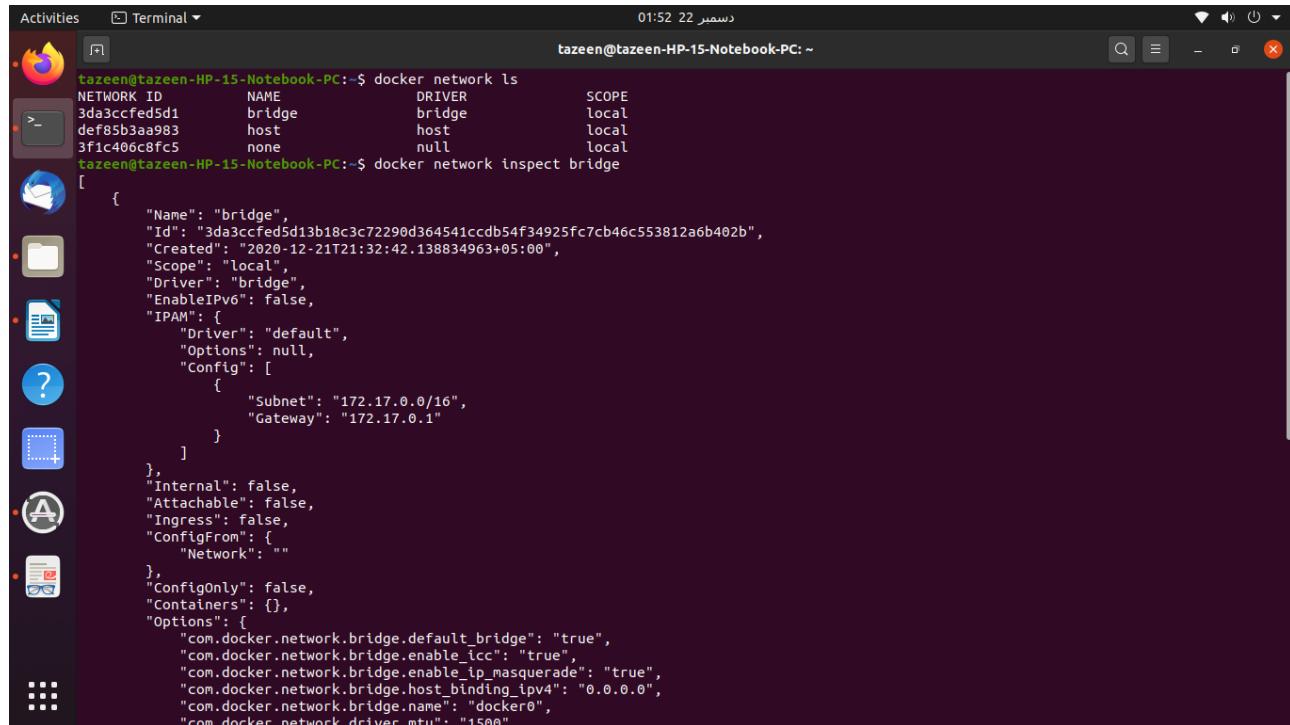


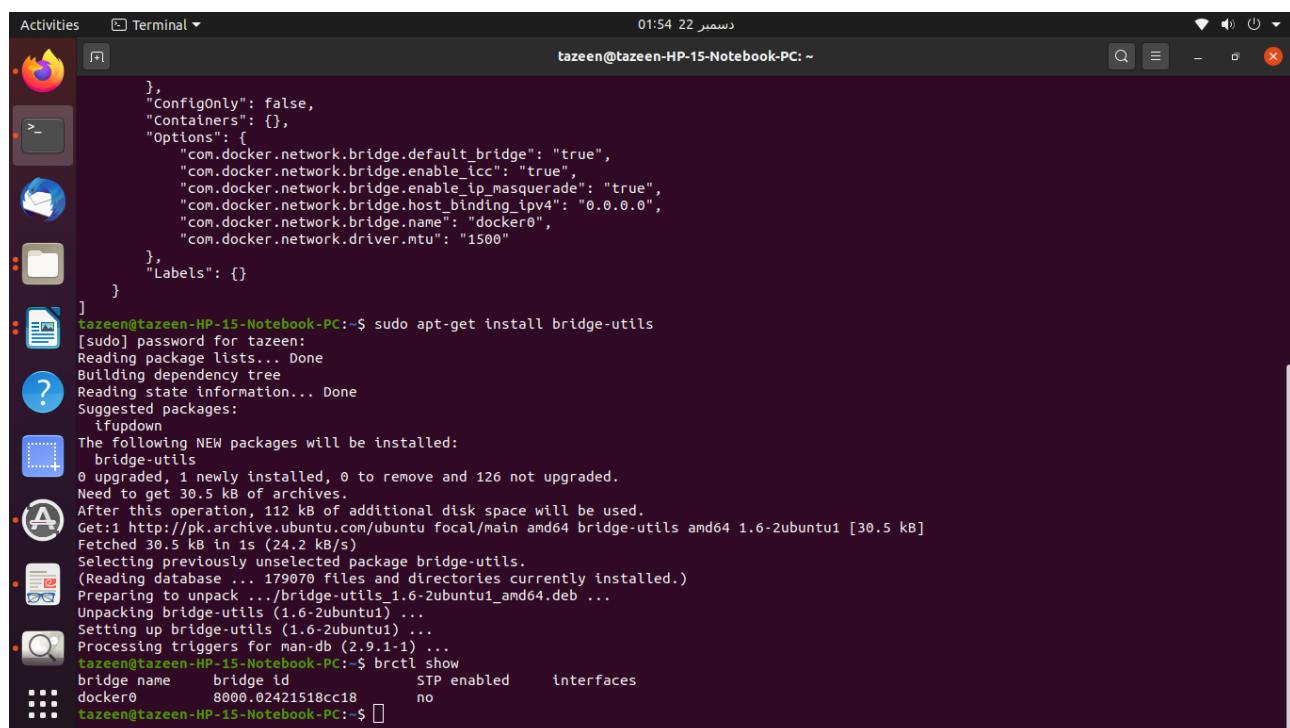
Lambda Architecture:

1- Database (mongodb layer)

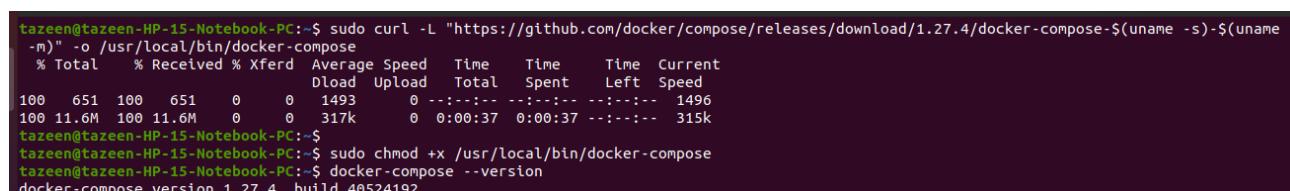
Creating docker-compose.yml file for mongo db container



```
tazeen@tazeen-HP-15-Notebook-PC:~$ docker network ls
NETWORK ID      NAME        DRIVER      SCOPE
3da3ccfed5d1   bridge      bridge      local
def85b3aa983   host        host       local
3f1c406c8fc5   none        null       local
tazeen@tazeen-HP-15-Notebook-PC:~$ docker network inspect bridge
[{"Name": "bridge", "Id": "3da3ccfed5d13b18c3c72290d364541ccdb54f34925fc7cb46c553812a6b402b", "Created": "2020-12-21T21:32:42.138834963+05:00", "Scope": "local", "Driver": "bridge", "EnableIPv6": false, "IPAM": {"Driver": "default", "Options": null, "Config": [{"Subnet": "172.17.0.0/16", "Gateway": "172.17.0.1"}]}, "Internal": false, "Attachable": false, "Ingress": false, "ConfigFrom": {"Network": ""}}, {"ConfigOnly": false, "Containers": {}, "Options": {"com.docker.network.bridge.default_bridge": "true", "com.docker.network.bridge.enable_icc": "true", "com.docker.network.bridge.enable_ip_masquerade": "true", "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0", "com.docker.network.bridge.name": "docker0", "com.docker.network.driver.mtu": "1500"}}, {"Labels": {}}], tazeen@tazeen-HP-15-Notebook-PC:~$
```



```
tazeen@tazeen-HP-15-Notebook-PC:~$ sudo apt-get install bridge-utils
[sudo] password for tazeen:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  ifupdown
The following NEW packages will be installed:
  bridge-utils
0 upgraded, 1 newly installed, 0 to remove and 126 not upgraded.
Need to get 30.5 kB of archives.
After this operation, 112 kB of additional disk space will be used.
Get:1 http://pk.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [30.5 kB]
Fetched 30.5 kB in 1s (24.2 kB/s)
Selecting previously unselected package bridge-utils.
(Reading database ... 179070 files and directories currently installed.)
Preparing to unpack .../bridge-utils_1.6-2ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.6-2ubuntu1) ...
Setting up bridge-utils (1.6-2ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
tazeen@tazeen-HP-15-Notebook-PC:~$ brctl show
bridge name            bridge id          STP enabled    interfaces
docker0                8000.02421518cc18  no           
```



```
tazeen@tazeen-HP-15-Notebook-PC:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total % Received % Xferd Average Speed Time Time Current
% Total % Received % Xferd Dload Upload Total Spent Left Speed
100 651 100 651 0 0 1493 0 --:--:-- --:--:-- 1496
100 11.6M 100 11.6M 0 0 317k 0 0:00:37 0:00:37 --:--:-- 315k
tazeen@tazeen-HP-15-Notebook-PC:~$ sudo chmod +x /usr/local/bin/docker-compose
tazeen@tazeen-HP-15-Notebook-PC:~$ docker-compose --version
docker-compose version 1.27.4, build 40524192
```

```
tazeen@tazeen-HP-15-Notebook-PC:~$ sudo nano docker-compose.yml
[sudo] password for tazeen: [REDACTED]
```

Activities Terminal 21:11 23 نسمر

```
GNU nano 4.8
Version: "3.2"
services:
  py-mongo:
    build:
      context: ${PWD}
    volumes:
      - $PWD/mongo-data:/data/db
      - $PWD/mongo-app:/var/www/html
    ports:
      - "27017:27017"
    environment:
      - MONGO_INITDB_ROOT_USERNAME=root
      - MONGO_INITDB_ROOT_PASSWORD=1234
```

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket
Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was

```
tazeen@tazeen-HP-15-Notebook-PC:~$ sudo nano Dockerfile
tazeen@tazeen-HP-15-Notebook-PC:~$ [REDACTED]
```

Activities Terminal 21:12 23 نسمر

```
GNU nano 4.8
FROM mongo:latest
# install Python 3
RUN apt-get update && apt-get install -y python3 python3-pip
RUN apt-get -y install python3.7-dev
RUN pip3 install pymongo
EXPOSE 27017
```

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket
Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was

سمبر 22:01 23 tazeen@tazeen-HP-15-Notebook-PC: ~

```
tazeen@tazeen-HP-15-Notebook-PC:~$ docker-compose up --build
Building py-mongo
Step 1/5 : FROM mongo:latest
latest: Pulling from library/mongo
f22ccc0b772: Pull complete
3cf8fb62ba5f: Pull complete
e80c964ece6a: Pull complete
329e632c35b3: Pull complete
3e1bd1325a3d: Pull complete
4aa0e3d64a4a: Pull complete
035bc87b778: Pull complete
874e4e43cb00: Pull complete
08cb97662b8b: Pull complete
f623cezb1e1: Pull complete
f100ac278196: Pull complete
gf5539f9b3ee: Pull complete
Digest: sha256:02e9941ddcb949424fa4eb01f9d235da91a5b7b64feb5887eab77e1ef84a3bad
Status: Downloaded newer image for mongo:latest
--> 3068f6bb852e
Step 2/5 : RUN apt-get update && apt-get install -y python3 python3-pip
--> Running in 9123a9b802de
Get:1 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Ign:3 http://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.4 InRelease
Get:4 http://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.4 Release [5391 B]
Get:5 http://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.4 Release.gpg [801 B]
Get:6 http://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.4/multiverse amd64 Packages [7139 B]
Get:7 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1372 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:13 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [15.3 kB]
Get:14 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [1816 kB]
Get:15 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [237 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [2136 kB]
```

سمبر 22:03 23 tazeen@tazeen-HP-15-Notebook-PC: ~

```
,log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minim um=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress,compact_progress,""]}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:29.855+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1608742829:855525][1:0x7f3dbc9e5ac0]", "txn-recover": "[WT_VERB_RECOVERY_PROGRESS] Recovering log 1 through 2"}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:29.957+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1608742829:957450][1:0x7f3dbc9e5ac0]", "txn-recover": "[WT_VERB_RECOVERY_PROGRESS] Recovering log 2 through 2"}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:30.054+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1608742830:54971][1:0x7f3dbc9e5ac0]", "txn-recover": "[WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Main recovery loop: starting at 1/29952 to 2/256"}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:30.158+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1608742830:158106][1:0x7f3dbc9e5ac0]", "txn-recover": "[WT_VERB_RECOVERY_PROGRESS] Recovering log 1 through 2"}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:30.285+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1608742830:285128][1:0x7f3dbc9e5ac0]", "txn-recover": "[WT_VERB_RECOVERY_PROGRESS] Recovering log 2 through 2"}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:30.351+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1608742830:351984][1:0x7f3dbc9e5ac0]", "txn-recover": "[WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global recovery timestamp: (0, 0)"}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:30.352+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1608742830:352043][1:0x7f3dbc9e5ac0]", "txn-recover": "[WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global oldest timestamp: (0, 0)"}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:30.934+00:00"}, "s": "I", "c": "STORAGE", "id": 4795906, "ctx": "initandlisten", "msg": "WiredTiger opened", "attr": {"durationMillis": 1919}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:30.934+00:00"}, "s": "I", "c": "RECOVERY", "id": 23987, "ctx": "initandlisten", "msg": "WiredTiger recoveryTimestamp", "attr": {"recoveryTimestamp": {"$timestamp": {"t": 0, "i": 0}}}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:30.937+00:00"}, "s": "I", "c": "STORAGE", "id": 4366408, "ctx": "initandlisten", "msg": "No table log ging settings modifications are required for existing WiredTiger tables", "attr": {"loggingEnabled": true}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:30.938+00:00"}, "s": "I", "c": "STORAGE", "id": 22262, "ctx": "initandlisten", "msg": "Timestamp monitor starting"}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:31.027+00:00"}, "s": "I", "c": "STORAGE", "id": 20536, "ctx": "initandlisten", "msg": "Flow Control is enabled on this deployment"}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:31.030+00:00"}, "s": "I", "c": "FTDC", "id": 20625, "ctx": "initandlisten", "msg": "Initializing full-time diagnostic data capture", "attr": {"dataDirectory": "/data/db/diagnostic.data"}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:31.033+00:00"}, "s": "I", "c": "NETWORK", "id": 23015, "ctx": "listener", "msg": "Listening on", "at tr": {"address": "/tmp/mongodb-27017.sock"}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:31.033+00:00"}, "s": "I", "c": "NETWORK", "id": 23015, "ctx": "listener", "msg": "Listening on", "at tr": {"address": "0.0.0.0"}}
py-mongo_1 | {"t": {"$date": "2020-12-23T17:00:31.034+00:00"}, "s": "I", "c": "NETWORK", "id": 23016, "ctx": "listener", "msg": "Waiting for connections", "attr": {"port": 27017, "ssl": "off"}}
```

Get the docker container id and inspect it to see the ip address

Activities Terminal ٠١:٣٦ ٢٤ دسمبر tazeen@tazeen-HP-15-Notebook-PC: ~

```
tazeen@tazeen-HP-15-Notebook-PC:~$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
90c3a25dffb1      tazeen_py-mongo   "docker-entrypoint.s..."  3 hours ago       Up 3 hours          0.0.0.0:27017->27017/tcp   tazeen_py-mongo_1

tazeen@tazeen-HP-15-Notebook-PC:~$ docker inspect 90c3a25dffb1
[{"Id": "90c3a25dffb1312ee5a802ca40497fe576e1c68d3fd4fcf9006a212c40bd23ff", "Created": "2020-12-23T17:40:02.032629196Z", "Path": "docker-entrypoint.sh", "Args": ["mongod"], "State": {"Status": "running", "Running": true, "Paused": false, "Restarting": false, "OOMKilled": false, "Dead": false, "Pid": 13810, "ExitCode": 0, "Error": "", "StartedAt": "2020-12-23T17:40:05.440689578Z", "FinishedAt": "2001-01-01T00:00:00Z"}, "Image": "sha256:575d0cd1ce89a06cc857aa7acb3bcab8eb4a964975d60f820c72d84c0f25702d", "ResolvConfPath": "/var/lib/docker/containers/90c3a25dffb1312ee5a802ca40497fe576e1c68d3fd4fcf9006a212c40bd23ff/resolv.conf", "HostnamePath": "/var/lib/docker/containers/90c3a25dffb1312ee5a802ca40497fe576e1c68d3fd4fcf9006a212c40bd23ff/hostname", "HostsPath": "/var/lib/docker/containers/90c3a25dffb1312ee5a802ca40497fe576e1c68d3fd4fcf9006a212c40bd23ff/hosts", "LogPath": "/var/lib/docker/containers/90c3a25dffb1312ee5a802ca40497fe576e1c68d3fd4fcf9006a212c40bd23ff/90c3a25dffb1312ee5a802ca40497fe576e1c68d3fd4fcf9006a212c40bd23ff-json.log", "Name": "/tazeen_py-mongo_1", "RestartCount": 0, "Driver": "overlay2", "Platform": "linux", "MountLabel": "", "ProcessLabel": "", "AppArmorProfile": "docker-default"}
```

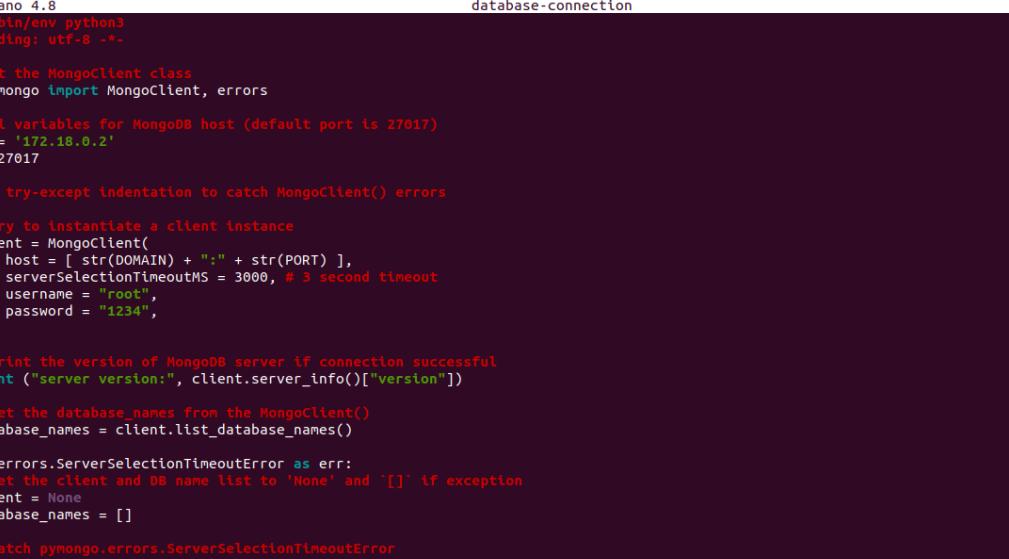
Activities Terminal ٠١:٣٧ ٢٤ دسمبر tazeen@tazeen-HP-15-Notebook-PC: ~

```
z7017@tazeen-HP-15-Notebook-PC:~$ curl -L
{
    "HostIp": "0.0.0.0",
    "HostPort": "27017"
}
},
"sandboxKey": "/var/run/docker/netns/c732369d4800",
"SecondaryIPv6Addresses": null,
"SecondaryIPv6PrefixLen": null,
"EndpointID": "",
"Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "",
"IPPrefixLen": 0,
"IPv6Gateway": "",
"MacAddress": "",
"Networks": {
    "tazeen_default": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": [
            "py-mongo",
            "90c3a25dffb1"
        ],
        "NetworkID": "93f176d001f83a0589b0751ddec10adcc77ec9b066f1eb865bcf68174cdb7fd8",
        "EndpointID": "b635c8a715a3e1dcea3f532e049518aca5b42a4db986f8d4b48655b2a6f779d9",
        "Gateway": "172.18.0.1",
        "IPAddress": "172.18.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:12:00:02",
        "DriverOpts": null
    }
}
}
```

```
Activities Terminal 01:44 24 سمسر tazeen@tazeen-HP-15-Notebook-PC: ~/mongo-app

tazeen@tazeen-HP-15-Notebook-PC:~$ ls -a
. .bashrc Desktop Downloads mongo-app Pictures .rediscli_history Templates
.. .cache docker-compose.yml get-docker.sh mongo-data .profile snap .thunderbird
.bash_history .config Dockerfile .gnupg .mozilla Public .ssh Videos
.bash_logout database-connection Documents .local Music .python_history .sudo_as_admin_successful

tazeen@tazeen-HP-15-Notebook-PC:~$ cd mongo-app
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$ sudo nano database-connection
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$
```



