Overview of solution skeleton

Following solution has been devised to address core requirements from IDI Take-Home Task

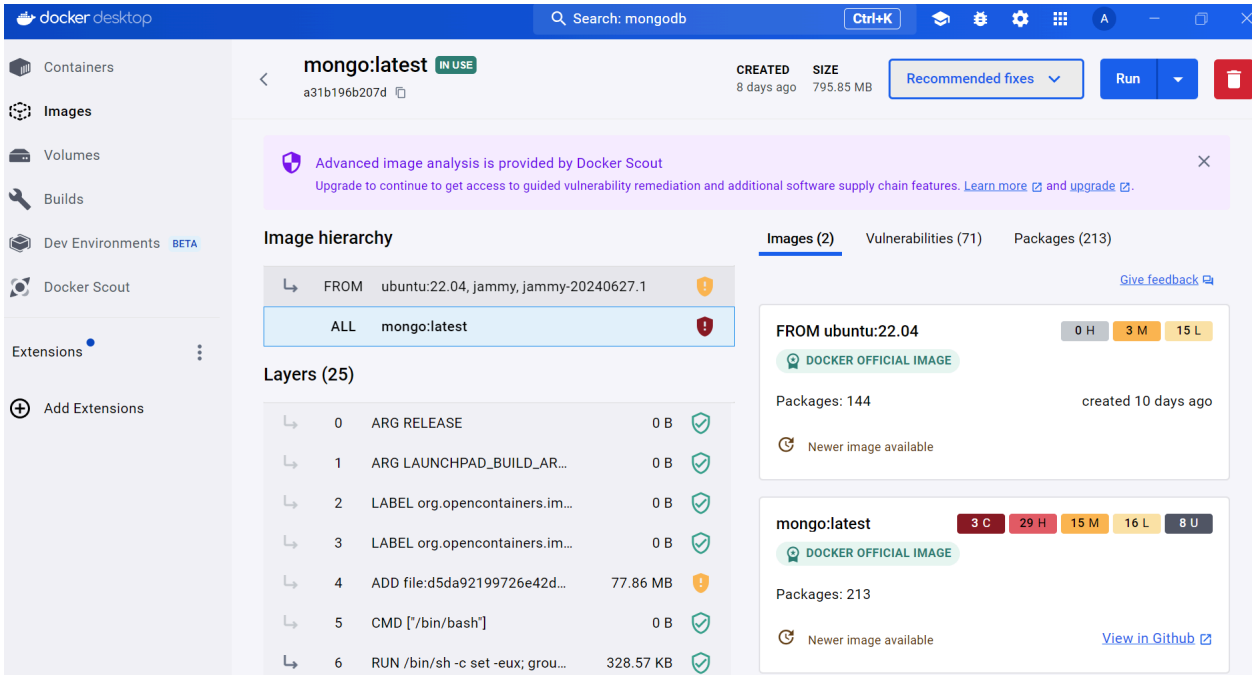
1. Loading and processing of TSV files is done using MongoDB to leverage TSV parsing and loading data into NoSQL db
2. MongoDB database is running on a standalone docker container to alleviate a need for installing mongodb on one's computer and leverage most stable software available to community
3. TSV file is mounted on the mongodb container for mongoimport utility
4. Default settings are used for mongoimport which means auto creation of 'test' database (hence the use of db.test prefix in queries below)
5. After loading data into a collection/table a series of mongodb queries are executed via mongosh prompt to address some of the points from **Part 2**

