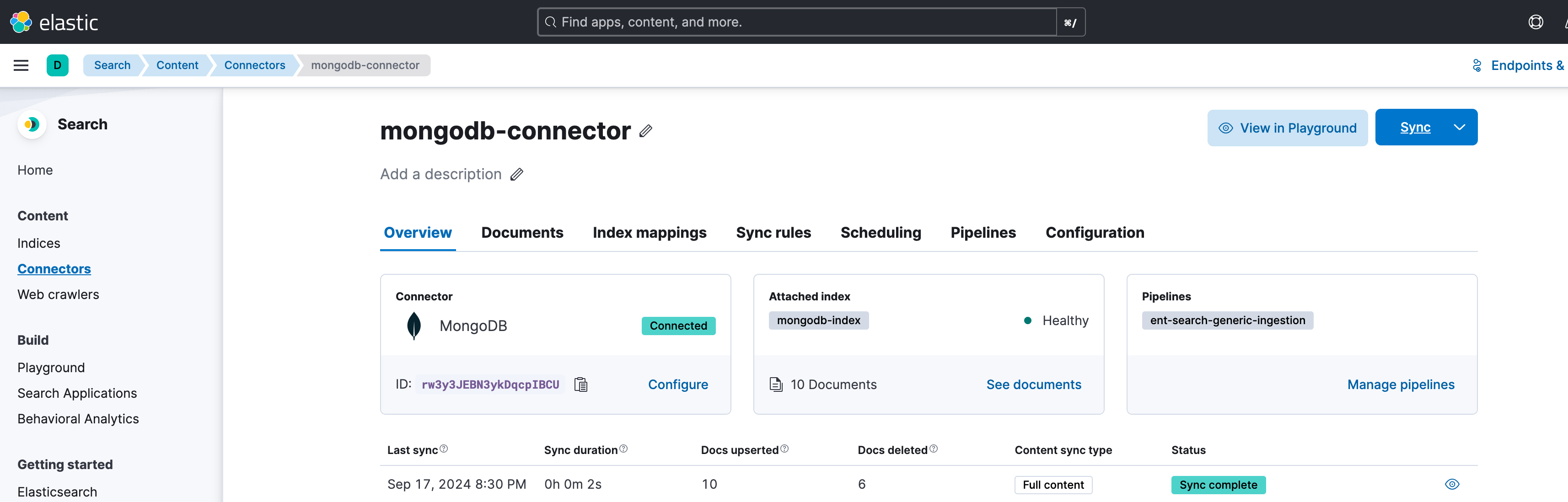
Once data is there in Mongo, there is a docker image for elastic connector running which picks up mongo records from restaurant collection periodically after 5 mins.

It will sync data from mongo to elastic in the background.