A screenshot of a Linux desktop environment (Ubuntu) showing a terminal window. The terminal title is "tazeen@tazeen-HP-15-Notebook-PC: ~/mongo-app". The code in the terminal is a Python script for connecting to a MongoDB database. It uses the `pymongo` library to handle the connection, specifying the host as '172.18.0.2' on port 27017, and attempting to log in as 'root' with password '1234'. It prints the server version if successful and lists all database names. It handles a timeout exception by setting the client and DB name list to 'None' and '[]'. It also catches a `pymongo.errors.ServerSelectionTimeoutError` and prints the error message.

```
GNU nano 4.8 database-connection
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

# import the MongoClient class
from pymongo import MongoClient, errors

# global variables for MongoDB host (default port is 27017)
DOMAIN = '172.18.0.2'
PORT = 27017

# use a try-except indentation to catch MongoClient() errors
try:
    # try to instantiate a client instance
    client = MongoClient(
        host = [ str(DOMAIN) + ":" + str(PORT) ],
        serverSelectionTimeoutMS = 3000, # 3 second timeout
        username = "root",
        password = "1234",
    )

    # print the version of MongoDB server if connection successful
    print ("server version:", client.server_info()["version"])

    # get the database_names from the MongoClient()
    database_names = client.list_database_names()

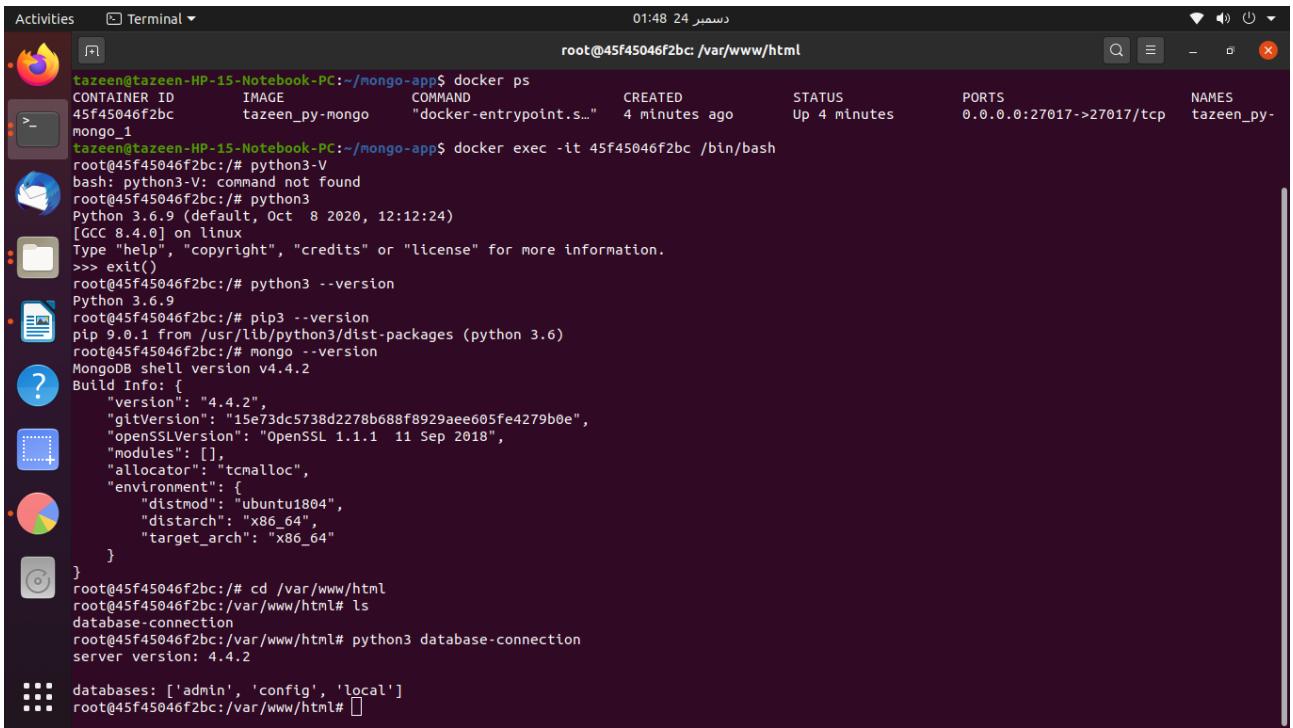
except errors.ServerSelectionTimeoutError as err:
    # set the client and DB name list to 'None' and '[]' if exception
    client = None
    database_names = []

    # catch pymongo.errors.ServerSelectionTimeoutError
    print ("pymongo ERROR:", err)

[ Read 35 lines ]

```

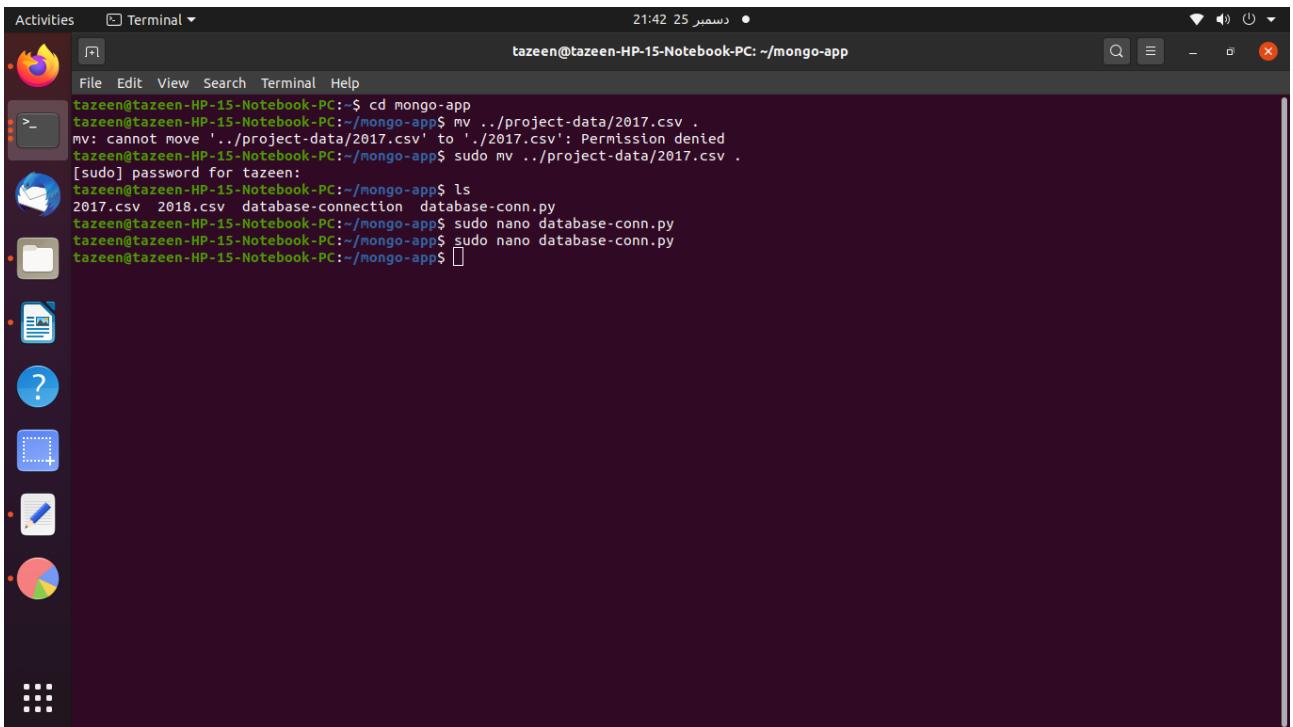
Execute the python script from docker to test the mongodb connection



Activities Terminal 01:48 24 دسمبر root@45f45046f2bc:/var/www/html

```
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
45f45046f2bc        tazeen_py-mongo   "docker-entrypoint.s..."  4 minutes ago     Up 4 minutes          0.0.0.0:27017->27017/tcp   tazeen_py-1
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$ docker exec -it 45f45046f2bc /bin/bash
root@45f45046f2bc:/# python3-V
bash: python3-V: command not found
root@45f45046f2bc:/# python3
Python 3.6.9 (default, Oct  8 2020, 12:12:24)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
root@45f45046f2bc:/# python3 --version
Python 3.6.9
root@45f45046f2bc:/# pip3 --version
pip 9.0.1 from /usr/lib/python3/dist-packages (python 3.6)
root@45f45046f2bc:/# mongo --version
MongoDB shell version v4.4.2
Build Info: {
    "version": "4.4.2",
    "gitVersion": "15e73dc5738d2278b688f8929aee605fe4279b0e",
    "openSSLVersion": "OpenSSL 1.1.1 11 Sep 2018",
    "modules": [],
    "allocator": "tcmalloc",
    "environment": {
        "distmod": "ubuntu1804",
        "distarch": "x86_64",
        "target_arch": "x86_64"
    }
}
root@45f45046f2bc:/# cd /var/www/html
root@45f45046f2bc:/var/www/html# ls
database-connection
root@45f45046f2bc:/var/www/html# python3 database-connection
server version: 4.4.2
databases: ['admin', 'config', 'local']
root@45f45046f2bc:/var/www/html#
```

Creating python script to save data in mongodb



Activities Terminal 21:42 25 دسمبر tazeen@tazeen-HP-15-Notebook-PC: ~/mongo-app

```
tazeen@tazeen-HP-15-Notebook-PC:~$ cd mongo-app
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$ mv ..project-data/2017.csv .
mv: cannot move '../project-data/2017.csv' to './2017.csv': Permission denied
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$ sudo mv ..project-data/2017.csv .
[sudo] password for tazeen:
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$ ls
2017.csv  2018.csv  database-connection  database-conn.py
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$ sudo nano database-conn.py
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$ sudo nano database-conn.py
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$
```

Saving data from 2018.csv and 2017.csv which is placed in mongo-app folder. These csv files are header based so starting with first index and inserting data into database.

Activities Terminal 23:37 29 سمبر tazeen@tazeen-HP-15-Notebook-PC: ~/mongo-app

```
GNU nano 4.8 database-conn.py
from pymongo import MongoClient, errors
import csv

# global variables for MongoDB host (default port is 27017)
DOMAIN = '172.18.0.2'
PORT = 27017

# use a try-except indentation to catch MongoClient() errors
try:
    # try to instantiate a client instance
    client = MongoClient(
        host = [ str(DOMAIN) + ":" + str(PORT) ],
        serverSelectionTimeoutMS = 3000, # 3 second timeout
        username = "root",
        password = "1234",
    )
    db=client.airline

    with open('2018.csv', 'r') as csvfile:
        header = ["FL_DATE","OP_CARRIER","OP_CARRIER_FL_NUM","ORIGIN","DEST","CRS_DEP_TIME","DEP_TIME","DEP_DELAY","TAXI_OUT","WHEELS_OFF","WHE">
        reader = csv.reader(csvfile)
        headerRow = next(reader)
        for row in reader:
            doc={}
            for n in range(0,len(header)):
                doc[header[n]] = row[n]
            db.flights.insert(doc)

    with open('2017.csv', 'r') as csvfile:
        header = ["FL_DATE","OP_CARRIER","OP_CARRIER_FL_NUM","ORIGIN","DEST","CRS_DEP_TIME","DEP_TIME","DEP_DELAY","TAXI_OUT","WHEELS_OFF","WHE">
        reader = csv.reader(csvfile)
        headerRow = next(reader)
        for row in reader:
            doc={}
            for n in range(0,len(header)):
                doc[header[n]] = row[n]
            db.flights.insert(doc)

# mongoimport --db=airline-db --collection=flights --type=csv --file=2018.csv --headerline --username=root --password=1234 --uri "mongodb://172.18.0.2:27017"
except errors.ServerSelectionTimeoutError as err:
    # catch pymongo.errors.ServerSelectionTimeoutError
    print ("pymongo ERROR:", err)
```

Read 42 lines]

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket
Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was

Activities Terminal 23:37 29 سمبر tazeen@tazeen-HP-15-Notebook-PC: ~/mongo-app

```
GNU nano 4.8 database-conn.py
# try to instantiate a client instance
client = MongoClient(
    host = [ str(DOMAIN) + ":" + str(PORT) ],
    serverSelectionTimeoutMS = 3000, # 3 second timeout
    username = "root",
    password = "1234",
)
db=client.airline

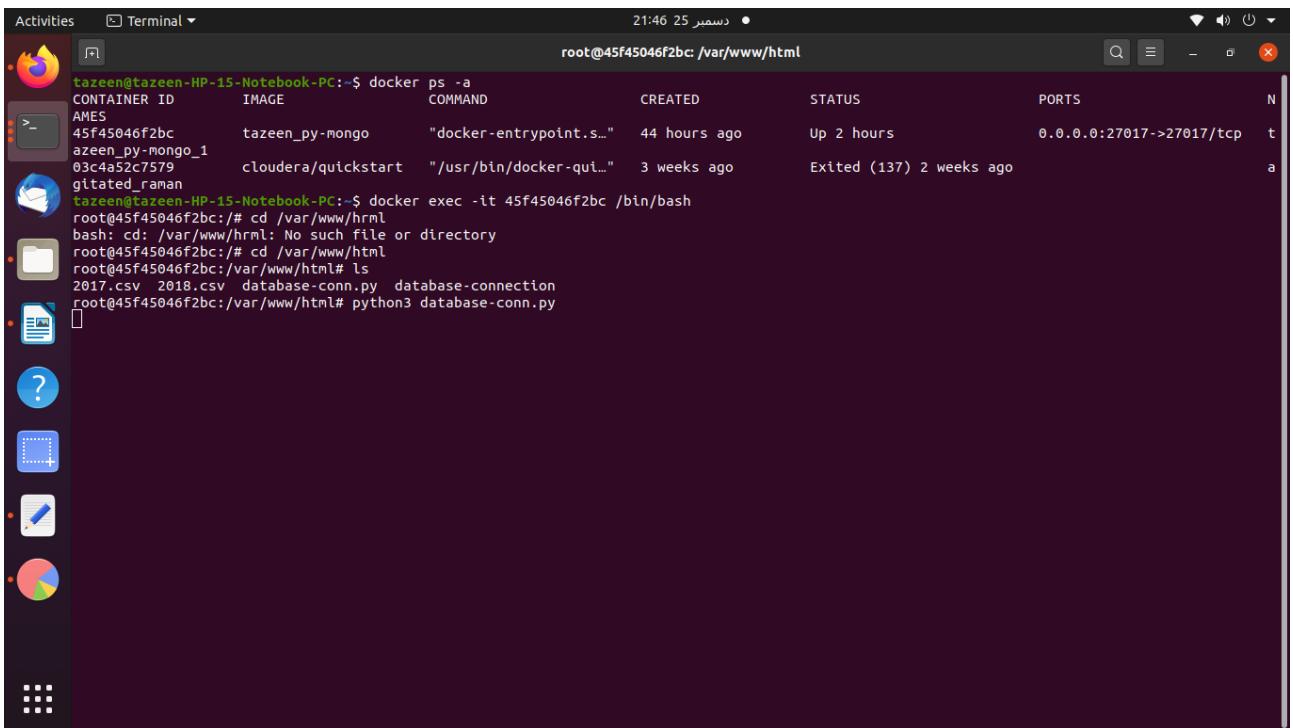
with open('2018.csv', 'r') as csvfile:
    header = ["FL_DATE","OP_CARRIER","OP_CARRIER_FL_NUM","ORIGIN","DEST","CRS_DEP_TIME","DEP_TIME","DEP_DELAY","TAXI_OUT","WHEELS_OFF","WHE">
    reader = csv.reader(csvfile)
    headerRow = next(reader)
    for row in reader:
        doc={}
        for n in range(0,len(header)):
            doc[header[n]] = row[n]
        db.flights.insert(doc)

with open('2017.csv', 'r') as csvfile:
    header = ["FL_DATE","OP_CARRIER","OP_CARRIER_FL_NUM","ORIGIN","DEST","CRS_DEP_TIME","DEP_TIME","DEP_DELAY","TAXI_OUT","WHEELS_OFF","WHE">
    reader = csv.reader(csvfile)
    headerRow = next(reader)
    for row in reader:
        doc={}
        for n in range(0,len(header)):
            doc[header[n]] = row[n]
        db.flights.insert(doc)

# mongoimport --db=airline-db --collection=flights --type=csv --file=2018.csv --headerline --username=root --password=1234 --uri "mongodb://172.18.0.2:27017"
except errors.ServerSelectionTimeoutError as err:
    # catch pymongo.errors.ServerSelectionTimeoutError
    print ("pymongo ERROR:", err)
```

Read 42 lines]

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket
Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was

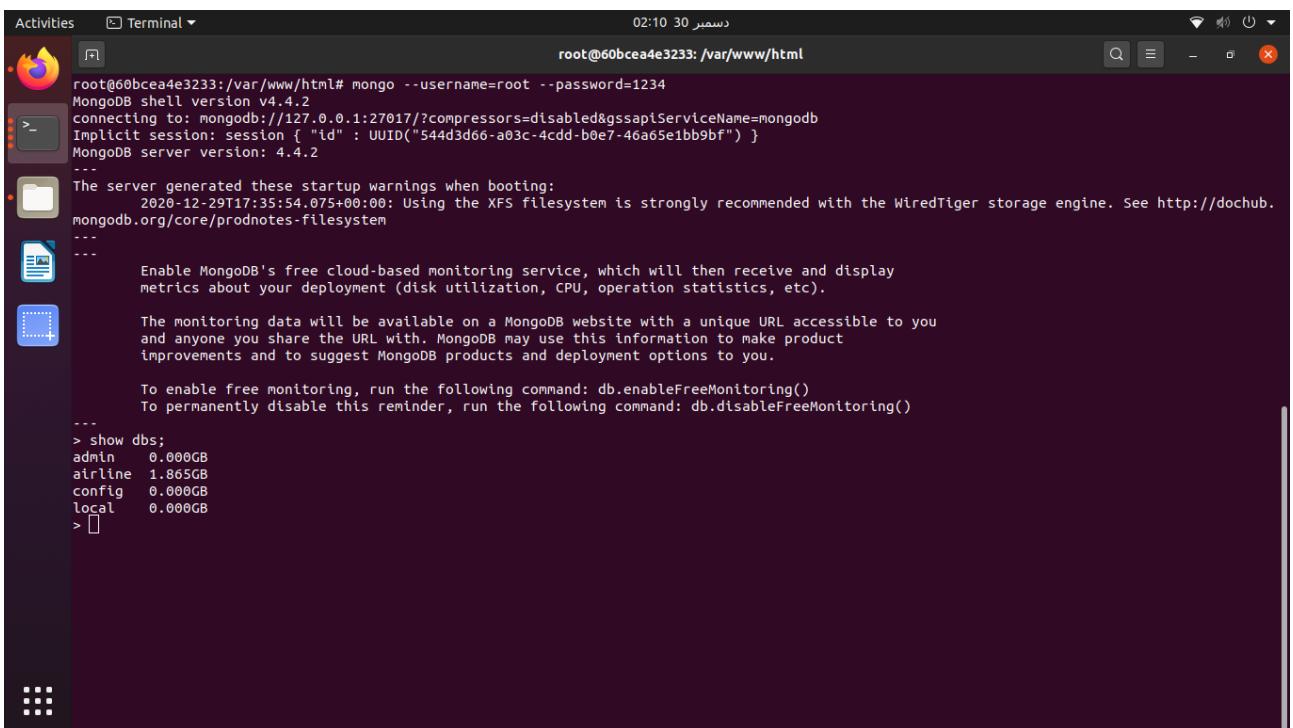


A screenshot of a Linux desktop environment (Ubuntu) showing a terminal window titled "Terminal". The terminal is running a command to list Docker containers. The output shows three containers:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
45f45046f2bc	tazeen_py-mongo	"docker-entrypoint.s..."	44 hours ago	Up 2 hours	0.0.0.0:27017->27017/tcp
03c4a52c7579	cloudera/quickstart	"/usr/bin/docker-qui..."	3 weeks ago	Exited (137) 2 weeks ago	

The user then runs "docker exec -it 45f45046f2bc /bin/bash" to enter the mongo container. Inside, they run "cd /var/www/html" and "ls" to see files 2017.csv and 2018.csv. They then run "python3 database-conn.py" to connect to the database.