Caveats

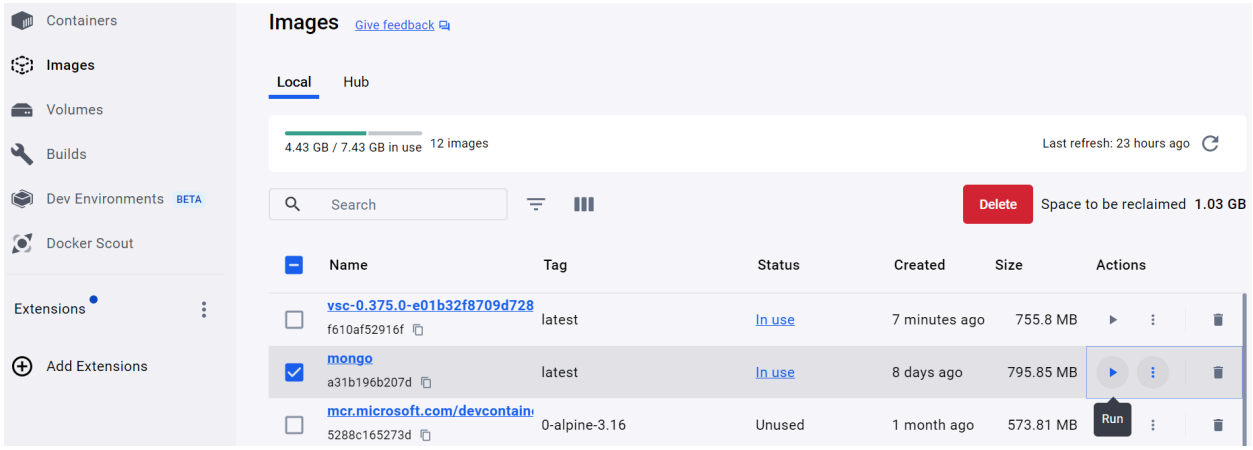
1. Small tsv file (Copy of correct_twitter 201904.tsv renamed to correct_twitter_201904.tsv) was used to perform queries.
2. TSV file was available locally (downloaded) for mounting on docker under /data/lz
3. TSV data containing timestamp columns was loaded by default as string type. Further query processing was required to reformat into date type - this was needed to perform date based aggregations. Mongodb aggregate along with $out operator was used as an equivalent.
4. Similarly Mongodb aggregate with $out operator was used to create a subset of data that met filtering criteria for a 'term'.
5. No further api was developed due to lack of familiarity with flask and time constraints

Instructions

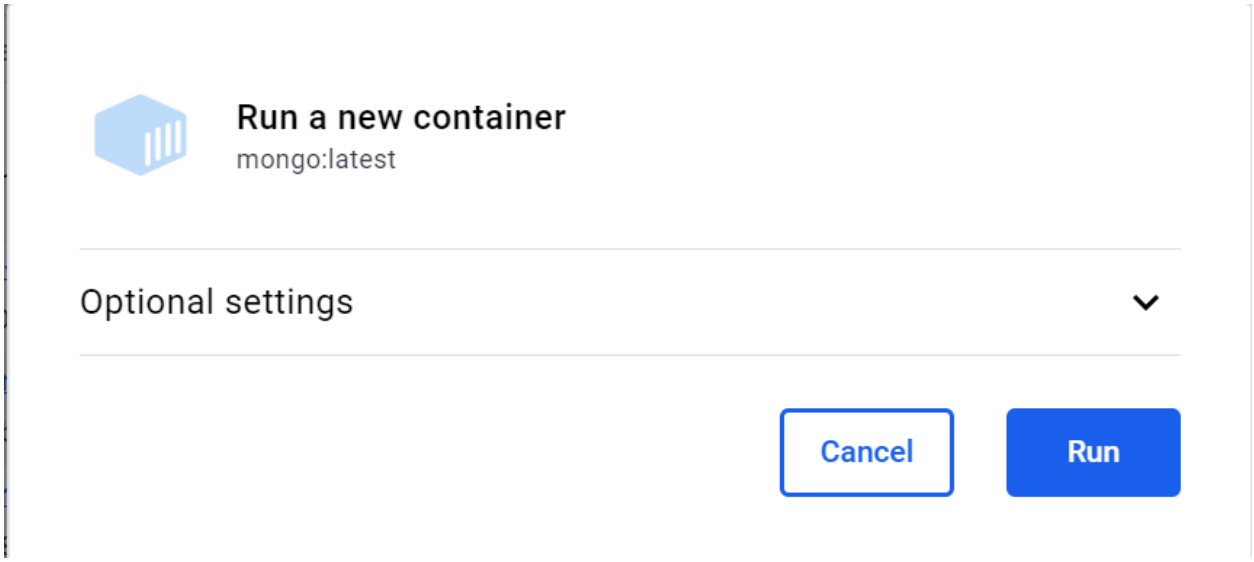
1. Install docker desktop if not available or use docker prompts if running on a console