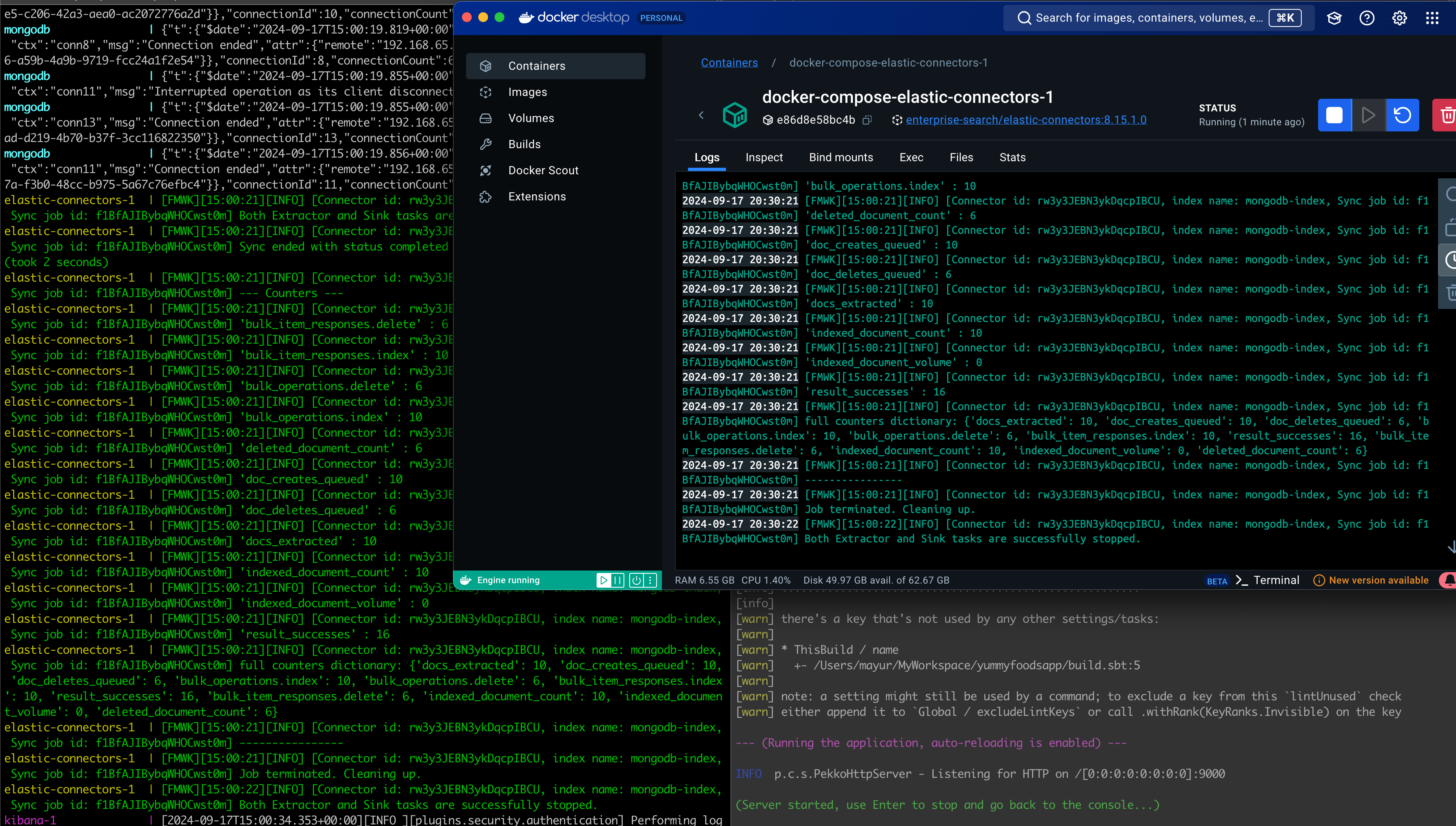
In case immediate sync is required, log in to the kibana

http://localhost:5601 using elastic/elastic credentials.

Got to Search > Content > Connectors > mongodb-connector > Sync



You can check the logs at container to verify.



On this page you will find a Documents tab, where you can verify the records are synced from Mongo. And elasticsearch queries can be fired on this console.

Postman

#################################################

Inside yummyfoodsapp/dist/postman-collections folder there are collections for elastic search and the application.

Import them in postman.