Connecting to mongodb to check that data is inserted in airline database.



A screenshot of a Linux desktop environment (Ubuntu) showing a terminal window titled "Terminal". The terminal is running a mongo shell session. The user connects to the MongoDB instance at 127.0.0.1:27017. The session shows startup warnings about XFS filesystem usage and MongoDB monitoring. The user then runs "show dbs" which lists databases: admin, airline, config, and local. The "airline" database has a size of 1.865GB.

2- Batch processing layer:

Pulling image for hadoop container

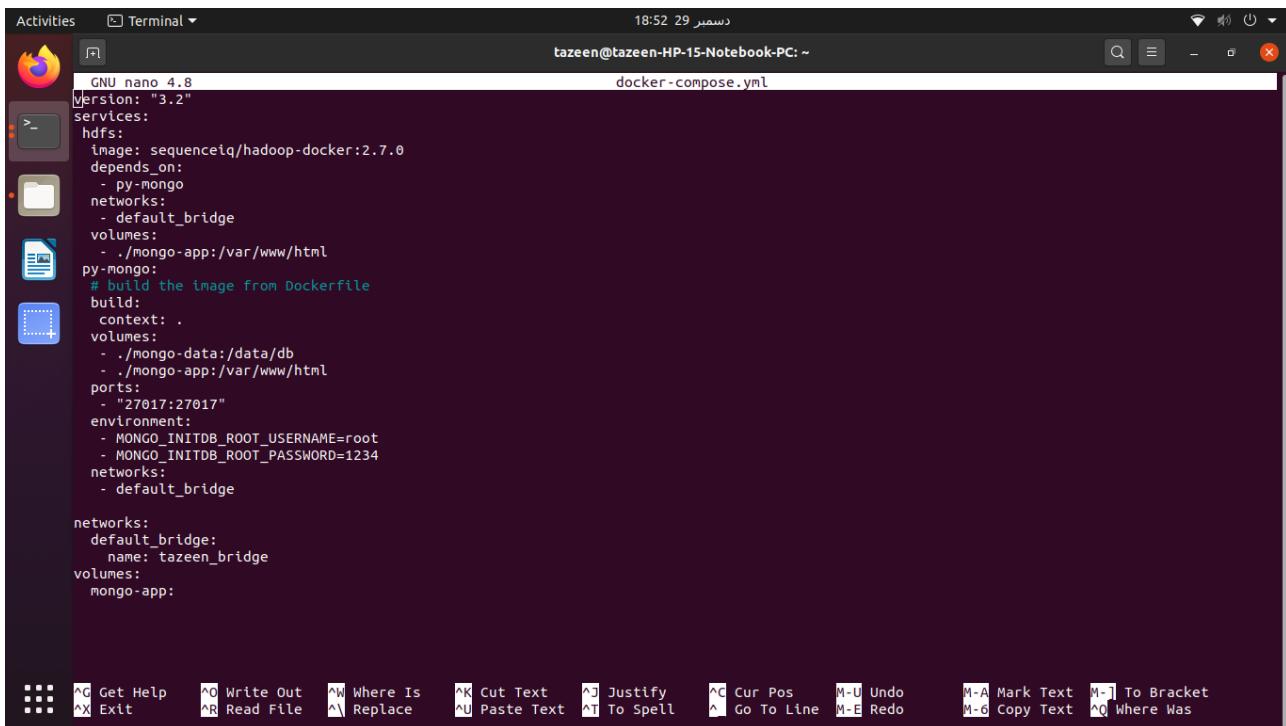
```
Activities Terminal 16:38 26 دسمبر tazeen@tazeen-HP-15-Notebook-PC:~$ docker pull sequenceiq/hadoop-docker:2.7.0
2.7.0: Pulling from sequenceiq/hadoop-docker
Image docker.io/sequenceiq/hadoop-docker:2.7.0 uses outdated schema1 manifest format. Please upgrade to a schema2 image for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
b253335dcf03: Pulling fs layer
a3ed95cae02: Pulling fs layer
69623ef05416: Pulling fs layer
63aebddf4bce: Pulling fs layer
46305a4cd4d: Pulling fs layer
70ff65ec2366: Pulling fs layer
72accd282f3: Pulling fs layer
5298ddb3b339: Pulling fs layer
ec461d25c2ea: Pulling fs layer
315b476b23a4: Pull complete
6e6acc31f8b1: Pull complete
38a227158d97: Pull complete
319a3b8afa25: Pull complete
11e1e16af8f3: Pull complete
834533551a37: Pull complete
c24255b6d9f4: Pull complete
8b4ea3c67dc2: Pull complete
40ba2c2cdf73: Pull complete
5424a04bc240: Pull complete
7df43f09096d: Pull complete
b34787ee2fde: Pull complete
4eaa47927d15: Pull complete
cb95b9da9646: Pull complete
e495e287a108: Pull complete
3158c449a54c: Pull complete
33b5a5de9544: Pull complete
d6f46cf55f0f: Pull complete
40c19fb76cf7d: Pull complete
018a1f3d7249: Pull complete
40f52c973507: Pull complete
49dca4de47eb: Pull complete
d26082bd2aa9: Pull complete
c4f97d87af86: Pull complete
fb839f93fc0f: Pull complete
43661864505e: Pull complete
```

```
Activities Terminal 16:38 26 دسمبر tazeen@tazeen-HP-15-Notebook-PC:~$ docker pull sequenceiq/hadoop-docker:2.7.0
2.7.0: Pulling from sequenceiq/hadoop-docker
319a3b8afa25: Pull complete
11e1e16af8f3: Pull complete
834533551a37: Pull complete
c24255b6d9f4: Pull complete
8b4ea3c67dc2: Pull complete
40ba2c2cdf73: Pull complete
5424a04bc240: Pull complete
7df43f09096d: Pull complete
b34787ee2fde: Pull complete
4eaa47927d15: Pull complete
cb95b9da9646: Pull complete
e495e287a108: Pull complete
3158c449a54c: Pull complete
33b5a5de9544: Pull complete
d6f46cf55f0f: Pull complete
40c19fb76cf7d: Pull complete
018a1f3d7249: Pull complete
40f52c973507: Pull complete
49dca4de47eb: Pull complete
d26082bd2aa9: Pull complete
c4f97d87af86: Pull complete
fb839f93fc0f: Pull complete
43661864505e: Pull complete
d8908a3648e: Pull complete
af8b6686deb23: Pull complete
c1214abd7b96: Pull complete
9d00f27ba8d2: Pull complete
09f787a7573b: Pull complete
4e86267d5247: Pull complete
3876cba35aed: Pull complete
23df48ffdb39: Pull complete
646aedbc2bb6: Pull complete
60a65f8179cf: Pull complete
046b321f8081: Pull complete
Digest: sha256:a40761746eca036fee6aafdf9fdbd6878ac3dd9a7cd83c0f3f5d8a0e6350c76a
Status: Downloaded newer image for sequenceiq/hadoop-docker:2.7.0
tazeen@tazeen-HP-15-Notebook-PC:~$ 
```

Updating docker compose to add another container for hadoop which links to pymongo container

```
Activities Terminal نسمر 28 04:43 tazeen@tazeen-HP-15-Notebook-PC:~  
tazeen@tazeen-HP-15-Notebook-PC:~$ sudo nano docker-compose.yml  
WARNING: The PWM variable is not set. Defaulting to a blank string.  
Creating network "tazeen_bridge" with the default driver  
Creating tazeen_py-mongo_1 ... done  
Creating tazeen_hdfs_1 ... done  
Attaching to tazeen_py-mongo_1, tazeen_hdfs_1  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:18.231+00:00"}, "s": "I", "c": "CONTROL", "id": 23285, "ctx": "main", "msg": "Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}  
hdfs_1 | /  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:18.234+00:00"}, "s": "W", "c": "ASIO", "id": 22601, "ctx": "main", "msg": "No TransportLayer configured during NetworkInterface startup"}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:18.235+00:00"}, "s": "I", "c": "NETWORK", "id": 4648601, "ctx": "main", "msg": "Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}  
hdfs_1 | Starting sshd: [ OK ]  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:18.235+00:00"}, "s": "I", "c": "STORAGE", "id": 4615611, "ctx": "initandlisten", "msg": "MongoDB starting", "attr": {"pid": 1, "port": 27017, "dbPath": "/data/db", "architecture": "64-bit", "host": "if7d90ced38e"}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:18.235+00:00"}, "s": "I", "c": "CONTROL", "id": 23403, "ctx": "initandlisten", "msg": "Build Info", "attr": {"buildInfo": {"version": "4.4.2", "gitVersion": "15e73dc5738d2278b688f8929aee605fe4279b0e", "openSSLVersion": "OpenSSL 1.1.1 11 Sep 2018", "modules": [], "allocator": "tcmalloc", "environment": {"distmod": "ubuntu1804", "distarch": "x86_64", "target_arch": "x86_64"}}}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:18.235+00:00"}, "s": "I", "c": "CONTROL", "id": 51765, "ctx": "initandlisten", "msg": "Operating System", "attr": {"os": {"name": "Ubuntu", "version": "18.04"}}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:18.235+00:00"}, "s": "I", "c": "CONTROL", "id": 21951, "ctx": "initandlisten", "msg": "Options set by command line", "attr": {"options": {"net": {"bindIp": "*"}, "security": {"authorization": "enabled"}}}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:18.237+00:00"}, "s": "I", "c": "STORAGE", "id": 22270, "ctx": "initandlisten", "msg": "Storage engine to use detected by data files", "attr": {"dbpath": "/data/db", "storageEngine": "wiredtiger"}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:18.237+00:00"}, "s": "I", "c": "STORAGE", "id": 22297, "ctx": "initandlisten", "msg": "Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem", "tags": ["startUpWarnings"]}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:18.237+00:00"}, "s": "I", "c": "STORAGE", "id": 22315, "ctx": "initandlisten", "msg": "Opening WiredTiger", "attr": {"config": "create,cache_size=5444M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimunm=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress,compact_progress]"}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:19.268+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1609112239:268056][1:0x7f0d75734ac0]", "txn_recover": "[WT_VERB_RECOVERY_PROGRESS] Recovering log 3 through 4"}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:19.326+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1609112239:326158][1:0x7f0d75734ac0]", "txn_recover": "[WT_VERB_RECOVERY_PROGRESS] Recovering log 4 through 4"}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:19.417+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1609112239:4176441][1:0x7f0d75734ac0]", "txn_recover": "[WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Main recovery lo
```

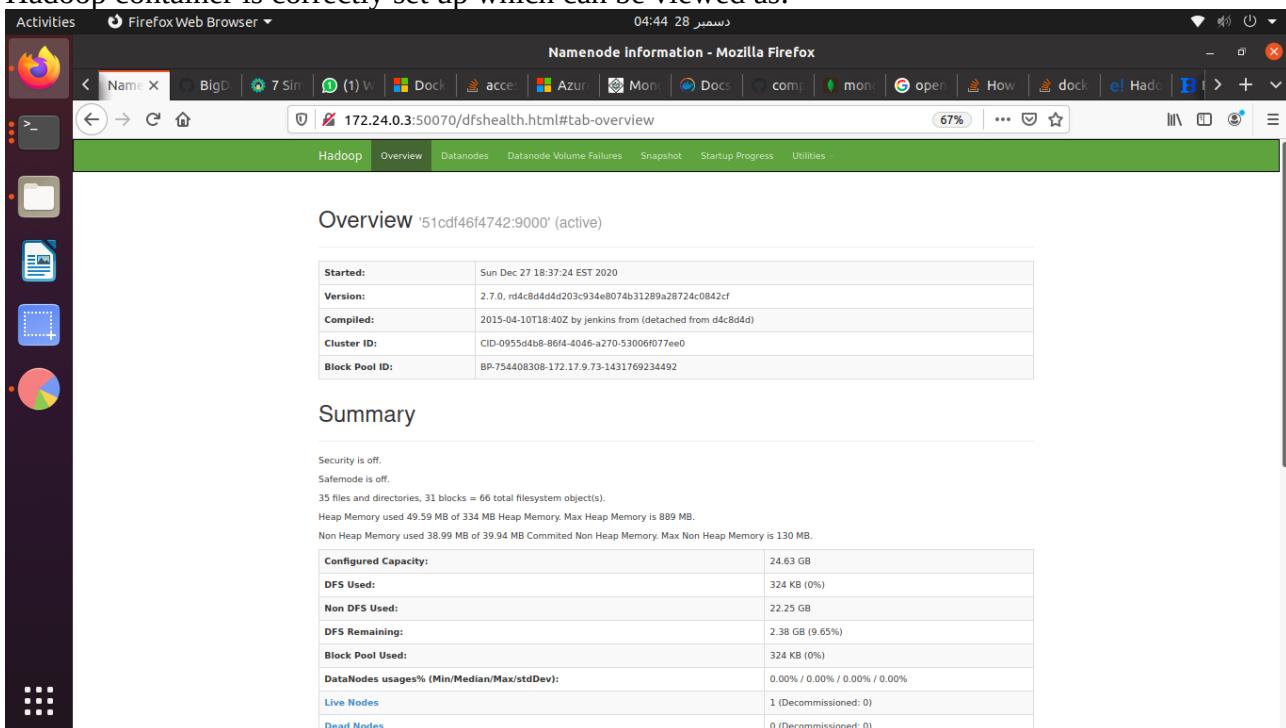
```
Activities Terminal نسمر 28 04:43 tazeen@tazeen-HP-15-Notebook-PC:~  
op: starting at 3/14464 to 4/256"]]  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:19.536+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1609112239:536855][1:0x7f0d75734ac0]", "txn_recover": "[WT_VERB_RECOVERY_PROGRESS] Recovering log 3 through 4"}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:19.631+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1609112239:631734][1:0x7f0d75734ac0]", "txn_recover": "[WT_VERB_RECOVERY_PROGRESS] Recovering log 4 through 4"}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:19.713+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1609112239:713227][1:0x7f0d75734ac0]", "txn_recover": "[WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global recovery timestamp: (0, 0)"}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:19.713+00:00"}, "s": "I", "c": "STORAGE", "id": 22430, "ctx": "initandlisten", "msg": "WiredTiger message", "attr": {"message": "[1609112239:713292][1:0x7f0d75734ac0]", "txn_recover": "[WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global oldest timestamp: (0, 0)"}}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:20.689+00:00"}, "s": "I", "c": "STORAGE", "id": 4795906, "ctx": "initandlisten", "msg": "WiredTiger opened", "attr": {"durationMillis": 12452}}]  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:20.689+00:00"}, "s": "I", "c": "RECOVERY", "id": 23987, "ctx": "initandlisten", "msg": "WiredTiger recoveryTimestamp", "attr": {"recoveryTimestamp": "$timestamp": {"$t": 0, "i": 0}}}]  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:20.691+00:00"}, "s": "I", "c": "STORAGE", "id": 4366408, "ctx": "initandlisten", "msg": "No table logging settings modifications are required for existing WiredTiger tables", "attr": {"loggingEnabled": true}}]  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:20.692+00:00"}, "s": "I", "c": "STORAGE", "id": 22262, "ctx": "initandlisten", "msg": "Timestamp monitor starting"}]  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:20.771+00:00"}, "s": "I", "c": "STORAGE", "id": 20536, "ctx": "initandlisten", "msg": "Flow Control is enabled on this deployment"}]  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:20.773+00:00"}, "s": "I", "c": "FTDC", "id": 20625, "ctx": "initandlisten", "msg": "Initializing full-time diagnostic data capture", "attr": {"dataDirectory": "/data/db/diagnostic.data"}]}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:20.776+00:00"}, "s": "I", "c": "NETWORK", "id": 23015, "ctx": "listener", "msg": "Listening on", "attr": {"address": "/tmp/mongodb-27017.sock"}]}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:20.776+00:00"}, "s": "I", "c": "NETWORK", "id": 23015, "ctx": "listener", "msg": "Listening on", "attr": {"address": "0.0.0.0"}]}  
py-mongo_1 | {"t": {"$date": "2020-12-27T23:37:20.776+00:00"}, "s": "I", "c": "NETWORK", "id": 23016, "ctx": "listener", "msg": "Waiting for connections", "attr": {"port": 27017, "ssl": "off"}]}  
hdfs_1 | Starting namenodes on [51cdf46f4742]  
hdfs_1 | 51cdf46f4742: starting namenode, logging to /usr/local/hadoop/logs/hadoop-root-namenode-51cdf46f4742.out  
hdfs_1 | localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-51cdf46f4742.out  
hdfs_1 | Starting secondary namenodes [0.0.0.0]  
hdfs_1 | 0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-root-secondarynamenode-51cdf46f4742.out  
hdfs_1 | starting yarn daemons  
hdfs_1 | starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-resourcemanager-51cdf46f4742.out  
hdfs_1 | localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-51cdf46f4742.out
```



```
GNU nano 4.8
version: "3.2"
services:
  hdfs:
    image: sequenceiq/hadoop-docker:2.7.0
    depends_on:
      - py-mongo
    networks:
      - default_bridge
    volumes:
      - ./mongo-app:/var/www/html
    py-mongo:
      # build the image from Dockerfile
      build:
        context: .
        volumes:
          - ./mongo-data:/data/db
          - ./mongo-app:/var/www/html
      ports:
        - "27017:27017"
      environment:
        - MONGO_INITDB_ROOT_USERNAME=root
        - MONGO_INITDB_ROOT_PASSWORD=1234
    networks:
      - default_bridge

networks:
  default_bridge:
    name: tazeen_bridge
volumes:
  mongo-app:
```

Hadoop container is correctly set up which can be viewed as:



Namenode Information - Mozilla Firefox

172.24.0.3:50070/dfshealth.html#tab-overview

Overview '51cdf46f4742:9000' (active)

Started:	Sun Dec 27 18:37:24 EST 2020
Version:	2.7.0, rd4c8d4d4d203c934e8074b31289a28724c0842cf
Compiled:	2015-04-10T18:40Z by jenkins from (detached from d4c8d4d)
Cluster ID:	CID-09554b8-86fa-4046-a270-53006f077ee0
Block Pool ID:	BP-754408308-172.17.9.73-1431769234492