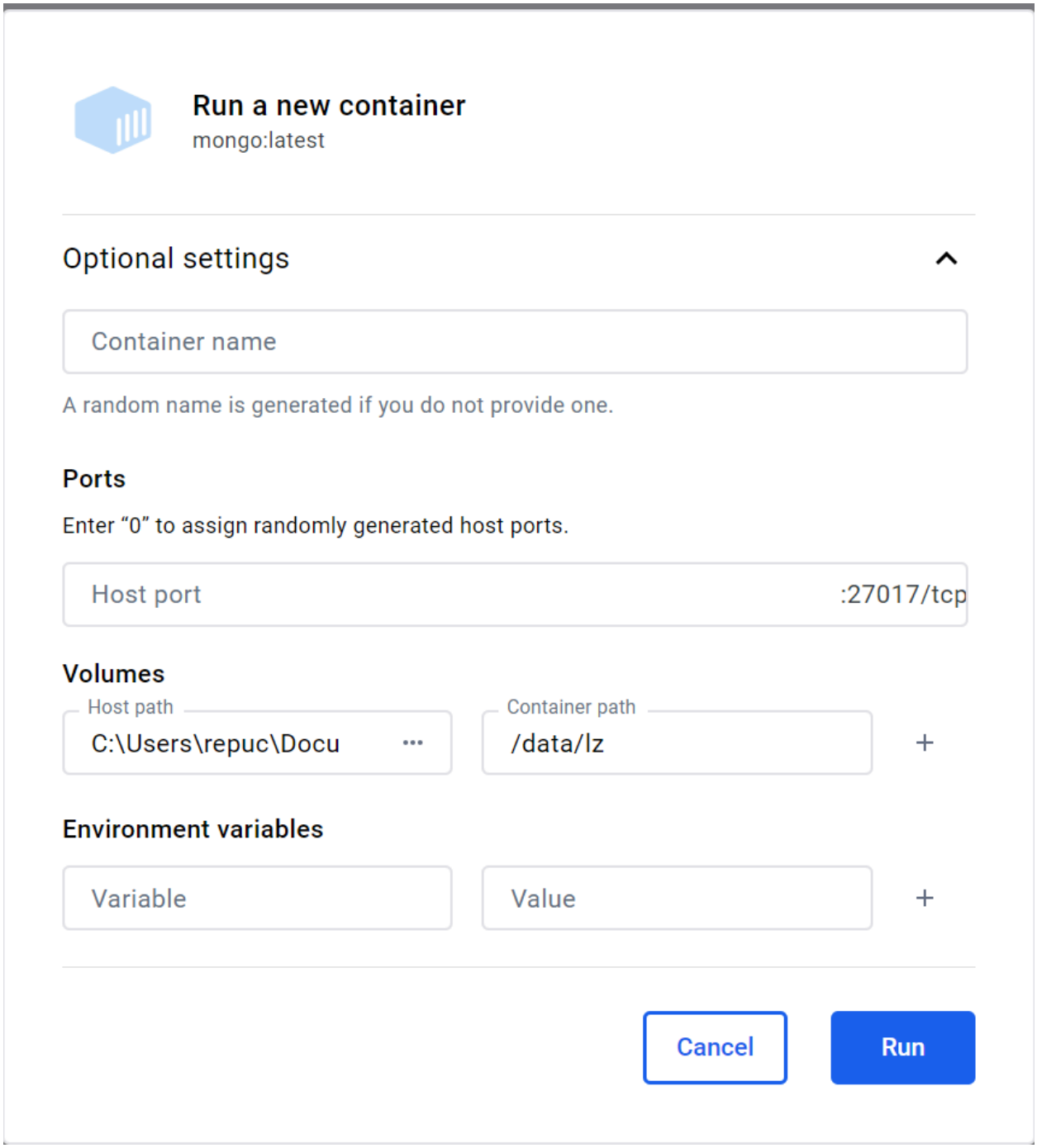
2. Start new container using mongodb:latest and mount tsv file directory under /data/lz
Click ‘Run’ under ‘Actions’ column for ‘mongo’ image in Docker Desktop ‘Images’ tab:



A popup window will be presented - open ‘Optional settings’ dropdown:



Fill in directory where tsv files are located and set it to mount to /data/lz on container and click ‘Run’:

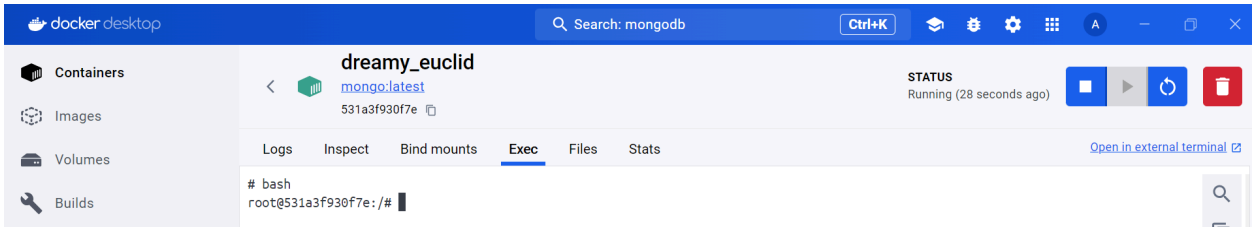


Or

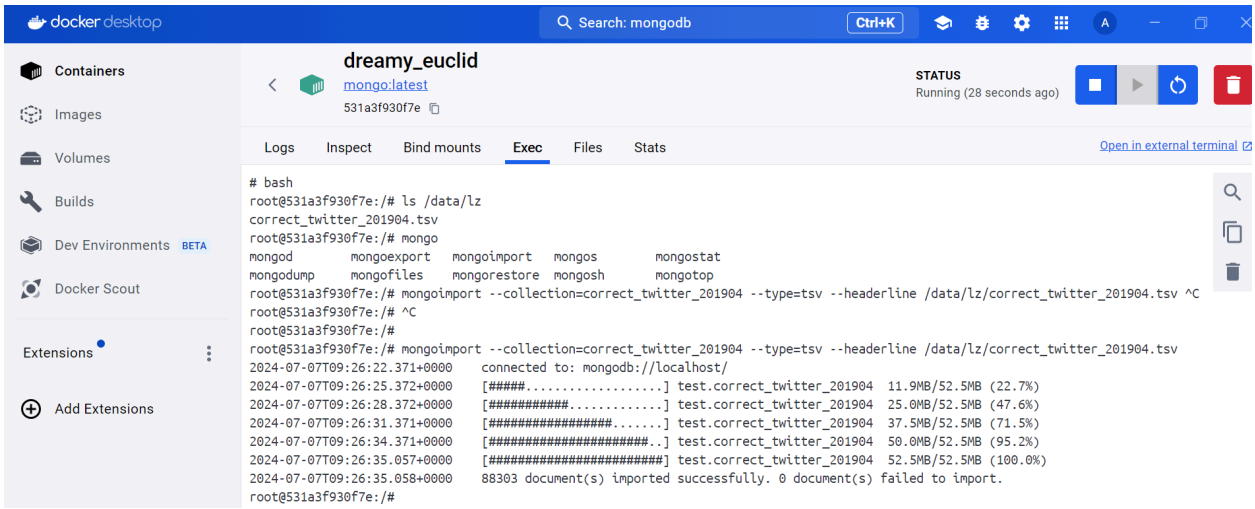
start from command line with mongodb:latest and volume mapping
docker run --hostname=531a3f930f7e
--env=PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
--env=GOSU_VERSION=1.17 --env=JSYAML_VERSION=3.13.1
--env=MONGO_PACKAGE=mongodb-org --env=MONGO_REPO=repo.mongodb.org

--env=MONGO_MAJOR=7.0 --env=MONGO_VERSION=7.0.12 --env=HOME=/data/db
--volume=C:\Users\repuc\Documents\lz:/data/lz --volume=/data/configdb
--volume=/data/db --restart=no --label='org.opencontainers.image.ref.name=ubuntu'
--label='org.opencontainers.image.version=22.04' --runtime=runc -d mongo:latest