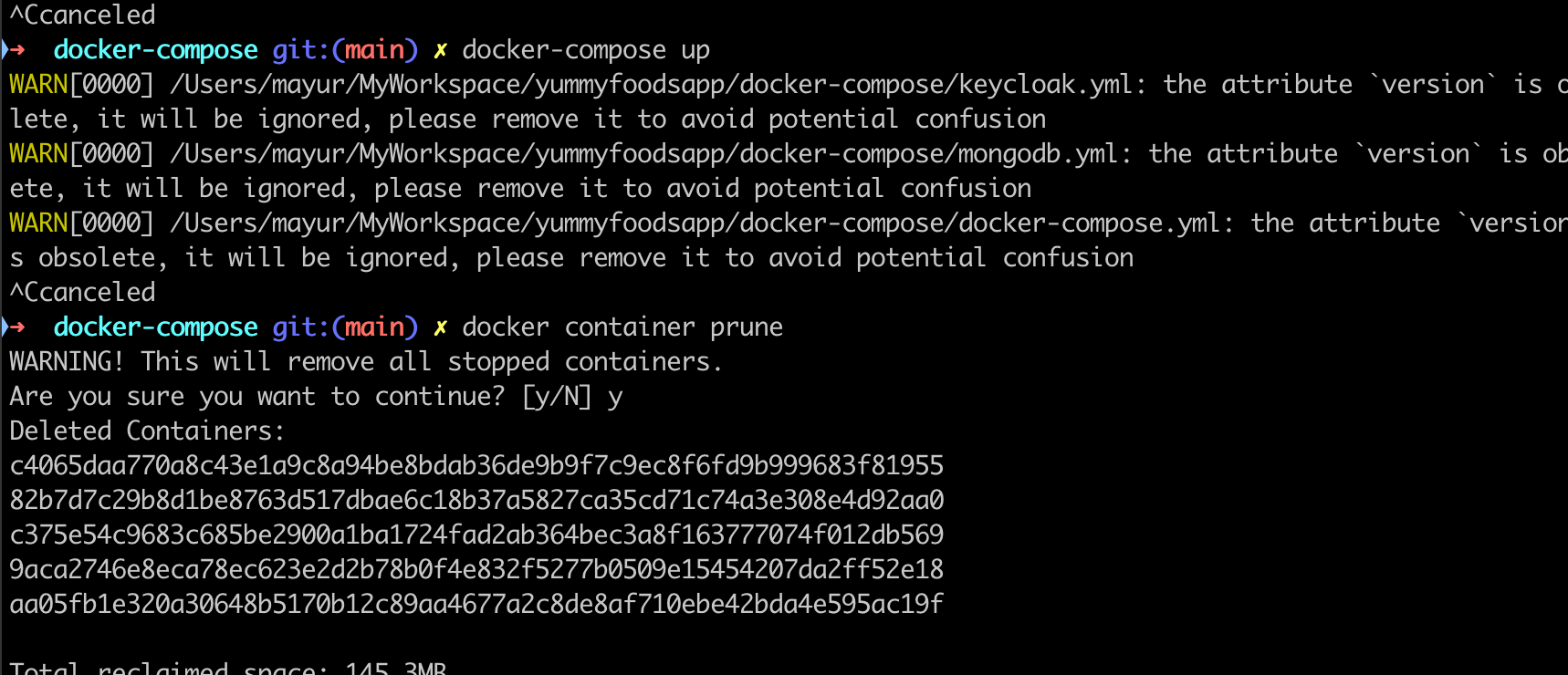
Runnning the app

######################################

Go to docker-compse folder and run

docker-compose up

Sometimes there is a Warning for stale containers, in that case run docker container prune and again fire docker-compose up.



Run the application by running sbt run

Wait some time till all images are up.

Connector image takes time.

There are prebuilt queries in the postman collections, which can be modified with search criteria in the input.

Currently keycloak is not setup, so please ignore it.

Some samples shown here to interact with app.

1. Crete restaurant

POST

http://localhost:9000/v1/restaurant

{

"restaurantName": "eiusmod in",

"contactNumber": "+91 (827) 588-2562",

"restaurantAddress": [

{

"addressId": "a334598d-5439-4989-a851-e1970e9e0f73",

"street": "irure duis nulla",

"addressLine1": "Lorem aliquip",

"addressLine2": "Lorem mollit",

"city": "Celeryville",

"state": "Louisiana",

"postalCode": 872544,

"countryCode": "US"

}

]

}

2. Get restaurant by id

http://localhost:9000/v1/restaurant/9ac70ee2-393a-415f-83b5-56ea07b96392

3. Get restaurant by keyword

http://localhost:9000/v1/restaurant?keywords=anim%20sint&offset=0&limit=5

{

"query": {

"match": {

"addressLine1": {

"query" : "sint"

}

}

}

}