Summary

Configured Capacity: 24.63 GB

DFS Used: 324 KB (0%)

Non DFS Used: 22.25 GB

DFS Remaining: 2.38 GB (9.65%)

Block Pool Used: 324 KB (0%)

DataNodes usages% (Min/Median/Max/stdDev): 0.00% / 0.00% / 0.00% / 0.00%

Live Nodes: 1 (Decommissioned: 0)

Dead Nodes: 0 (Decommissioned: 0)

The network used to connect both containers is: tazeen_bridge which is a user defined bridge network. Both containers are connected on the same network which can be seen in below screenshot.

```
Activities Terminal ٠٤:٤٤ ٢٨ دسمبر tazeen@tazeen-HP-15-Notebook-PC:~$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
51cdf46f4742      sequenceiq/hadoop-docker:2.7.0    "/etc/bootstrap.sh -d"   3 minutes ago     Up 3 minutes       2122/tcp, 8030-8033/tcp,
8040/tcp, 8042/tcp, 8088/tcp, 19888/tcp, 49707/tcp, 50010/tcp, 50020/tcp, 50070/tcp, 50075/tcp, 50090/tcp   tazeen_hdfs_1
1f7d90ced38e      tazeen_py-mongo                "docker-entrypoint.s..."  3 minutes ago     Up 3 minutes       0.0.0.0:27017->27017/tcp
                                                               tazeen_py-mongo_1

tazeen@tazeen-HP-15-Notebook-PC:~$ docker network inspect tazeen_bridge
[
  {
    "Name": "tazeen_bridge",
    "Id": "6c17e4b209def233e54056319e63b36034c8842f129c8d0bd3284e1440f1bc52",
    "Created": "2020-12-28T04:37:14.150030627+05:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.24.0.0/16",
          "Gateway": "172.24.0.1"
        }
      ],
      "Internal": false,
      "Attachable": true,
      "Ingress": false,
      "ConfigFrom": {
        "Network": ""
      },
      "ConfigOnly": false,
      "Containers": {
        "1f7d90ced38e3429d83b5c6408f950a15713dccb00b3acc2304ed6ce91f1da5b": {
          "Name": "tazeen_py-mongo_1",
          "EndpointID": "722f1eb78031dd42ab9b877d100dd5d9ac8c9d8b0f8bd52f9311881c80530da1",
          "MacAddress": "02:42:ac:18:00:02",
          "IPv4Address": "172.24.0.2/16"
        }
      }
    }
  }
]
```

```
Activities Terminal ٠٤:٤٤ ٢٨ دسمبر tazeen@tazeen-HP-15-Notebook-PC:~$ docker network inspect tazeen_bridge
[
  {
    "Name": "tazeen_bridge",
    "Id": "6c17e4b209def233e54056319e63b36034c8842f129c8d0bd3284e1440f1bc52",
    "Created": "2020-12-28T04:37:14.150030627+05:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.24.0.0/16",
          "Gateway": "172.24.0.1"
        }
      ],
      "Internal": false,
      "Attachable": true,
      "Ingress": false,
      "ConfigFrom": {
        "Network": ""
      },
      "ConfigOnly": false,
      "Containers": {
        "1f7d90ced38e3429d83b5c6408f950a15713dccb00b3acc2304ed6ce91f1da5b": {
          "Name": "tazeen_py-mongo_1",
          "EndpointID": "722f1eb78031dd42ab9b877d100dd5d9ac8c9d8b0f8bd52f9311881c80530da1",
          "MacAddress": "02:42:ac:18:00:02",
          "IPv4Address": "172.24.0.2/16",
          "IPv6Address": ""
        },
        "51cdf46f47424c5c2e3df85fd715803c8c8b94c89b5cf0ff501e7ca6dff21cb": {
          "Name": "tazeen_hdfs_1",
          "EndpointID": "86984e910a0ab1c5a517e1474f430dd62d68322f154a8fb007fe2f7587fbf61c",
          "MacAddress": "02:42:ac:18:00:03",
          "IPv4Address": "172.24.0.3/16",
          "IPv6Address": ""
        }
      },
      "Options": {},
      "Labels": {
        "com.docker.compose.network": "tazeen_bridge",
        "com.docker.compose.project": "tazeen",
        "com.docker.compose.version": "1.27.4"
      }
    }
  }
]
```

Now getting data for hadoop batch processing from mongodb. First checking the ip address to connect to database using pymongo container

```
tazeen@tazeen-HP-15-Notebook-PC:~$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
167c92585cc2      sequenceiq/hadoop-docker:2.7.0    "/etc/bootstrap.sh -d"   39 hours ago       Up 2 hours         tazeen_hdfs_1
8040/tcp, 8042/tcp, 8088/tcp, 19888/tcp, 49707/tcp, 50010/tcp, 50020/tcp, 50070/tcp, 50075/tcp, 50090/tcp
60bcea4e3233      tazeen_py-mongo           "docker-entrypoint.s..."  39 hours ago       Up 2 hours         tazeen_py_mongo_1

tazeen@tazeen-HP-15-Notebook-PC:~$ docker inspect 60bcea4e3233
[{"Id": "60bcea4e3233f1b8f9ceeeec0bdd5e3a80587fc37dfd98c123dcde68fd26f429c", "Created": "2020-12-28T00:29:00.015482546Z", "Path": "docker-entrypoint.sh", "Args": ["mongod"], "State": {"Status": "running", "Running": true, "Paused": false, "Restarting": false, "OOMKilled": false, "Dead": false, "Pid": 1980, "ExitCode": 0, "Error": "", "StartedAt": "2020-12-29T13:49:52.127086248Z", "FinishedAt": "2020-12-29T18:49:02.039602681+05:00"}, "Image": "sha256:575d0cd1ce89a06cc857aa7acb3bcab8eb4a964975d60f820c72d84c0f25702d", "ResolvConfPath": "/var/lib/docker/containers/60bcea4e3233f1b8f9ceeeec0bdd5e3a80587fc37dfd98c123dcde68fd26f429c/resolv.conf", "HostnamePath": "/var/lib/docker/containers/60bcea4e3233f1b8f9ceeeec0bdd5e3a80587fc37dfd98c123dcde68fd26f429c/hostname", "HostPath": "/var/lib/docker/containers/60bcea4e3233f1b8f9ceeeec0bdd5e3a80587fc37dfd98c123dcde68fd26f429c/hosts", "LogPath": "/var/lib/docker/containers/60bcea4e3233f1b8f9ceeeec0bdd5e3a80587fc37dfd98c123dcde68fd26f429c/60bcea4e3233f1b8f9ceeeec0bdd5e3a80587fc37dfd98c123dcde68fd26f429c.json.log", "Name": "/tazeen_py-mongo_1"}
```

```
tazeen@tazeen-HP-15-Notebook-PC:~$ docker inspect 60bcea4e3233f1b8f9ceeeec0bdd5e3a80587fc37dfd98c123dcde68fd26f429c
[{"SandboxKey": "/var/run/docker/netns/c8e84133c070", "SecondaryIPAddresses": null, "SecondaryIPv6Addresses": null, "EndpointID": "", "Gateway": "", "GlobalIPv6Address": "", "GlobalIPv6PrefixLen": 0, "IPAddress": "", "IPPrefixLen": 0, "IPv6Gateway": "", "MacAddress": "", "Networks": {"tazeen_bridge": {"IPAMConfig": null, "Links": null, "Aliases": [{"Name": "py-mongo", "IP": "60bcea4e3233"}], "NetworkID": "02014f58a32fb5e8163e33eacf1782f8279a40ba353db237b6b6d790b66b22b4", "EndpointID": "c967331fa810b36be9def589d674bcf078467986d52ad2e4d7ec02138e54c7ca", "Gateway": "172.28.0.1", "IPAddress": "172.28.0.2", "IPPrefixLen": 16, "IPv6Gateway": "", "GlobalIPv6Address": "", "GlobalIPv6PrefixLen": 0, "MacAddress": "02:42:ac:1c:00:02", "DriverOpts": null}}, {"Name": "py-mongo"}]
tazeen@tazeen-HP-15-Notebook-PC:~$ cd mongo-app
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$ sudo nano batch-data.py
[sudo] password for tazeen:
tazeen@tazeen-HP-15-Notebook-PC:~/mongo-app$
```

Creating multiple python scripts to export mongodb data for hadoop container. Because scripts take a lot of time in processing and stuck due to less resources. Each script creates text file data for specific months, total data is used from 1/1/2017 – 31/5/2017 and 1/1/2018 – 31/5/2018. The screenshots for batch-data.py is added below, on it's execution mongodb data is exported in text files shared using same bridge network.

A screenshot of a Linux desktop environment showing a terminal window titled "mongo-app/batch-data.py". The terminal is running under the user "tazeen" at the IP address "c0d1b0d988fa". The script content is as follows:

```
GNU nano 4.8
from pymongo import MongoClient, errors
import csv
import datetime
import json

# global variables for MongoDB host (default port is 27017)
DOMAIN = '172.18.0.2'
PORT = 27017

# use a try-except indentation to catch MongoClient() errors
try:
    # try to instantiate a client instance
    client = MongoClient(
        host = [ str(DOMAIN) + ":" + str(PORT) ],
        serverSelectionTimeoutMS = 3000, # 3 second timeout
        username = "root",
        password = "1234",
    )
    db = client.airline;
    batchData = db.flights.find({'$or':[{'$and':[{'FL_DATE':{'$gte':'2018-01-01'},{'FL_DATE':{'$lte':'2018-03-31'}}]},{'$and':[{'FL_DATE':{'$gte':'2017-01-01'},{'FL_DATE':{'$lte':'2017-03-31'}}]}]}, { '$OR':[{'CANCELLED':1, 'ARR_DELAY':1, 'FL_DATE':1, '_id':0 }]} );
    batchStr = str();
    for document in batchData:
        print (document);
        if batchStr:
            batchStr = batchStr + ";";
            batchStr = batchStr + json.dumps(document);
            #batchJson.append(document);
    with open('batch-data.txt', 'w') as outfile:
        outfile.write(batchStr)
```

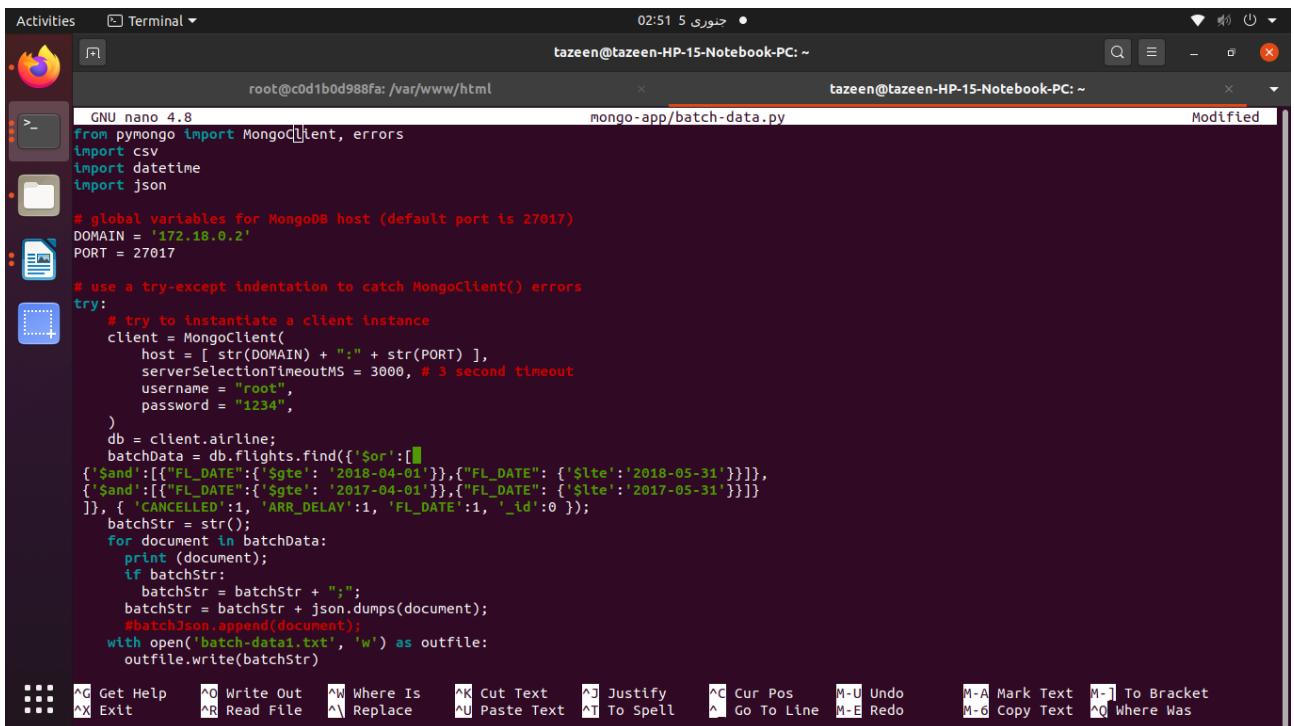
A screenshot of a Linux desktop environment showing the same terminal window as above, but with an additional "except" block added to the script:

```
GNU nano 4.8
# global variables for MongoDB host (default port is 27017)
DOMAIN = '172.18.0.2'
PORT = 27017

# use a try-except indentation to catch MongoClient() errors
try:
    # try to instantiate a client instance
    client = MongoClient(
        host = [ str(DOMAIN) + ":" + str(PORT) ],
        serverSelectionTimeoutMS = 3000, # 3 second timeout
        username = "root",
        password = "1234",
    )
    db = client.airline;
    batchData = db.flights.find({'$or':[{'$and':[{'FL_DATE':{'$gte':'2018-01-01'},{'FL_DATE':{'$lte':'2018-03-31'}}]},{'$and':[{'FL_DATE':{'$gte':'2017-01-01'},{'FL_DATE':{'$lte':'2017-03-31'}}]}]}, { '$OR':[{'CANCELLED':1, 'ARR_DELAY':1, 'FL_DATE':1, '_id':0 }]} );
    batchStr = str();
    for document in batchData:
        print (document);
        if batchStr:
            batchStr = batchStr + ";";
            batchStr = batchStr + json.dumps(document);
            #batchJson.append(document);
    with open('batch-data.txt', 'w') as outfile:
        outfile.write(batchStr)

except errors.ServerSelectionTimeoutError as err:
    # catch pymongo.errors.ServerSelectionTimeoutError
    print ("pymongo ERROR:", err)
```

After execution, same file is updated for different data.

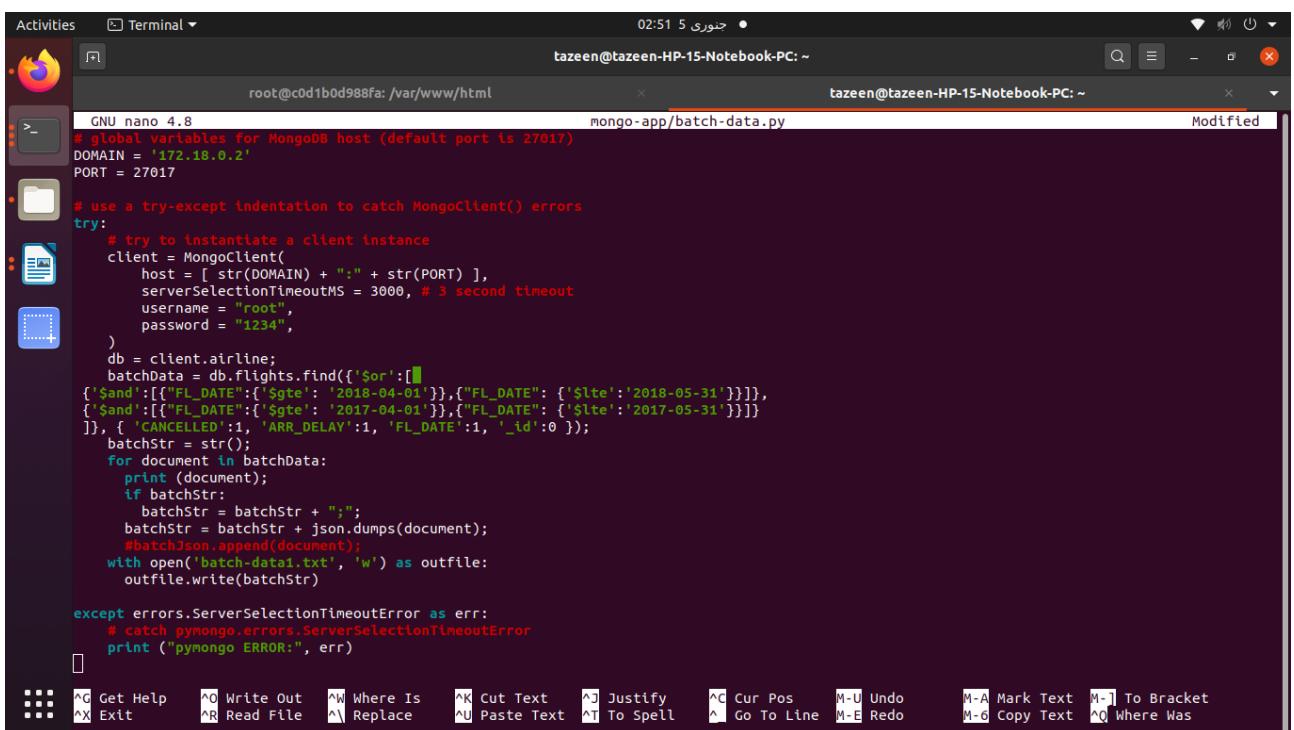


```
GNU nano 4.8
from pymongo import MongoClient, errors
import csv
import datetime
import json

# global variables for MongoDB host (default port is 27017)
DOMAIN = '172.18.0.2'
PORT = 27017

# use a try-except indentation to catch MongoClient() errors
try:
    # try to instantiate a client instance
    client = MongoClient(
        host = [ str(DOMAIN) + ":" + str(PORT) ],
        serverSelectionTimeoutMS = 3000, # 3 second timeout
        username = "root",
        password = "1234",
    )
    db = client.airline;
    batchData = db.flights.find({'$or':[{'$and':[['FL_DATE':{'$gte':'2018-04-01'}], {'$and':[['FL_DATE':{'$lte':'2018-05-31'}]]}], {'$and':[['FL_DATE':{'$gte':'2017-04-01'}], {'$and':[['FL_DATE':{'$lte':'2017-05-31'}]]}]}, { '$and':[['FL_DATE':{'$gt':'2017-05-31'}], {'$and':[['ARR_DELAY':1, 'FL_DATE':1, '_id':0 ]]}]}, batchStr = str();
    for document in batchData:
        print (document);
        if batchStr:
            batchStr = batchStr + ";";
            batchStr = batchStr + json.dumps(document);
            #batchJson.append(document);
    with open('batch-data1.txt', 'w') as outfile:
        outfile.write(batchStr)

except errors.ServerSelectionTimeoutError as err:
    # catch pymongo.errors.ServerSelectionTimeoutError
    print ("pymongo ERROR:", err)
```



```
GNU nano 4.8
# global variables for MongoDB host (default port is 27017)
DOMAIN = '172.18.0.2'
PORT = 27017

# use a try-except indentation to catch MongoClient() errors
try:
    # try to instantiate a client instance
    client = MongoClient(
        host = [ str(DOMAIN) + ":" + str(PORT) ],
        serverSelectionTimeoutMS = 3000, # 3 second timeout
        username = "root",
        password = "1234",
    )
    db = client.airline;
    batchData = db.flights.find({'$or':[{'$and':[['FL_DATE':{'$gte':'2018-04-01'}], {'$and':[['FL_DATE':{'$lte':'2018-05-31'}]]}], {'$and':[['FL_DATE':{'$gte':'2017-04-01'}], {'$and':[['FL_DATE':{'$lte':'2017-05-31'}]]}]}, { '$and':[['FL_DATE':{'$gt':'2017-05-31'}], {'$and':[['ARR_DELAY':1, 'FL_DATE':1, '_id':0 ]]}]}, batchStr = str();
    for document in batchData:
        print (document);
        if batchStr:
            batchStr = batchStr + ";";
            batchStr = batchStr + json.dumps(document);
            #batchJson.append(document);
    with open('batch-data1.txt', 'w') as outfile:
        outfile.write(batchStr)

except errors.ServerSelectionTimeoutError as err:
    # catch pymongo.errors.ServerSelectionTimeoutError
    print ("pymongo ERROR:", err)
```