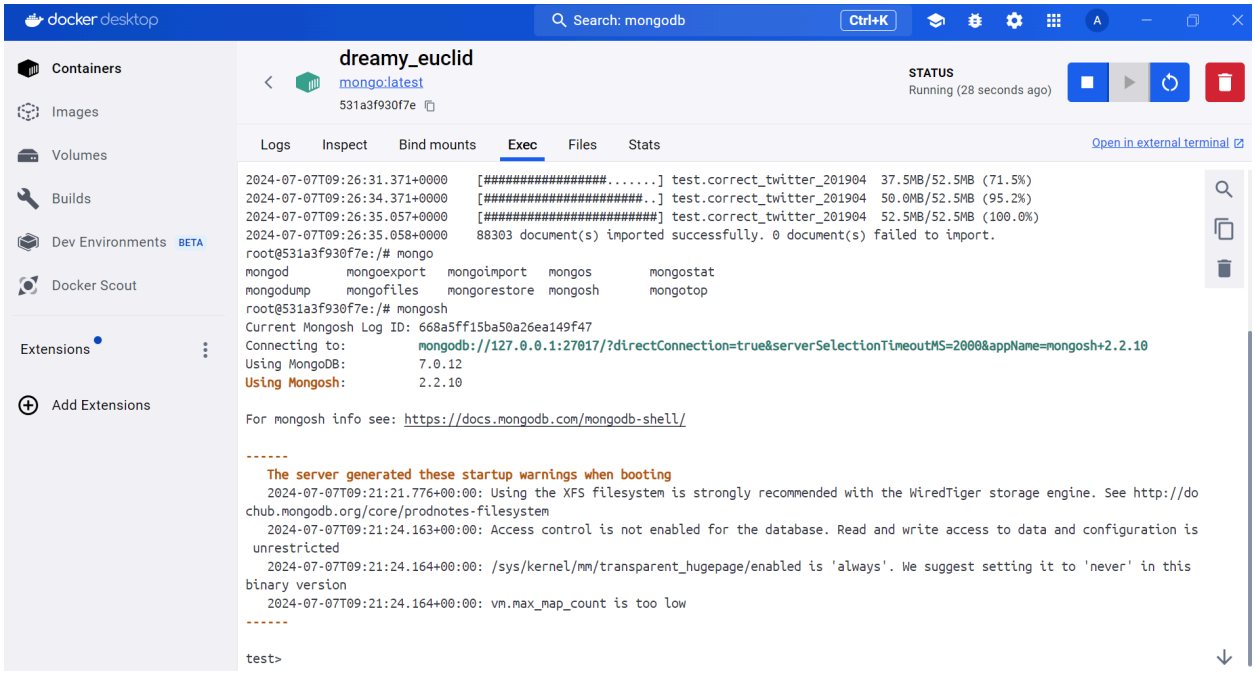
3. Pop into container console to start shell session, switch to bash shell:



root@531a3f930f7e:/# mongoimport --collection=correct_twitter_201904 --type=tsv
--headerline /data/lz/correct_twitter_201904.tsv



4. Kick off mongosh prompt to start interactive session:



5. Execute queries given below(no screenshots attached)

Queries

Date conversion from string to timestamp:

```
test>
db.correct_twitter_201904.aggregate([{"$addFields":{"ts1Converted":{"$convert":{"input":"$ts1","
to":"date"}}}}, {$out:"dt_conv_correct_twitter_201904"}])
```

```
test> show tables
correct_twitter_201904
dt_conv_correct_twitter_201904
```

Query to count tweets per day

```
test> db.ts_created_twitter_201904.aggregate( [ { $group: { _id: { $dateTrunc: {date:
"$ts_created_at", unit: "day" }}, countTweets: { $sum:  1 } } }  ] )
[
  { _id: ISODate('2019-03-21T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-16T00:00:00.000Z'), countTweets: 6405 },
  { _id: ISODate('2019-04-30T00:00:00.000Z'), countTweets: 2007 },
  { _id: ISODate('2019-05-02T00:00:00.000Z'), countTweets: 1647 },
  { _id: ISODate('2019-04-20T00:00:00.000Z'), countTweets: 8 },
  { _id: ISODate('2019-04-08T00:00:00.000Z'), countTweets: 5 },
```

```
  { _id: ISODate('2019-04-06T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-04-14T00:00:00.000Z'), countTweets: 4 },
  { _id: ISODate('2019-05-05T00:00:00.000Z'), countTweets: 1992 },
  { _id: ISODate('2019-04-01T00:00:00.000Z'), countTweets: 3 },
  { _id: ISODate('2019-05-15T00:00:00.000Z'), countTweets: 6612 },
  { _id: ISODate('2019-05-03T00:00:00.000Z'), countTweets: 2326 },
  { _id: ISODate('2019-04-28T00:00:00.000Z'), countTweets: 768 },
  { _id: ISODate('2019-03-16T00:00:00.000Z'), countTweets: 2 },
  { _id: ISODate('2019-04-04T00:00:00.000Z'), countTweets: 10 },
  { _id: ISODate('2019-04-29T00:00:00.000Z'), countTweets: 1964 },
  { _id: ISODate('2019-05-18T00:00:00.000Z'), countTweets: 3512 },
  { _id: ISODate('2019-04-23T00:00:00.000Z'), countTweets: 5 },
  { _id: ISODate('2019-04-16T00:00:00.000Z'), countTweets: 5 },
  { _id: ISODate('2019-03-02T00:00:00.000Z'), countTweets: 2 }
]
```