Executing the scripts by the following command

```

Activities Terminal ٠٣:٥٥ ٣٠ سمبر
root@60bcea4e3233:/var/www/html
tazeen@tazeen-HP-15-Notebook-PC:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
167c92585cc2      sequenceiq/hadoop-docker:2.7.0   "/etc/bootstrap.sh -d"   45 hours ago       Up 11 minutes      2122/tcp, 8030-8033/tcp,
8040/tcp, 8042/tcp, 8088/tcp, 19888/tcp, 49707/tcp, 50010/tcp, 50020/tcp, 50070/tcp, 50075/tcp, 50090/tcp   tazeen_hdfs_1
60bcea4e3233      tazeen_py-mongo                "docker-entrypoint.s..."  46 hours ago       Up 11 minutes      0.0.0.0:27017->27017/tcp
                                                              
tazeen@tazeen-HP-15-Notebook-PC:~$ docker exec -it 60bcea4e3233 /bin/bash
root@60bcea4e3233:/# cd /var/www/html
root@60bcea4e3233:/var/www/html# python3 batch-data.py

```

```

Activities Terminal ١٧:٥٢ ٣ جویر
root@79f4d4bad0c8:/var/www/html
root@79f4d4bad0c8:/var/www/html# python3 batch-data.py
SED_TIME': '176.0', 'AIR_TIME': '148.0', 'DISTANCE': '1189.0', 'CARRIER_DELAY': '', 'WEATHER_DELAY': '', 'SECURITY_DELAY': '',
'LATE_AIRCRAFT_DELAY': '', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2945', 'ORIGIN': 'LAS', 'DEST': 'JFK', 'CRS_DEP_TIME': '1554', 'DEP_TIME':
'1554.0', 'DEP_DELAY': '0.0', 'TAXI_OUT': '12.0', 'WHEELS_OFF': '1606.0', 'WHEELS_ON': '2336.0', 'TAXI_IN': '6.0', 'CRS_ARR_TIME': '2357', 'ARR_
TIME': '2342.0', 'ARR_DELAY': '-15.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '303.0', 'ACTUAL_
ELAPSED_TIME': '288.0', 'AIR_TIME': '270.0', 'DISTANCE': '2248.0', 'CARRIER_DELAY': '', 'WEATHER_DELAY': '', 'NAS_DELAY': '',
'SECURITY_DELAY': '', 'LATE_AIRCRAFT_DELAY': '', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2949', 'ORIGIN': 'BOS', 'DEST': 'SAV', 'CRS_DEP_TIME': '1500', 'DEP_TIME':
'1534.0', 'DEP_DELAY': '34.0', 'TAXI_OUT': '11.0', 'WHEELS_OFF': '1545.0', 'WHEELS_ON': '1753.0', 'TAXI_IN': '8.0', 'CRS_ARR_TIME': '1733', 'ARR_
TIME': '1801.0', 'ARR_DELAY': '28.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '153.0', 'ACTUAL_
ELAPSED_TIME': '147.0', 'AIR_TIME': '288.0', 'DISTANCE': '901.0', 'CARRIER_DELAY': '0.0', 'WEATHER_DELAY': '0.0', 'NAS_DELAY': '28.0', 'SECUR
ITY_DELAY': '0.0', 'LATE_AIRCRAFT_DELAY': '0.0', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2949', 'ORIGIN': 'BOS', 'DEST': 'SAV', 'CRS_DEP_TIME': '1810', 'DEP_TIME':
'1833.0', 'DEP_DELAY': '23.0', 'TAXI_OUT': '10.0', 'WHEELS_OFF': '1843.0', 'WHEELS_ON': '2051.0', 'TAXI_IN': '5.0', 'CRS_ARR_TIME': '2034', 'ARR_
TIME': '2056.0', 'ARR_DELAY': '22.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '144.0', 'ACTUAL_
ELAPSED_TIME': '143.0', 'AIR_TIME': '128.0', 'DISTANCE': '901.0', 'CARRIER_DELAY': '2.0', 'WEATHER_DELAY': '0.0', 'NAS_DELAY': '0.0', 'SECURI
TY_DELAY': '0.0', 'LATE_AIRCRAFT_DELAY': '20.0', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2950', 'ORIGIN': 'SAV', 'DEST': 'BOS', 'CRS_DEP_TIME': '1810', 'DEP_TIME':
'1833.0', 'DEP_DELAY': '23.0', 'TAXI_OUT': '10.0', 'WHEELS_OFF': '1843.0', 'WHEELS_ON': '2051.0', 'TAXI_IN': '5.0', 'CRS_ARR_TIME': '2034', 'ARR_
TIME': '2056.0', 'ARR_DELAY': '22.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '144.0', 'ACTUAL_
ELAPSED_TIME': '143.0', 'AIR_TIME': '128.0', 'DISTANCE': '901.0', 'CARRIER_DELAY': '2.0', 'WEATHER_DELAY': '0.0', 'NAS_DELAY': '0.0', 'SECURI
TY_DELAY': '0.0', 'LATE_AIRCRAFT_DELAY': '20.0', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2951', 'ORIGIN': 'TPA', 'DEST': 'SJU', 'CRS_DEP_TIME': '1820', 'DEP_TIME':
'1812.0', 'DEP_DELAY': '8.0', 'TAXI_OUT': '11.0', 'WHEELS_OFF': '1843.0', 'WHEELS_ON': '2051.0', 'TAXI_IN': '5.0', 'CRS_ARR_TIME': '2034', 'ARR_
TIME': '2056.0', 'ARR_DELAY': '10.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '174.0', 'ACTUAL_E
LAPSED_TIME': '172.0', 'AIR_TIME': '158.0', 'DISTANCE': '1237.0', 'CARRIER_DELAY': '', 'WEATHER_DELAY': '', 'NAS_DELAY': '', 'SECURITY_DELAY':
'', 'LATE_AIRCRAFT_DELAY': '', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2951', 'ORIGIN': 'TPA', 'DEST': 'SJU', 'CRS_DEP_TIME': '1810', 'DEP_TIME':
'1833.0', 'DEP_DELAY': '23.0', 'TAXI_OUT': '10.0', 'WHEELS_OFF': '1843.0', 'WHEELS_ON': '2051.0', 'TAXI_IN': '5.0', 'CRS_ARR_TIME': '2034', 'ARR_
TIME': '2056.0', 'ARR_DELAY': '22.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '144.0', 'ACTUAL_E
LAPSED_TIME': '143.0', 'AIR_TIME': '128.0', 'DISTANCE': '901.0', 'CARRIER_DELAY': '2.0', 'WEATHER_DELAY': '0.0', 'NAS_DELAY': '0.0', 'SECURI
TY_DELAY': '0.0', 'LATE_AIRCRAFT_DELAY': '20.0', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2952', 'ORIGIN': 'FLL', 'DEST': 'SJU', 'CRS_DEP_TIME': '1820', 'DEP_TIME':
'1812.0', 'DEP_DELAY': '8.0', 'TAXI_OUT': '11.0', 'WHEELS_OFF': '1843.0', 'WHEELS_ON': '2051.0', 'TAXI_IN': '5.0', 'CRS_ARR_TIME': '2034', 'ARR_
TIME': '2056.0', 'ARR_DELAY': '10.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '174.0', 'ACTUAL_E
LAPSED_TIME': '172.0', 'AIR_TIME': '158.0', 'DISTANCE': '1237.0', 'CARRIER_DELAY': '', 'WEATHER_DELAY': '', 'NAS_DELAY': '', 'SECURITY_DELAY':
'', 'LATE_AIRCRAFT_DELAY': '', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2953', 'ORIGIN': 'FLL', 'DEST': 'SJU', 'CRS_DEP_TIME': '1820', 'DEP_TIME':
'1812.0', 'DEP_DELAY': '8.0', 'TAXI_OUT': '11.0', 'WHEELS_OFF': '1843.0', 'WHEELS_ON': '2051.0', 'TAXI_IN': '5.0', 'CRS_ARR_TIME': '2034', 'ARR_
TIME': '2056.0', 'ARR_DELAY': '10.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '152.0', 'ACTUAL_E
LAPSED_TIME': '162.0', 'AIR_TIME': '139.0', 'DISTANCE': '1046.0', 'CARRIER_DELAY': '10.0', 'WEATHER_DELAY': '0.0', 'NAS_DELAY': '10.0', 'SECURI
TY_DELAY': '0.0', 'LATE_AIRCRAFT_DELAY': '0.0', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2953', 'ORIGIN': 'FLL', 'DEST': 'SJU', 'CRS_DEP_TIME': '1828', 'DEP_TIME':
'1838.0', 'DEP_DELAY': '10.0', 'TAXI_OUT': '19.0', 'WHEELS_OFF': '1857.0', 'WHEELS_ON': '2116.0', 'TAXI_IN': '4.0', 'CRS_ARR_TIME': '1100', 'ARR_
TIME': '1104.0', 'ARR_DELAY': '-10.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '174.0', 'ACTUAL_E
LAPSED_TIME': '172.0', 'AIR_TIME': '158.0', 'DISTANCE': '1237.0', 'CARRIER_DELAY': '', 'WEATHER_DELAY': '', 'NAS_DELAY': '', 'SECURITY_DELAY':
'', 'LATE_AIRCRAFT_DELAY': '', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2953', 'ORIGIN': 'FLL', 'DEST': 'SJU', 'CRS_DEP_TIME': '1828', 'DEP_TIME':
'1838.0', 'DEP_DELAY': '10.0', 'TAXI_OUT': '19.0', 'WHEELS_OFF': '1857.0', 'WHEELS_ON': '2116.0', 'TAXI_IN': '4.0', 'CRS_ARR_TIME': '1100', 'ARR_
TIME': '1104.0', 'ARR_DELAY': '-10.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '152.0', 'ACTUAL_E
LAPSED_TIME': '162.0', 'AIR_TIME': '139.0', 'DISTANCE': '1046.0', 'CARRIER_DELAY': '10.0', 'WEATHER_DELAY': '0.0', 'NAS_DELAY': '10.0', 'SECURI
TY_DELAY': '0.0', 'LATE_AIRCRAFT_DELAY': '0.0', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2954', 'ORIGIN': 'FLL', 'DEST': 'SJU', 'CRS_DEP_TIME': '1828', 'DEP_TIME':
'1838.0', 'DEP_DELAY': '10.0', 'TAXI_OUT': '19.0', 'WHEELS_OFF': '1857.0', 'WHEELS_ON': '2116.0', 'TAXI_IN': '4.0', 'CRS_ARR_TIME': '1100', 'ARR_
TIME': '1104.0', 'ARR_DELAY': '-10.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '152.0', 'ACTUAL_E
LAPSED_TIME': '162.0', 'AIR_TIME': '139.0', 'DISTANCE': '1046.0', 'CARRIER_DELAY': '10.0', 'WEATHER_DELAY': '0.0', 'NAS_DELAY': '10.0', 'SECURI
TY_DELAY': '0.0', 'LATE_AIRCRAFT_DELAY': '0.0', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2954', 'ORIGIN': 'FLL', 'DEST': 'SJU', 'CRS_DEP_TIME': '1828', 'DEP_TIME':
'1838.0', 'DEP_DELAY': '10.0', 'TAXI_OUT': '19.0', 'WHEELS_OFF': '1857.0', 'WHEELS_ON': '2116.0', 'TAXI_IN': '4.0', 'CRS_ARR_TIME': '1100', 'ARR_
TIME': '1104.0', 'ARR_DELAY': '-10.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '152.0', 'ACTUAL_E
LAPSED_TIME': '162.0', 'AIR_TIME': '139.0', 'DISTANCE': '1046.0', 'CARRIER_DELAY': '10.0', 'WEATHER_DELAY': '0.0', 'NAS_DELAY': '10.0', 'SECURI
TY_DELAY': '0.0', 'LATE_AIRCRAFT_DELAY': '0.0', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2954', 'ORIGIN': 'FLL', 'DEST': 'SJU', 'CRS_DEP_TIME': '1828', 'DEP_TIME':
'1838.0', 'DEP_DELAY': '10.0', 'TAXI_OUT': '19.0', 'WHEELS_OFF': '1857.0', 'WHEELS_ON': '2116.0', 'TAXI_IN': '4.0', 'CRS_ARR_TIME': '1100', 'ARR_
TIME': '1104.0', 'ARR_DELAY': '-10.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '152.0', 'ACTUAL_E
LAPSED_TIME': '162.0', 'AIR_TIME': '139.0', 'DISTANCE': '1046.0', 'CARRIER_DELAY': '10.0', 'WEATHER_DELAY': '0.0', 'NAS_DELAY': '10.0', 'SECURI
TY_DELAY': '0.0', 'LATE_AIRCRAFT_DELAY': '0.0', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2955', 'ORIGIN': 'MCO', 'DEST': 'JFK', 'CRS_DEP_TIME': '2100', 'DEP_TIME':
'2100.0', 'DEP_DELAY': '0.0', 'TAXI_OUT': '32.0', 'WHEELS_OFF': '2132.0', 'WHEELS_ON': '2344.0', 'TAXI_IN': '36.0', 'CRS_ARR_TIME': '2358', 'ARR_
TIME': '10.0', 'ARR_DELAY': '12.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '178.0', 'ACTUAL_E
LAPSED_TIME': '190.0', 'AIR_TIME': '122.0', 'DISTANCE': '944.0', 'CARRIER_DELAY': '', 'WEATHER_DELAY': '', 'NAS_DELAY': '', 'SECURITY_DELAY':
'', 'LATE_AIRCRAFT_DELAY': '', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2955', 'ORIGIN': 'MCO', 'DEST': 'JFK', 'CRS_DEP_TIME': '2100', 'DEP_TIME':
'2100.0', 'DEP_DELAY': '0.0', 'TAXI_OUT': '32.0', 'WHEELS_OFF': '2132.0', 'WHEELS_ON': '2344.0', 'TAXI_IN': '36.0', 'CRS_ARR_TIME': '2358', 'ARR_
TIME': '10.0', 'ARR_DELAY': '12.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '178.0', 'ACTUAL_E
LAPSED_TIME': '190.0', 'AIR_TIME': '122.0', 'DISTANCE': '944.0', 'CARRIER_DELAY': '', 'WEATHER_DELAY': '', 'NAS_DELAY': '', 'SECURITY_DELAY':
'', 'LATE_AIRCRAFT_DELAY': '', 'Unnamed': ''},
{'FL_DATE': '2018-06-30', 'OP_CARRIER': 'B6', 'OP_CARRIER_FL_NUM': '2955', 'ORIGIN': 'MCO', 'DEST': 'JFK', 'CRS_DEP_TIME': '2100', 'DEP_TIME':
'2100.0', 'DEP_DELAY': '0.0', 'TAXI_OUT': '32.0', 'WHEELS_OFF': '2132.0', 'WHEELS_ON': '2344.0', 'TAXI_IN': '36.0', 'CRS_ARR_TIME': '2358', 'ARR_
TIME': '10.0', 'ARR_DELAY': '12.0', 'CANCELLED': '0.0', 'CANCELLATION_CODE': '', 'DIVERTED': '0.0', 'CRS_ELAPSED_TIME': '178.0', 'ACTUAL_E
LAPSED_TIME': '190.0', 'AIR_TIME': '122.0', 'DISTANCE': '944.0', 'CARRIER_DELAY': '', 'WEATHER_DELAY': '', 'NAS_DELAY': '', 'SECURITY_DELAY':
'', 'LATE_AIRCRAFT_DELAY': '', 'Unnamed': ''}
root@79f4d4bad0c8:/var/www/html#

```

After executing the batch-data.py script with different data, two files batch-data.txt and batch-data1.txt are created which will be used in HDFS.

Now taking the above created files shared on the same bridge network as input files for hadoop and executing a map/reduce job. The map/reduce job calculates the average delay time minutes for each year. The data consists of canceled and delayed flights. Firstly, removing the canceled flights to find the correct delay time. Secondly, only considering the positive delay time as negative represents the flights were departed late. The map and reduce job files are shown below.

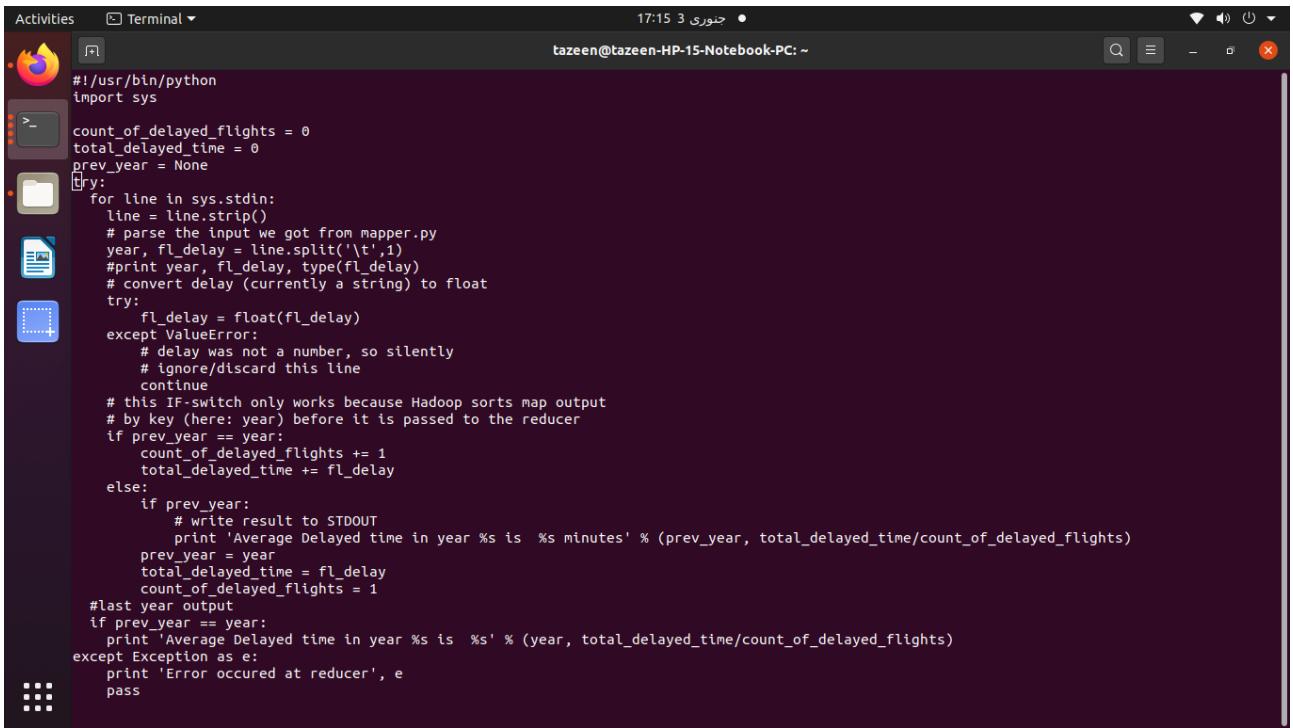
```
Activities Terminal ١٧:١٦ جوری ٣ tazeen@tazeen-HP-15-Notebook-PC:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
3145279367fe      sequenceiq/hadoop-docker:2.7.0    "/etc/bootstrap.sh -d"   41 hours ago       Up 4 hours         2122/tcp, 8030-8033/tcp,
8040/tcp, 8042/tcp, 8088/tcp, 19888/tcp, 49707/tcp, 50010/tcp, 50020/tcp, 50070/tcp, 50075/tcp, 50090/tcp   tazeen_hdfs_1
79f4d4bad0c8      tazeen_py-mongo                 "docker-entrypoint.s..."  41 hours ago       Up 4 hours         0.0.0.0:27017->27017/tcp
tazeen@tazeen-HP-15-Notebook-PC:~$ docker exec -it 3145279367fe /bin/bash
bash-4.1# vi /usr/local/hadoop/share/hadoop/mapreduce/mapper.py
bash-4.1#
```

```
Activities Terminal ١٧:١٢ جوری ٣ tazeen@tazeen-HP-15-Notebook-PC:~$ 
#!/usr/bin/python

import sys
import json

try:
    #input from json files exported from database
    data_arr = list()
    for line in sys.stdin:
        data_arr = data_arr + line.strip().split(';')
    for obj in data_arr:
        #print 'obj' , obj
        json_obj = eval(obj)
        #all delayed flights which were not cancelled
        if json_obj['CANCELLED'] and json_obj['ARR_DELAY'] and int(float(json_obj['CANCELLED'])) == 0:
            fl_delay = float(json_obj['ARR_DELAY'])
            if fl_delay > 0:
                #write the results to standard output STDOUT
                print json_obj['FL_DATE'].split('-')[0],'\t',fl_delay
except Exception as e:
    print 'Error occurred at mapper', e
pass
-- INSERT --
```

```
bash-4.1# vi /usr/local/hadoop/share/hadoop/mapreduce/reducer.py
```

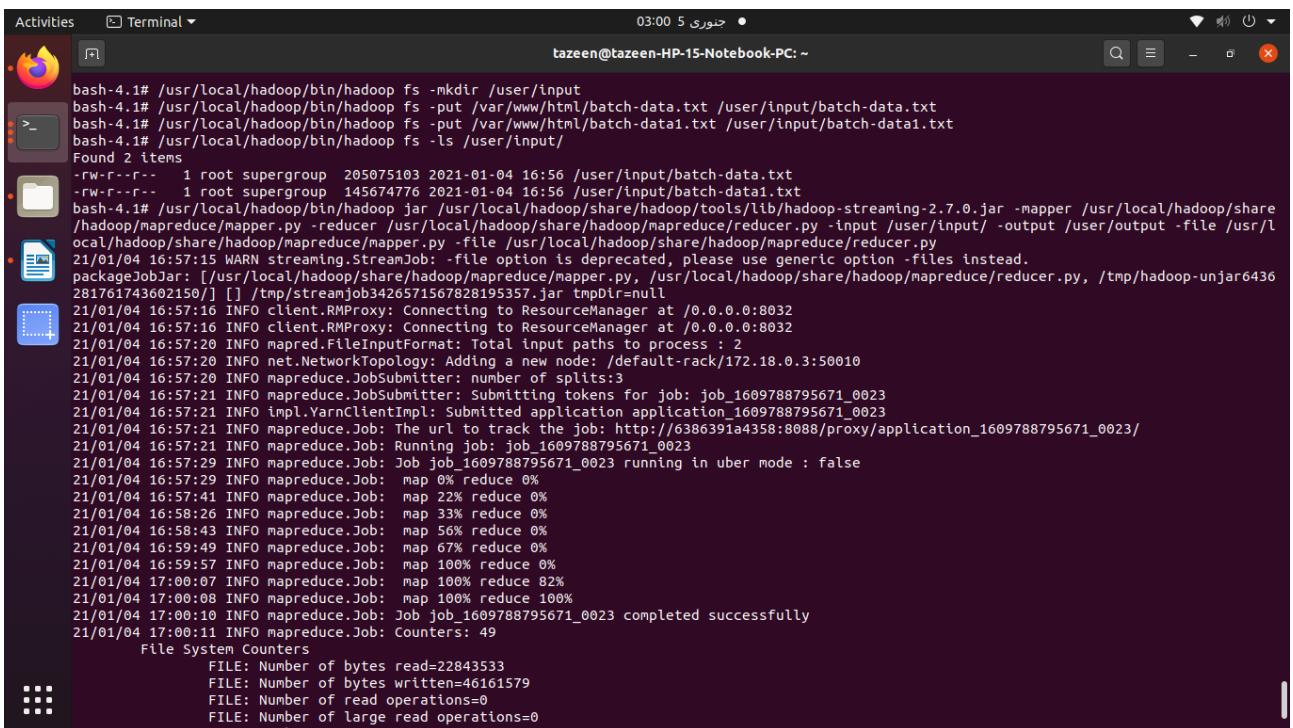


```
#!/usr/bin/python
import sys

count_of_delayed_flights = 0
total_delayed_time = 0
prev_year = None

try:
    for line in sys.stdin:
        line = line.strip()
        # parse the input we got from mapper.py
        year, fl_delay = line.split('\t',1)
        #print year, fl_delay, type(fl_delay)
        # convert delay (currently a string) to float
        try:
            fl_delay = float(fl_delay)
        except ValueError:
            # delay was not a number, so silently
            # ignore/discard this line
            continue
        # this IF-switch only works because Hadoop sorts map output
        # by key (here: year) before it is passed to the reducer
        if prev_year == year:
            count_of_delayed_flights += 1
            total_delayed_time += fl_delay
        else:
            if prev_year:
                # write result to STDOUT
                print 'Average Delayed time in year %s is %s minutes' % (prev_year, total_delayed_time/count_of_delayed_flights)
            prev_year = year
            total_delayed_time = fl_delay
            count_of_delayed_flights = 1
    #last year output
    if prev_year == year:
        print 'Average Delayed time in year %s is %s' % (year, total_delayed_time/count_of_delayed_flights)
except Exception as e:
    print 'Error occurred at reducer', e
    pass
```

Now putting these batch data files (batch-data.txt and batch-data1.txt) in HDFS. A new folder is created as /user/input which will be for hadoop input data and output will be saved in /user/output.



```
bash-4.1# /usr/local/hadoop/bin/hadoop fs -mkdir /user/input
bash-4.1# /usr/local/hadoop/bin/hadoop fs -put /var/www/html/batch-data.txt /user/input/batch-data.txt
bash-4.1# /usr/local/hadoop/bin/hadoop fs -put /var/www/html/batch-data1.txt /user/input/batch-data1.txt
Found 2 items
-rw-r--r-- 1 root supergroup 205075103 2021-01-04 16:56 /user/input/batch-data.txt
-rw-r--r-- 1 root supergroup 145674776 2021-01-04 16:56 /user/input/batch-data1.txt
bash-4.1# /usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.0.jar -mapper /usr/local/hadoop/share/hadoop/mapreduce/mapper.py -reducer /usr/local/hadoop/share/hadoop/mapreduce/reducer.py -input /user/input/ -output /user/output -file /usr/local/hadoop/share/hadoop/mapreduce/mapper.py -file /usr/local/hadoop/share/hadoop/mapreduce/reducer.py
21/01/04 16:57:15 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [/usr/local/hadoop/share/hadoop/mapreduce/mapper.py, /usr/local/hadoop/share/hadoop/mapreduce/reducer.py, /tmp/hadoop-unjar6436281761743602150/] [] /tmp/streamjob3426571567828195357.jar tmpDir=null
21/01/04 16:57:16 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/01/04 16:57:16 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/01/04 16:57:20 INFO mapred.FileInputFormat: Total input paths to process : 2
21/01/04 16:57:20 INFO net.NetworkTopology: Adding a new node: /default-rack/172.18.0.3:50010
21/01/04 16:57:20 INFO mapreduce.JobSubmitter: number of splits:3
21/01/04 16:57:21 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1609788795671_0023
21/01/04 16:57:21 INFO impl.YarnClientImpl: Submitted application application_1609788795671_0023
21/01/04 16:57:21 INFO mapreduce.Job: The url to track the job: http://6386391a4358:8088/proxy/application_1609788795671_0023/
21/01/04 16:57:21 INFO mapreduce.Job: Running job: job_1609788795671_0023
21/01/04 16:57:29 INFO mapreduce.Job: Job job_1609788795671_0023 running in uber mode : false
21/01/04 16:57:29 INFO mapreduce.Job: map 0% reduce 0%
21/01/04 16:57:29 INFO mapreduce.Job: map 22% reduce 0%
21/01/04 16:58:26 INFO mapreduce.Job: map 33% reduce 0%
21/01/04 16:58:43 INFO mapreduce.Job: map 56% reduce 0%
21/01/04 16:59:49 INFO mapreduce.Job: map 67% reduce 0%
21/01/04 16:59:57 INFO mapreduce.Job: map 100% reduce 0%
21/01/04 17:00:07 INFO mapreduce.Job: map 100% reduce 82%
21/01/04 17:00:08 INFO mapreduce.Job: map 100% reduce 100%
21/01/04 17:00:10 INFO mapreduce.Job: Job job_1609788795671_0023 completed successfully
21/01/04 17:00:11 INFO mapreduce.Job: Counters: 49
      File System Counters
          FILE: Number of bytes read=22843533
          FILE: Number of bytes written=46161579
          FILE: Number of read operations=0
          FILE: Number of large read operations=0
```

The job is successfully executed and output is saved in /user/output. The output can be viewed as 'cat /user/output/part-00000.txt' which contains the output on success.

```

Activities Terminal ٣:٠١ جوري ٥ tazeen@tazeen-HP-15-Notebook-PC: ~
Total megabyte-seconds taken by all reduce tasks=42672128
Map-Reduce Framework
  Map input records=2
  Map output records=1793260
  Map output bytes=19257007
  Map output materialized bytes=22843545
  Input split bytes=307
  Combine input records=0
  Combine output records=0
  Reduce input groups=2
  Reduce shuffle bytes=22843545
  Reduce input records=1793260
  Reduce output records=2
  Spilled Records=3586520
  Shuffled Maps =3
  Failed Shuffles=0
  Merged Map outputs=3
  GC time elapsed (ms)=513
  CPU time spent (ms)=135970
  Physical memory (bytes) snapshot=7076593664
  Virtual memory (bytes) snapshot=16343990272
  Total committed heap usage (bytes)=6973554688
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=421667254
File Output Format Counters
  Bytes Written=121
21/01/04 17:00:11 INFO streaming.StreamJob: Output directory: /user/output
bash-4.1# /usr/local/hadoop/bin/hadoop fs -cat /user/output/part-00000
Average Delayed time in year 2017 is 37.647597526 minutes
Average Delayed time in year 2018 is 36.3042229856 minutes
bash-4.1# 

```

Navigating to hadoop url to see the applications as below:

The screenshot shows a Firefox browser window with the URL <http://172.18.0.3:8088/cluster/apps/RUNNING>. The page title is "RUNNING Applications". On the left, there is a sidebar with a tree view of cluster metrics and application states. The main content area displays two tables: "Cluster Metrics" and "Scheduler Metrics". The "Cluster Metrics" table provides a summary of the cluster's resource usage. The "Scheduler Metrics" table shows the configuration of the Capacity Scheduler, including the scheduling resource type as [MEMORY] and minimum allocation as <memory:1024, vCores:1>. Below these tables is a detailed table of running applications, with one entry visible: "application_1609788795671_0023" with details: User: root, Name: streamjob3426571567828195357.jar, Application Type: MAPREDUCE, Queue: default, Start Time: Tue Jan 5 02:57:21 +0500, Finish Time: N/A, State: RUNNING, Final Status: UNDEFINED, Progress: 0%, Tracking UI: ApplicationMaster.

	Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers	Memory Used	Memory Total	Memory Reserved	Vcores Used	Vcores Total	Vcores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
22	0	1	21	2	6 GB	8 GB	0 B	2	8	0	1	0	0	0	0	0

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:8>

ID	User	Name	Application Type	Queue	Start Time	Finish Time	State	Final Status	Progress	Tracking UI
application_1609788795671_0023	root	streamjob3426571567828195357.jar	MAPREDUCE	default	Tue Jan 5 02:57:21 +0500	N/A	RUNNING	UNDEFINED	0%	ApplicationMaster

Application application_1609788795671_0023

Kill Application

User: root
Name: streamjob3426571567828195357.jar
Application Type: MAPREDUCE
Application Tags:
YarnApplicationState: FINISHED
FinalStatus Reported by AM: SUCCEEDED
Started: Mon Jan 04 16:57:21 -0500 2021
Elapsed: 2mins, 48sec
Tracking URL: History
Diagnostics:

Application Overview

Total Resource Preempted: <memory:0, vCores:0>
Total Number of Non-AM Container Preempted: 0
Total Number of AM Container Preempted: 0
Resource Preempted from Current Attempt: <memory:0, vCores:0>
Number of Non-AM Container Preempted from Current Attempt: 0
Aggregate Resource Allocation: 1010793 MB-seconds, 330 vcore-seconds

Application Metrics

Attempt ID	Started	Node	Logs
appattempt_1609788795671_0023_000001	N/A	N/A	N/A

Show 20 entries Search: First Previous 1 Next Last

Showing 1 to 1 of 1 entries

io. Highlight All Match Case Match Diacritics Whole Words 10 of 25 matches

3- Streaming Layer

Now starting the spark job with redis as streaming layer. The data will be read from mongodb and streams will be created in redis which will be used in spark for processing.

The dockerfile for spark is created as following:

```
tazeen@tazeen-HP-15-Notebook-PC:~$ sudo nano spark.Dockerfile
tazeen@tazeen-HP-15-Notebook-PC:~$
```

```
GNU nano 4.8
FROM python:3.6-slim-jessie

RUN apt-get update \
    && apt-get install -y locales \
    && dpkg-reconfigure -f noninteractive locales \
    && locale-gen C.UTF-8 \
    && /usr/sbin/update-locale LANG=C.UTF-8 \
    && echo "en_US.UTF-8 UTF-8" >> /etc/locale.gen \
    && locale-gen \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*

ENV LANG en_US.UTF-8
ENV LANGUAGE en_US:en
ENV LC_ALL en_US.UTF-8

RUN apt-get update \
    && apt-get install -y curl unzip \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*

ENV PYTHONHASHSEED 0
ENV PYTHONIOENCODING UTF-8
ENV PIP_DISABLE_PIP_VERSION_CHECK 1

# JAVA
ARG JAVA_MAJOR_VERSION=8
ARG JAVA_UPDATE_VERSION=131
ARG JAVA_BUILD_NUMBER=11
ENV JAVA_HOME /usr/jdk1.${JAVA_MAJOR_VERSION}.0_${JAVA_UPDATE_VERSION}

ENV PATH $PATH:$JAVA_HOME/bin
RUN curl -sL --retry 3 --insecure \
    --header "Cookie: oraclelicense=accept-securebackup-cookie;" \
```

Get Help F1 Get Help F1 Write Out F2 Where Is F3 Cut Text F4 Justify F5 Cur Pos F6 Undo F7 Mark Text F8 To Bracket F9 Exit F10 Read File F11 Replace F12 Paste Text F13 To Spell F14 Go To Line F15 Redo F16 Copy Text F17 Where Was F18

```
GNU nano 4.8
tazeen@tazeen-HP-15-Notebook-PC: ~
```

```
# JAVA
ARG JAVA_MAJOR_VERSION=8
ARG JAVA_UPDATE_VERSION=131
ARG JAVA_BUILD_NUMBER=11
ENV JAVA_HOME /usr/jdk1.${JAVA_MAJOR_VERSION}.0_${JAVA_UPDATE_VERSION}

# SPARK
ENV SPARK_VERSION 2.4.7
ENV SPARK_PACKAGE spark-${SPARK_VERSION}-bin-hadoop2.7
ENV SPARK_HOME /usr/spark
#ENV SPARK_DIST_CLASSPATH=$SHADOOP_HOME/etc/hadoop/*:$SHADOOP_HOME/share/hadoop/common/lib/*:$SHADOOP_HOME/share/hadoop/common/*:$SHADOOP_HOME/share/hadoop/tools/lib/*
ENV PATH $PATH:${SPARK_HOME}/bin
RUN curl -sL --retry 3 --insecure \
    --header "Cookie: oraclelicense=accept-securebackup-cookie;" \
    "http://download.oracle.com/otn-pub/java/jdk/${JAVA_MAJOR_VERSION}u${JAVA_UPDATE_VERSION}-b${JAVA_BUILD_NUMBER}/d54c1d3a095b4ff2b6607d096fae94b30/jdk-8u${JAVA_UPDATE_VERSION}-linux-x64.tar.gz" \
    | gunzip \
    | tar x -C /usr/ \
    && ln -s ${JAVA_HOME} /usr/java \
    && rm -rf ${JAVA_HOME}/man

WORKDIR $SPARK_HOME
CMD ["bin/spark-class", "org.apache.spark.deploy.master.Master"]
```

Updating the docker-compose.yml file for streaming layer. Spark master and worker docker are created as separate dockers and another docker is created for redis.

```
tazeen@tazeen-HP-15-Notebook-PC: ~$ sudo nano spark.Dockerfile
tazeen@tazeen-HP-15-Notebook-PC: ~$
```

```
GNU nano 4.8
tazeen@tazeen-HP-15-Notebook-PC: ~
```

```
version: "3.2"
services:
  hdfs:
    image: sequenceiq/hadoop-docker:2.7.0
    depends_on:
      - py-mongo
    networks:
      - default_bridge
    volumes:
      - ./mongo-app:/var/www/html
  py-mongo:
    # build the image from Dockerfile
    build:
      context: .
      volumes:
        - ./mongo-data:/data/db
        - ./mongo-app:/var/www/html
    ports:
      - "27017:27017"
    environment:
      - MONGO_INITDB_ROOT_USERNAME=root
      - MONGO_INITDB_ROOT_PASSWORD=1234
    networks:
      - default_bridge
    image: py_mongo
  spark-master:
    container_name: spark-master
    image: spark
    build:
      context: .
      dockerfile: spark.Dockerfile
    command: bin/spark-class org.apache.spark.deploy.master.Master -h spark-master
    hostname: spark-master
    environment:
```

Activities Terminal ٠٤:٢٠ جورى ١٥ tazeen@tazeen-HP-15-Notebook-PC: ~

```
GNU nano 4.8 docker-compose.yml
environment:
  MASTER: spark://spark-master:7077
  SPARK_CONF_DIR: /conf
  SPARK_PUBLIC_DNS: localhost
expose:
  - 7001
  - 7002
  - 7003
  - 7004
  - 7005
  - 7006
  - 7077
  - 6066
ports:
  - 4040:4040
  - 6066:6066
  - 7077:7077
  - 8080:8080
volumes:
  - ./services/spark/dependencies:/master/lib
  - ./services/spark/py-scripts:/master/scripts
networks:
  - default_bridge
spark-worker:
  image: spark
  container_name: spark-worker
  command: bin/spark-class org.apache.spark.deploy.worker.Worker spark://spark-master:7077
  hostname: spark-worker
  environment:
    SPARK_CONF_DIR: /conf
    SPARK_WORKER_CORES: 2
    SPARK_WORKER_MEMORY: 1g
    SPARK_WORKER_PORT: 8881
    SPARK_WORKER_WEBUI_PORT: 8081
```

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket
Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was

Activities Terminal ٠٤:٢٠ جورى ١٥ tazeen@tazeen-HP-15-Notebook-PC: ~

```
GNU nano 4.8 docker-compose.yml
environment:
  SPARK_CONF_DIR: /conf
  SPARK_WORKER_CORES: 2
  SPARK_WORKER_MEMORY: 1g
  SPARK_WORKER_PORT: 8881
  SPARK_WORKER_WEBUI_PORT: 8081
  SPARK_PUBLIC_DNS: localhost
expose:
  - 7012
  - 7013
  - 7014
  - 7015
  - 7016
  - 8881
ports:
  - 8081:8081
links:
  - spark-master
depends_on:
  - spark-master
networks:
  - default_bridge
redis:
  image: redis:latest
networks:
  - default_bridge
volumes:
  - ./mongo-app:/var/www/html
networks:
  default_bridge:
    name: tazeen_bridge
volumes:
  mongo-app:
```

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket
Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was

Navigate to localhost:8080 to see spark master is running.