Create a subset of data in a separate table/collection containing only matches for regular expression for the term (here ‘tired’ was used). Output saved under tired_twitter_201904:

```
test>  db.ts_created_twitter_201904.aggregate([ { $match: {text:{'$regex' : 'tired', '$options' : 'i'}} }, {$out:"tired_twitter_201904"} ])
```

Query 1:
Find given phrase and aggregate

Count tweets per day containing term ‘tired’

```
test> db.tired_twitter_201904.aggregate( [ { $group: { _id: { $dateTrunc: {date: "$ts_created_at", unit: "day" }}, countTweets: { $sum:  1 } } }  ] )
[
  { _id: ISODate('2019-05-09T00:00:00.000Z'), countTweets: 42 },
  { _id: ISODate('2019-05-04T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-03T00:00:00.000Z'), countTweets: 2 },
  { _id: ISODate('2019-05-12T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-04-30T00:00:00.000Z'), countTweets: 3 },
  { _id: ISODate('2019-05-28T00:00:00.000Z'), countTweets: 3 },
  { _id: ISODate('2019-05-29T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-08T00:00:00.000Z'), countTweets: 4 },
  { _id: ISODate('2019-05-17T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-13T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-26T00:00:00.000Z'), countTweets: 2 },
  { _id: ISODate('2019-05-24T00:00:00.000Z'), countTweets: 1 },
```

```
  { _id: ISODate('2019-05-07T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-16T00:00:00.000Z'), countTweets: 2 },
  { _id: ISODate('2019-05-23T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-10T00:00:00.000Z'), countTweets: 9 },
  { _id: ISODate('2019-05-30T00:00:00.000Z'), countTweets: 2 },
  { _id: ISODate('2019-04-26T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-01T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-04-29T00:00:00.000Z'), countTweets: 1 }
]
```

Count tweets per day containing term ‘tired’ sorted in date descending order:

```
test> db.tired_twitter_201904.aggregate([
  {
    $group: {
      _id: {
        $dateTrunc: {
          date: "$ts_created_at",
          unit: "day"
        }
      },
      countTweets: {
        $sum: 1
      }
    }
  },
  {
    $sort: {
      _id: -1
    }
  }
])

[
  { _id: ISODate('2019-05-30T00:00:00.000Z'), countTweets: 2 },
  { _id: ISODate('2019-05-29T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-28T00:00:00.000Z'), countTweets: 3 },
  { _id: ISODate('2019-05-26T00:00:00.000Z'), countTweets: 2 },
  { _id: ISODate('2019-05-24T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-23T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-22T00:00:00.000Z'), countTweets: 2 },
  { _id: ISODate('2019-05-20T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-19T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-17T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-16T00:00:00.000Z'), countTweets: 2 },
```

```
  { _id: ISODate('2019-05-15T00:00:00.000Z'), countTweets: 2 },
  { _id: ISODate('2019-05-13T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-12T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-11T00:00:00.000Z'), countTweets: 8 },
  { _id: ISODate('2019-05-10T00:00:00.000Z'), countTweets: 9 },
  { _id: ISODate('2019-05-09T00:00:00.000Z'), countTweets: 42 },
  { _id: ISODate('2019-05-08T00:00:00.000Z'), countTweets: 4 },
  { _id: ISODate('2019-05-07T00:00:00.000Z'), countTweets: 1 },
  { _id: ISODate('2019-05-05T00:00:00.000Z'), countTweets: 3 }
]
```