Activities Firefox Web Browser 04:22 10 جوری

Spark Master at spark://spark-master:7077 - Mozilla Firefox

localhost:8080

Apache Spark 2.4.7

Spark Master at spark://spark-master:7077

URL: spark://spark-master:7077
Alive Workers: 1
Cores in use: 2 Total, 0 Used
Memory in use: 1024.0 MB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (1)

Worker Id	Address	State	Cores	Memory
worker-20210109205136-172.18.0.5-8881	172.18.0.5:8881	ALIVE	2 (0 Used)	1024.0 MB (0.0 B Used)

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Activities Firefox Web Browser 04:22 10 جوری

Spark Worker at 172.18.0.5:8881 - Mozilla Firefox

localhost:8081

Apache Spark 2.4.7

Spark Worker at 172.18.0.5:8881

ID: worker-20210109205136-172.18.0.5-8881
Master URL: spark://spark-master:7077
Cores: 2 (0 Used)
Memory: 1024.0 MB (0.0 B Used)

Back to Master

Running Executors (0)

ExecutorID	Cores	State	Memory	Job Details	Logs

To create streaming data, a text file is generated with redis streams from mongodb database. The data for November and December 2018. The mongodb script is saved as mongo-app/create-stream-data.py to generate data is below. I will only find the data for the flights which are not canceled, therefore, column 'CANCELLED' should be 0.

Activities Terminal ١٤:٣٣ جووى ١٠ tazeen@tazeen-HP-15-Notebook-PC: ~ mongo-app/create-stream-data.py Modified

```
GNU nano 4.8
from pymongo import MongoClient, errors

# global variables for MongoDB host (default port is 27017)
DOMAIN = '172.18.0.3'
PORT = 27017

# use a try-except indentation to catch MongoClient() errors
try:
    # try to instantiate a client instance
    client = MongoClient(
        host = [ str(DOMAIN) + ":" + str(PORT) ],
        serverSelectionTimeoutMS = 3000, # 3 second timeout
        username = "root",
        password = "1234",
    )
    db = client.airline;
    batchData = db.flights.find({'$and':[{"FL_DATE": {"$gte": '2018-12-01'}}], {"CANCELLED": {"$eq": '0.0'}}},
    {"FL_DATE": {$lte:'2018-12-31'}}, {"ARR_DELAY":1, "FL_DATE":1, '_id':0 });
    batchStr = str();
    for document in batchData:
        print (document);
        batchStr = batchStr + "XADD streams * FL_DATE " + document["FL_DATE"] + " ARR_DELAY " + document["ARR_DELAY"] + "\n";
        #batchJson.append(document);
    with open('stream-data1.txt', 'w') as outfile:
        outfile.write(batchStr)

except errors.ServerSelectionTimeoutError as err:
    # catch pymongo.errors.ServerSelectionTimeoutError
    print ("pymongo ERROR:", err)
```

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket
Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was

Activities Terminal ١٥:٣٣ جووى ١٠ tazeen@tazeen-HP-15-Notebook-PC: ~ mongo-app/create-stream-data.py Modified

```
GNU nano 4.8
from pymongo import MongoClient, errors

# global variables for MongoDB host (default port is 27017)
DOMAIN = '172.18.0.3'
PORT = 27017

# use a try-except indentation to catch MongoClient() errors
try:
    # try to instantiate a client instance
    client = MongoClient(
        host = [ str(DOMAIN) + ":" + str(PORT) ],
        serverSelectionTimeoutMS = 3000, # 3 second timeout
        username = "root",
        password = "1234",
    )
    db = client.airline;
    batchData = db.flights.find({'$and':[ {"$gte": '2018-11-01'}}], {"CANCELLED": {"$eq": '0.0'}}},
    {"FL_DATE": {$lte:'2018-11-30'}}, {"ARR_DELAY":1, "FL_DATE":1, '_id':0 });
    batchStr = str();
    for document in batchData:
        print (document);
        batchStr = batchStr + "XADD streams * FL_DATE " + document["FL_DATE"] + " ARR_DELAY " + document["ARR_DELAY"] + "\n";
        #batchJson.append(document);
    with open('stream-data2.txt', 'w') as outfile:
        outfile.write(batchStr)

except errors.ServerSelectionTimeoutError as err:
    # catch pymongo.errors.ServerSelectionTimeoutError
    print ("pymongo ERROR:", err)
```

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket
Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was

Now executing the script in mongo db docker container. It will create two stream data files as tream-data1.txt and stream-data2.txt. These files will be used by redis docker container to generate streams.

```

root@f818c348f361:/var/www/html
tazeen@tazeen-HP-15-Notebook-PC:~$ sudo nano mongo-app/create-stream-data.py
[tudo] password for tazeen:
tazeen@tazeen-HP-15-Notebook-PC:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
914ac8b3ff5d      redis:latest        "docker-entrypoint.s..."   23 hours ago       Up 2 hours         tazeen_redis_1
e7f3401af964      spark               "bin/spark-class org..."  27 hours ago       Up 2 hours         tazeen_spark-worker
2911db20f56       spark               "bin/spark-class org..."  27 hours ago       Up 2 hours         tazeen_spark-master
340c4Bbb6220      sequenceiq/hadoop-docker:2.7.0    "/etc/bootstrap.sh -d"   2 days ago        Up 2 hours         tazeen_hdfs_1
f818c348f361      py_mongo            "docker-entrypoint.s..."   2 days ago        Up 2 hours         tazeen_py-mongo_1
tazeen@tazeen-HP-15-Notebook-PC:~$ docker exec -it f818c348f361 /bin/bash
root@f818c348f361:/# cd /var/www/html
root@f818c348f361:/var/www/html# python3 create-stream-data.py

```

Let's execute the script to execute streaming data from redis to spark master. The dependencies required for spark is mounted on /master/lib folder in spark master. These dependencies consists of jar files required to read redis streams in pyspark.

Copying jar files to ./services/spark/dependencies folder which is mounted as /master/lib in spark master.

```

Activities Terminal 04:38 10 جنوری •
tazeen@tazeen-HP-15-Notebook-PC:~/services/spark/dependencies
tazeen@tazeen-HP-15-Notebook-PC:~/services/spark/dependencies$ cp /home/tazeen/spark-
spark-base.Dockerfile    spark-redis-2.4.0.jar    spark-stream.py
spark-master.Dockerfile  spark-redis-dependencies/ spark-worker.Dockerfile
tazeen@tazeen-HP-15-Notebook-PC:~/services/spark/dependencies$ cp /home/tazeen/spark-redis-dependencies/spark-redis-2.4.1.jar
cp: missing destination file operand after '/home/tazeen/spark-redis-dependencies/spark-redis-2.4.1.jar'
Try 'cp --help' for more information.
tazeen@tazeen-HP-15-Notebook-PC:~/services/spark/dependencies$ cp /home/tazeen/spark-redis-dependencies/spark-redis-2.4.1.jar .
cp: cannot create regular file './spark-redis-2.4.1.jar': Permission denied
tazeen@tazeen-HP-15-Notebook-PC:~/services/spark/dependencies$ sudo cp /home/tazeen/spark-redis-dependencies/spark-redis-2.4.1.jar .
tazeen@tazeen-HP-15-Notebook-PC:~/services/spark/dependencies$ sudo cp /home/tazeen/spark-redis-dependencies/jedis-3.2.0.jar .
tazeen@tazeen-HP-15-Notebook-PC:~/services/spark/dependencies$ sudo cp /home/tazeen/spark-redis-dependencies/commons-pool2-2.0.jar .
tazeen@tazeen-HP-15-Notebook-PC:~/services/spark/dependencies$ ls
commons-pool2-2.0.jar  jedis-3.2.0.jar  spark-redis-2.4.1.jar
tazeen@tazeen-HP-15-Notebook-PC:~/services/spark/dependencies$ 

```

The python script to execute streaming reads streaming data from redis host, it's host ip can be seen from bridge network. For redis-docker ip is : 172.18.0.4

```
Activities Terminal ٠٤:٤٦ ١٥ جنوری tazeen@tazeen-HP-15-Notebook-PC:~$ docker network inspect tazeen_bridge
[
  {
    "Name": "tazeen_bridge",
    "Id": "dedd0bcc7794a920abff474b5a60420925dc41a4b1009e8abf56c9980ecbf727",
    "Created": "2021-01-04T16:03:48.017315154+05:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": true,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "2911db20f56af3aec48fd0c310390d97332a13348d260f45e083da0ce473ec": {
        "Name": "spark-master",
        "EndpointID": "679a0bdccc207e58a33d3b2abf0d2485cbea955a33293edf27061f6f1699f03f",
        "MacAddress": "02:42:ac:12:00:02",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
      },
      "340c48bb8220ec3fddf0a0c6449105aafcc92f9baf9957ca6792752a60bb979e": {
        "Name": "tazeen_hdfs_1",
        "EndpointID": "60c7c504525b53a272541979538bcb17cb95bde0eb0ef382789f9153bd007e5a",
        "MacAddress": "02:42:ac:12:00:06",
        "IPv4Address": "172.18.0.2/16"
      }
    }
  }
]
```

```
Activities Terminal ٠٤:٤٦ ١٥ جنوری tazeen@tazeen-HP-15-Notebook-PC:~$ docker network inspect tazeen_bridge
[
  {
    "Name": "tazeen_hdfs_1",
    "EndpointID": "60c7c504525b53a272541979538bcb17cb95bde0eb0ef382789f9153bd007e5a",
    "MacAddress": "02:42:ac:12:00:06",
    "IPv4Address": "172.18.0.6/16",
    "IPv6Address": ""
  },
  {
    "Name": "tazeen_redis_1",
    "EndpointID": "b5c97a7b4d48b4db722aa9e2c0e645ec4c3036baa8f231cac8c93058042af433",
    "MacAddress": "02:42:ac:12:00:04",
    "IPv4Address": "172.18.0.4/16",
    "IPv6Address": ""
  },
  {
    "Name": "spark-worker",
    "EndpointID": "80b071c8c3a69b0610a9932ce6cf5e0acb4c77aeb5a18d28f0700f8eb4c63169",
    "MacAddress": "02:42:ac:12:00:05",
    "IPv4Address": "172.18.0.5/16",
    "IPv6Address": ""
  },
  {
    "Name": "tazeen_py_mongo_1",
    "EndpointID": "15ae562a43de1c70253a0c31f5a480d83b6497abf68d1442617cb96abcae5ec7",
    "MacAddress": "02:42:ac:12:00:03",
    "IPv4Address": "172.18.0.3/16",
    "IPv6Address": ""
  }
]
{
  "Options": {},
  "Labels": {
    "com.docker.compose.network": "tazeen_bridge",
    "com.docker.compose.project": "tazeen",
    "com.docker.compose.version": "1.27.4"
  }
}
]
```

The python script is summing the delay time for positive flight delays, the number of rows provided in each stream and the month of flight date. The script is following:

```
Activities Terminal ١٧:٥٢ جورى ١٠ tazeen@tazeen-HP-15-Notebook-PC: ~ Modified
GNU nano 4.8 services/spark/py-scripts/spark-stream.py
from pyspark.sql import SparkSession, SQLContext, DataFrame
from pyspark.sql.types import *
from pyspark.sql.functions import col, sum as _sum, first, concat_ws, split, count as _count

spark = SparkSession \
    .builder \
    .master("local[*]") \
    .config("spark.redis.host", "172.18.0.4") \
    .config("spark.redis.port", "6379") \
    .getOrCreate()

sensors = spark \
    .readStream \
    .format("redis") \
    .option("stream.keys", "streams") \
    .schema(StructType([
        StructField("FL_DATE", StringType()), \
        StructField("ARR_DELAY", FloatType()) \
    ])) \
    .load()
def process_row(row, id):
    # Process row
    df_filtered = row.filter(col("ARR_DELAY") >=0)
    df_stats = df_filtered.select(_sum(col("ARR_DELAY")).alias('sum'), _count(col("ARR_DELAY")).alias('count'), \
concat_ws('-', split(first(col("FL_DATE")), '[-]')[1], \
split(first(col("FL_DATE")), '[-]')[0]).alias('mon-year')).collect()
    print (df_stats)
month = df_stats[0]['mon-year']
#store results in redis table
sc = spark.sparkContext
myJson = sc.parallelize([{"sum_of_delay":df_stats[0]['sum'], "count": df_stats[0]['count'], "month": month}])
myDf = spark.read.json(myJson).write.format("org.apache.spark.sql.redis") \
    .option("table", "avgDelay") \
    .mode("append") \
    .save()

FG Get Help PO Write Out NW Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo M-A Mark Text M-] To Bracket
FX Exit PR Read File ^R Replace ^U Paste Text ^T To Spell ^G Go To Line M-E Redo M-6 Copy Text M-Q Where Was
```

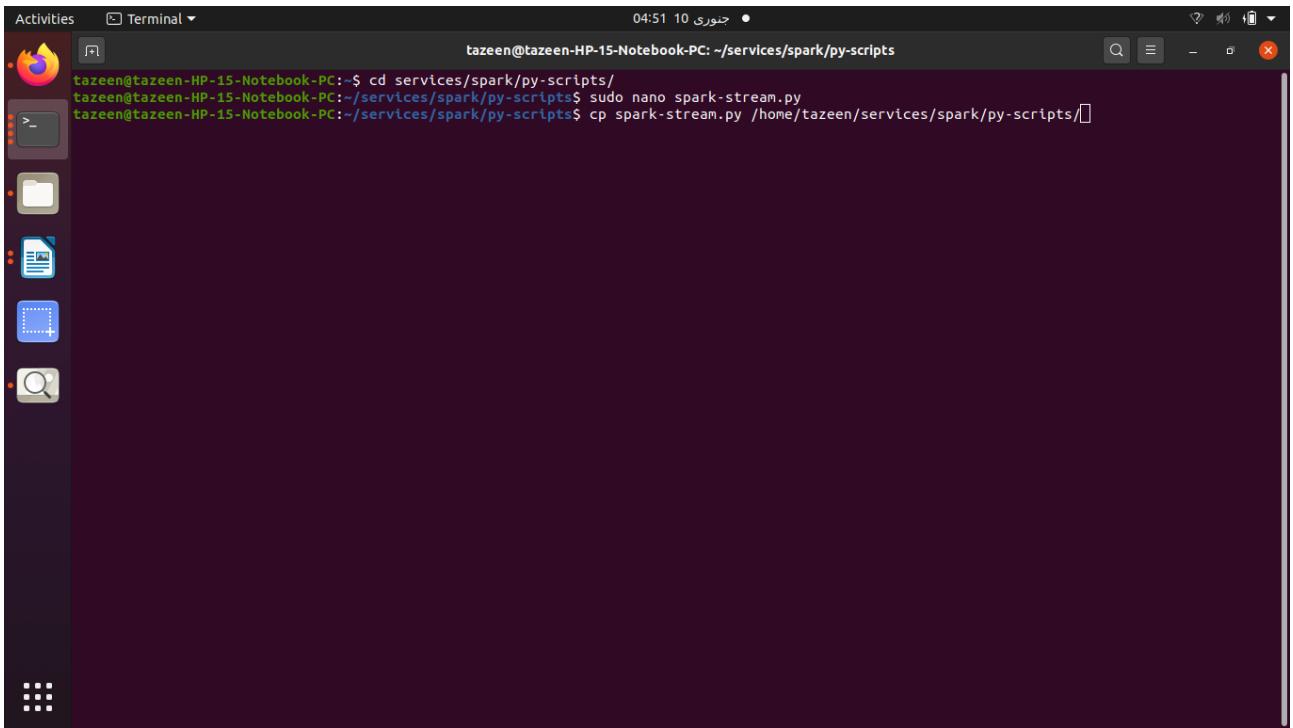
```
Activities Terminal ١٧:٥٢ جورى ١٠ tazeen@tazeen-HP-15-Notebook-PC: ~ Modified
GNU nano 4.8 services/spark/py-scripts/spark-stream.py
.option("stream.keys", "streams") \
.schema(StructType([
    StructField("FL_DATE", StringType()), \
    StructField("ARR_DELAY", FloatType()) \
])) \
.load()
def process_row(row, id):
    # Process row
    df_filtered = row.filter(col("ARR_DELAY") >=0)
    df_stats = df_filtered.select(_sum(col("ARR_DELAY")).alias('sum'), _count(col("ARR_DELAY")).alias('count'), \
concat_ws('-', split(first(col("FL_DATE")), '[-]')[1], \
split(first(col("FL_DATE")), '[-]')[0]).alias('mon-year')).collect()
    print (df_stats)
month = df_stats[0]['mon-year']
#store results in redis table
sc = spark.sparkContext
myJson = sc.parallelize([{"sum_of_delay":df_stats[0]['sum'], "count": df_stats[0]['count'], "month": month}])
myDf = spark.read.json(myJson).write.format("org.apache.spark.sql.redis") \
    .option("table", "avgDelay") \
    .mode("append") \
    .save()

pass
query = sensors \
    .writeStream \
    .outputMode("update") \
    .foreachBatch(process_row) \
    .start()

try:
    query.awaitTermination()
except Exception as error:
    print ('Streaming query exception', error)
[]

FG Get Help PO Write Out NW Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo M-A Mark Text M-] To Bracket
FX Exit PR Read File ^R Replace ^U Paste Text ^T To Spell ^G Go To Line M-E Redo M-6 Copy Text M-Q Where Was
```

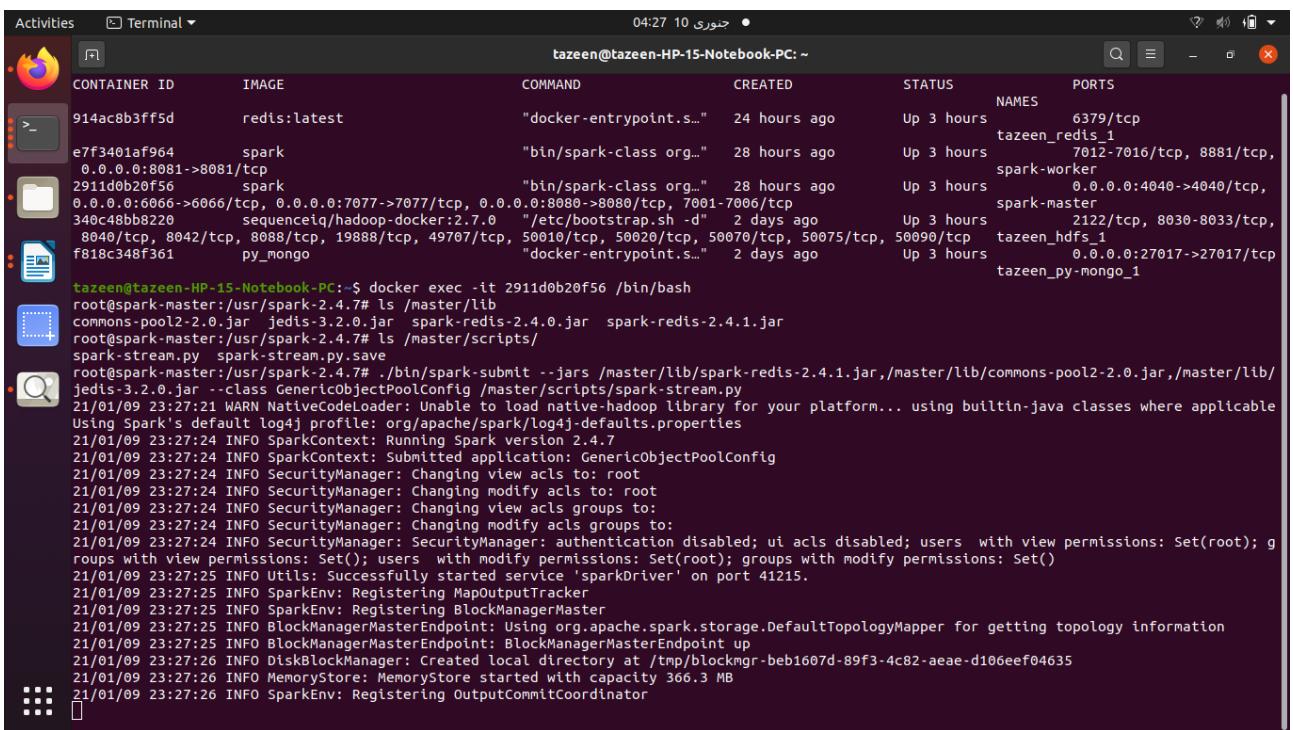
Now copying the script in ./services/spark/py-scripts/ folders which is mounted on /master/scripts folder in spark master.