Query 2
Unique users (based on author_handle) in the set containing term ‘tired’

```
test> db.tired_twitter_201904.aggregate([
  {
    $group: {
      _id: {
        $dateTrunc: {
          date: "$ts_created_at",
          unit: "day"
        }
      },
      uniqueUsers: {
        $addToSet: "$author_handle"
      }
    }
  },
  {
    $set: {
      uniqueUsers: {
        $size: "$uniqueUsers"
      }
    }
  }
])

[
  { _id: ISODate('2019-05-07T00:00:00.000Z'), uniqueUsers: 1 },
  { _id: ISODate('2019-05-03T00:00:00.000Z'), uniqueUsers: 2 },
  { _id: ISODate('2019-05-29T00:00:00.000Z'), uniqueUsers: 1 },
  { _id: ISODate('2019-05-24T00:00:00.000Z'), uniqueUsers: 1 },
```

```
  { _id: ISODate('2019-05-08T00:00:00.000Z'), uniqueUsers: 4 },
  { _id: ISODate('2019-05-23T00:00:00.000Z'), uniqueUsers: 1 },
  { _id: ISODate('2019-05-30T00:00:00.000Z'), uniqueUsers: 2 },
  { _id: ISODate('2019-04-30T00:00:00.000Z'), uniqueUsers: 3 },
  { _id: ISODate('2019-05-11T00:00:00.000Z'), uniqueUsers: 8 },
  { _id: ISODate('2019-05-26T00:00:00.000Z'), uniqueUsers: 2 },
  { _id: ISODate('2019-05-19T00:00:00.000Z'), uniqueUsers: 1 },
  { _id: ISODate('2019-05-17T00:00:00.000Z'), uniqueUsers: 1 },
  { _id: ISODate('2019-05-13T00:00:00.000Z'), uniqueUsers: 1 },
  { _id: ISODate('2019-05-10T00:00:00.000Z'), uniqueUsers: 9 },
  { _id: ISODate('2019-05-28T00:00:00.000Z'), uniqueUsers: 2 },
  { _id: ISODate('2019-05-09T00:00:00.000Z'), uniqueUsers: 42 },
  { _id: ISODate('2019-05-05T00:00:00.000Z'), uniqueUsers: 3 },
  { _id: ISODate('2019-04-29T00:00:00.000Z'), uniqueUsers: 1 },
  { _id: ISODate('2019-05-22T00:00:00.000Z'), uniqueUsers: 2 },
  { _id: ISODate('2019-05-04T00:00:00.000Z'), uniqueUsers: 1 }
]
```

Unique users (based on author_handle) in the set containing term ‘tired’ sorted in date descending order:

```
test> db.tired_twitter_201904.aggregate([
  {
    $group: {
      _id: {
        $dateTrunc: {
          date: "$ts_created_at",
          unit: "day"
        }
      },
      uniqueUsers: {
        $addToSet: "$author_handle"
      }
    }
  },
  {
    $set: {
      uniqueUsers: {
        $size: "$uniqueUsers"
      }
    }
  },
  {
    $sort: {
```

```
      _id: -1
    }
  }
])
```

Query 4.
Where did the tweets come from in terms of place_id?

```
test> db.ts_created_twitter_201904.aggregate( [ { $group: { _id: "$place_id", countTweets: {
$sum:  1 }}}  , {$sort: {countTweets: -1}}  ] )
[
  { _id: 'None', countTweets: 87279 },
  { _id: '3b77caf94bfc81fe', countTweets: 45 },
  { _id: '1ef1183ed7056dc1', countTweets: 17 },
  { _id: '01a9a39529b27f36', countTweets: 13 },
  { _id: '1c69a67ad480e1b1', countTweets: 11 },
  { _id: '1d9a5370a355ab0c', countTweets: 11 },
  { _id: '06b9691f34c91d1c', countTweets: 11 },
  { _id: '68e019afec7d0ba5', countTweets: 11 },
  { _id: '011add077f4d2da3', countTweets: 10 },
  { _id: '0570f015c264cbd9', countTweets: 9 },
  { _id: 'e4a0d228eb6be76b', countTweets: 9 },
  { _id: '00ab941b685334e3', countTweets: 9 },
  { _id: 'de599025180e2ee7', countTweets: 9 },
  { _id: '300bcc6e23a88361', countTweets: 8 },
  { _id: '8e9665cec9370f0f', countTweets: 8 },
  { _id: 'a612c69b44b2e5da', countTweets: 8 },
  { _id: 'c3f37afa9efcf94b', countTweets: 7 },
  { _id: 'ac88a4f17a51c7fc', countTweets: 7 },
  { _id: '4ec01c9dbc693497', countTweets: 7 },
  { _id: '5c62ffb0f0f3479d', countTweets: 6 }
]
```