A screenshot of an Ubuntu desktop environment. A terminal window is open in the center, titled 'Terminal'. The command line shows the user navigating to the directory '/services/spark/py-scripts' and then executing three commands: 'sudo nano spark-stream.py', 'cp spark-stream.py /home/tazeen/services/spark/py-scripts/'. The terminal window has a dark purple background. On the left side of the screen, there is a vertical dock with icons for various applications like a browser, file manager, and search.

```
tazeen@tazeen-HP-15-Notebook-PC:~$ cd services/spark/py-scripts/
tazeen@tazeen-HP-15-Notebook-PC:~/services/spark/py-scripts$ sudo nano spark-stream.py
tazeen@tazeen-HP-15-Notebook-PC:~/services/spark/py-scripts$ cp spark-stream.py /home/tazeen/services/spark/py-scripts/
```

Connecting to the spark master terminal and executing the script for streaming.



A screenshot of an Ubuntu desktop environment. A terminal window is open in the center, titled 'Terminal'. The command line shows the user listing Docker containers with 'docker ps' and then executing 'docker exec -it 2911d0b20f56 /bin/bash' to enter the spark master container. Inside the container, the user runs 'ls /master/lib' and then executes the 'spark-stream.py' script. The terminal window has a dark purple background. On the left side of the screen, there is a vertical dock with icons for various applications like a browser, file manager, and search.

```
tazeen@tazeen-HP-15-Notebook-PC:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
914ac8b3ff5d        redis:latest        "docker-entrypoint.s..."   24 hours ago       Up 3 hours          6379/tcp
e7f3401af964        spark              "bin/spark-class org..."  28 hours ago       Up 3 hours          7012-7016/tcp, 8881/tcp,
0.0.0.0:8081->8081/tcp
2911d0b20f56        spark              "bin/spark-class org..."  28 hours ago       Up 3 hours          0.0.0.0:4040->4040/tcp,
0.0.0.0:6066->6066/tcp, 0.0.0.0:7077->7077/tcp, 0.0.0.0:8080->8080/tcp, 7001-7006/tcp
340c49bb8220        sequenceiq/hadoop-docker:2.7.0  "/etc/bootstrap.sh -d"   2 days ago        Up 3 hours          2122/tcp, 8030-8033/tcp,
8040/tcp, 8042/tcp, 8088/tcp, 19888/tcp, 49707/tcp, 50010/tcp, 50020/tcp, 50075/tcp, 50090/tcp
f818c348f361        py_mongo           "docker-entrypoint.s..."   2 days ago        Up 3 hours          0.0.0.0:27017->27017/tcp
tazeen@tazeen-HP-15-Notebook-PC:~$ docker exec -it 2911d0b20f56 /bin/bash
root@spark-master:/usr/spark-2.4.7# ls /master/lib
commons-pool2-2.0.jar  jedis-3.2.0.jar  spark-redis-2.4.1.jar
root@spark-master:/usr/spark-2.4.7# ls /master/scripts/
spark-stream.py  spark-stream.py.save
root@spark-master:/usr/spark-2.4.7# ./bin/spark-submit --jars /master/lib/spark-redis-2.4.1.jar,/master/lib/commons-pool2-2.0.jar,/master/lib/jedis-3.2.0.jar -class GenericObjectPoolConfig /master/scripts/spark-stream.py
21/01/09 23:27:21 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
21/01/09 23:27:24 INFO SparkContext: Running Spark version 2.4.7
21/01/09 23:27:24 INFO SparkContext: Submitted application: GenericObjectPoolConfig
21/01/09 23:27:24 INFO SecurityManager: Changing view acls to: root
21/01/09 23:27:24 INFO SecurityManager: Changing modify acls to: root
21/01/09 23:27:24 INFO SecurityManager: Changing view acls groups to:
21/01/09 23:27:24 INFO SecurityManager: Changing modify acls groups to:
21/01/09 23:27:24 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(root); groups with view permissions: Set(); users with modify permissions: Set(root); groups with modify permissions: Set()
21/01/09 23:27:25 INFO Utils: Successfully started service 'sparkDriver' on port 41215.
21/01/09 23:27:25 INFO SparkEnv: Registering MapOutputTracker
21/01/09 23:27:25 INFO SparkEnv: Registering BlockManagerMaster
21/01/09 23:27:25 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
21/01/09 23:27:25 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
21/01/09 23:27:26 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-beb1607d-89f3-4c82-aeae-d106ee04635
21/01/09 23:27:26 INFO MemoryStore: MemoryStore started with capacity 366.3 MB
21/01/09 23:27:26 INFO SparkEnv: Registering OutputCommitCoordinator
```

Now connecting to the redis docker container to send streams to spark. It consists of stream-data1.txt and stream-data2.txt as the redis streams for November 2018 and December 2018. The data is mounted as /var/www/html in redis docker.

```
tazeen@tazeen-HP-15-Notebook-PC:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
914ac8b3ff5d      redis:latest        "docker-entrypoint.s..."   37 hours ago     Up 5 hours          NAMES
                                                               6379/tcp
e7f3401af964      spark              "bin/spark-class org..."  41 hours ago     Up 5 hours          tazeen_redis_1
                                                               7012-7016/tcp, 8881/tcp,
2911db20f56       spark              "bin/spark-class org..."  41 hours ago     Up 5 hours          spark-worker
                                                               0.0.0.0:4040->4040/tcp,
0.0.0.0:6066->6066/tcp, 0.0.0.0:7077->7077/tcp, 0.0.0.0:8080->8080/tcp, 7001-7006/tcp
340c49bb8220      sequenceiq/hadoop-docker:2.7.0  "/etc/bootstrap.sh -d"  2 days ago       Up 5 hours          spark-master
                                                               2122/tcp, 8030-8033/tcp,
8040/tcp, 8042/tcp, 8088/tcp, 19888/tcp, 49707/tcp, 50010/tcp, 50020/tcp, 50070/tcp, 50075/tcp, 50090/tcp
f818c348f361      py_mongo           "docker-entrypoint.s..."  2 days ago       Up 5 hours          tazeen_hdbs_1
                                                               0.0.0.0:27017->27017/tcp
                                                               0.0.0.0:27017->27017/tcp
tazeen@tazeen-HP-15-Notebook-PC:~$ docker exec -it 914ac8b3ff5d /bin/bash
root@914ac8b3ff5d:/data# ls /var/www/html/
batch-data.py  batch-data.txt  batch-data1.txt  create-stream-data.py  output.txt  stream-data.txt  stream-data1.txt  stream-data2.txt
root@914ac8b3ff5d:/data# redis-cli < /var/www/html/stream-data1.txt
```

```
"1610283002444-1"
"1610283002444-2"
"1610283002444-3"
"1610283002444-4"
"1610283002444-5"
"1610283002444-6"
"1610283002444-7"
"1610283002445-0"
"1610283002445-1"
"1610283002445-2"
"1610283002445-3"
"1610283002445-4"
"1610283002445-5"
"1610283002445-6"
"1610283002445-7"
"1610283002445-8"
"1610283002446-0"
"1610283002446-1"
"1610283002446-2"
"1610283002446-3"
"1610283002446-4"
"1610283002446-5"
"1610283002446-6"
"1610283002446-7"
"1610283002446-8"
"1610283002447-0"
"1610283002447-1"
"1610283002447-2"
"1610283002447-3"
"1610283002447-4"
"1610283002447-5"
"1610283002447-6"
"1610283002447-7"
"1610283002447-8"
"1610283002448-0"
"1610283002448-1"
"1610283002448-2"
root@914ac8b3ff5d:/data# redis-cli < /var/www/html/stream-data2.txt
```

Using the cli to check the putput saved in redis table. Keys * lists all the keys created in redis. All keys starting with aygAverage are created by streaming. So saving all these results in stream-output.txt file. The output file will be saved in mongo-app folder in disk as /var/www/html is mounted for it.

Activities Terminal ١٧:٥٣ جورى ١٠ tazeen@tazeen-HP-15-Notebook-PC: ~

```
root@914ac8b3ff5d:/data# redis-cli
127.0.0.1:6379> keys *
 1) "avgDelay:32a14818738d4c30abdeb29ecf0126e9"
 2) "avgDelay:552f8584eb6d4e97a7e5b0723fe11b97"
 3) "avgDelay:13ab1e135bb849ef990250ad2051af"
 4) "avgDelay:cf396dcbb4a46f2aaaadea80ebcc223"
 5) "avgDelay:071f85bef03c4bea9719e9e27014bb87"
 6) "avgDelay:b7ff7e927a27d42ebbe39338e35bef9a"
 7) "avgDelay:d05f504f2cf941debe2643cedad008b"
 8) "avgDelay:976f3e87cf26483baa9f64800d22a3b"
 9) "avgDelay:dcaf63cf593f4a53897369d76183891f"
10) "avgDelay:0a34323b0fc44cce8502c6a652daf5fe"
11) "avgDelay:50507c56e054c2aa671af610bbebed0"
12) "avgDelay:326279eda5dd43cd83d9861cb81dd80"
13) "avgDelay:202c09196dd479db39724c30f2722cf"
14) "avgDelay:b9c4378e36064ce698e3e9b93aab6380"
15) "avgDelay:66e66ae9681f4ff959fb9eae4055e9a"
16) "avgDelay:6d7ff531755c4fe687bba189c633e9f9"
17) "avgDelay:4916537331e14087b4c663e77842fd7f"
18) "avgDelay:669db4dbc16649d69db692eda94258f"
19) "avgDelay:e2ae2866cd274fc797377d858aa66621"
20) "avgDelay:bab7291faf0441e093da598f06a7a163"
21) "avgDelay:2c1462bd1aa346ff970dd4e1d658f5cb"
22) "avgDelay:2dad59d09814996986a34c366127b46"
23) "avgDelay:048c5164ba804f24bebdfdd4e3df58be"
24) "avgDelay:70770a65120e468e8e25cf94b4350b80"
25) "avgDelay:7d3e6f9780564a97be7ff4b01083055"
26) "avgDelay:f259cd4385ee642d28ceb4321e366d883"
27) "avgDelay:esb44f9fa5da4ed1a1a35252b56efbe"
28) "avgDelay:32a07d301d784b67bde567d7dd1c91"
29) "avgDelay:0537eab651e4e6c80f8e2f628701317"
30) "avgDelay:72aab758d3545eabaedf32131c500489"
31) "avgDelay:8040a55e584947a9bda85930ec86b5c"
32) "avgDelay:3bf82bacab4a4671af95b733c80fab1b"
33) "avgDelay:5dd093d1babd49338a21023f114e1da"
34) "avgDelay:8764dcca1684031b1f0ed87bb58bcd0"
35) "avgDelay:ad805a71634342af5cb21872aff34b"
36) "avgDelay:cbc4df722ff64965a0ad5baa8f8e094d"
37) "avgDelay:0...0722ff600a45b4a0e85e8e8b2200e"
```

Activities Terminal ١٧:٥٣ جورى ١٠ tazeen@tazeen-HP-15-Notebook-PC: ~

```
root@914ac8b3ff5d:/data# redis-cli --raw keys avgDelay* | awk '{printf "hgetall %s\n", $1}' | redis-cli --raw > /var/www/html/stream-output.txt
t
root@914ac8b3ff5d:/data#
```

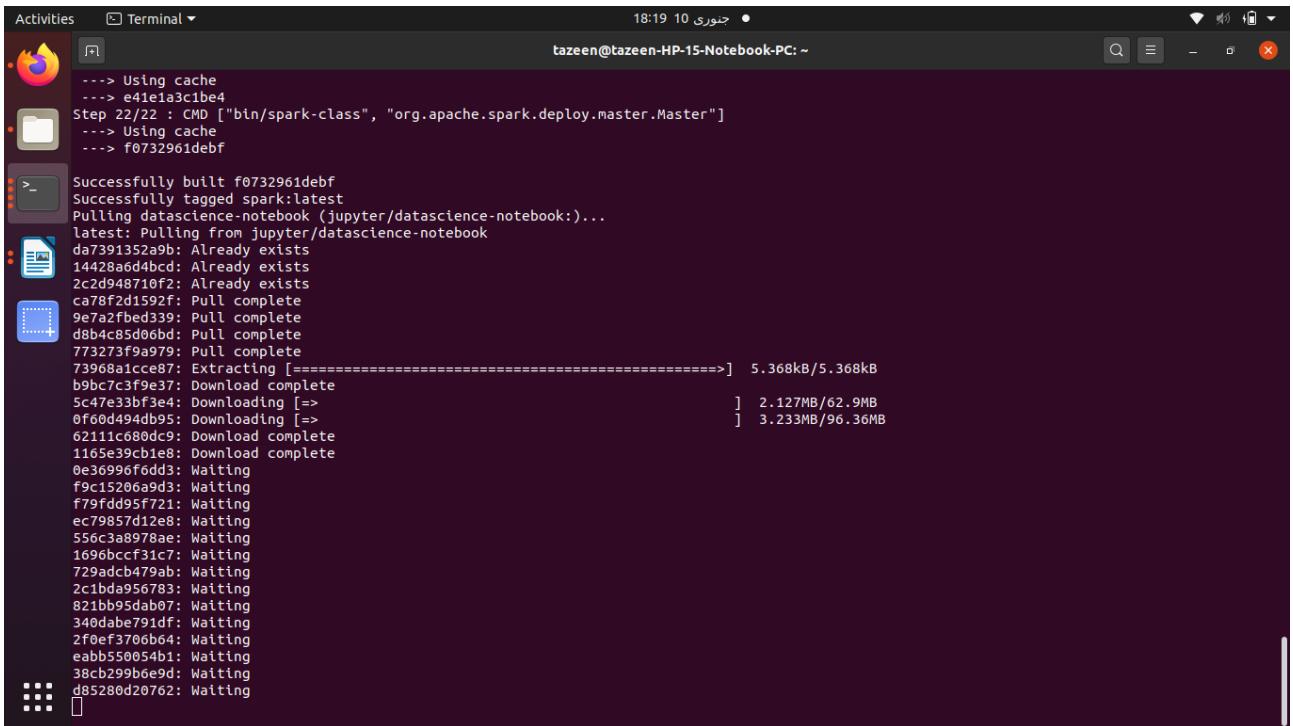
```
1 count
2 1382
3 month
4 11-2018
5 sum_of_delay
6 40290.0
7 month
8 11-2018
9 count
10 805
11 sum_of_delay
12 25499.0
13 sum_of_delay
14 68845.0
15 count
16 1679
17 month
18 12-2018
19 month
20 11-2018
21 sum_of_delay
22 26080.0
23 count
24 1051
25 count
26 723
27 month
28 11-2018
29 sum_of_delay
30 21854.0
31 sum_of_delay
32 19908.0
33 count
34 976
35 month
36 11-2018
37 count
```

4. Serving Layer

The last step is to create the serving layer which will show the results from batch and stream layers. To create a serving layer, jupyter notebook image is used in container. The updated docker-compose.yml is adds the jupyter image as below:

```
version: '3'
services:
  mongo-app:
    image: mongo:latest
    volumes:
      - ./mongo-app:/var/www/html
  redis:
    image: redis:latest
  spark-master:
    image: spark:master
    ports:
      - 8081:8081
    links:
      - spark-master
    depends_on:
      - spark-master
  datascience-notebook:
    image: jupyter/datascience-notebook
    volumes:
      - /mongo-app:/home/tazeen/jup-notebook
    ports:
      - 8888:8888
    container_name: datascience-notebook-container
    networks:
      - default_bridge
  networks:
    default_bridge:
      name: tazeen_bridge
    volumes:
      mongo-app:
```

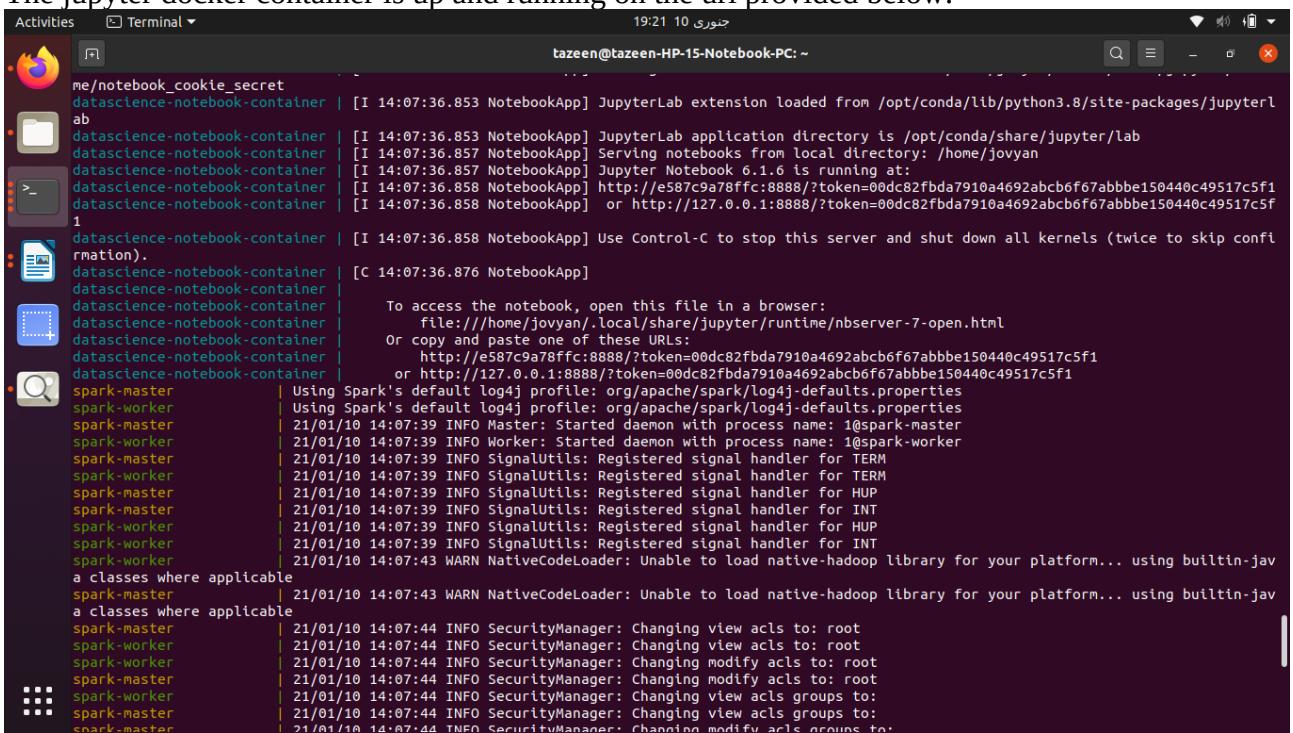
Now executing docker-compose –build



```
Activities Terminal ١٨:١٩ ١٠ جووى tazeen@tazeen-HP-15-Notebook-PC: ~
.
--> Using cache
--> e4ie1a3c1be4
Step 22/22 : CMD ["bin/spark-class", "org.apache.spark.deploy.master.Master"]
--> Using cache
--> f0732961deb

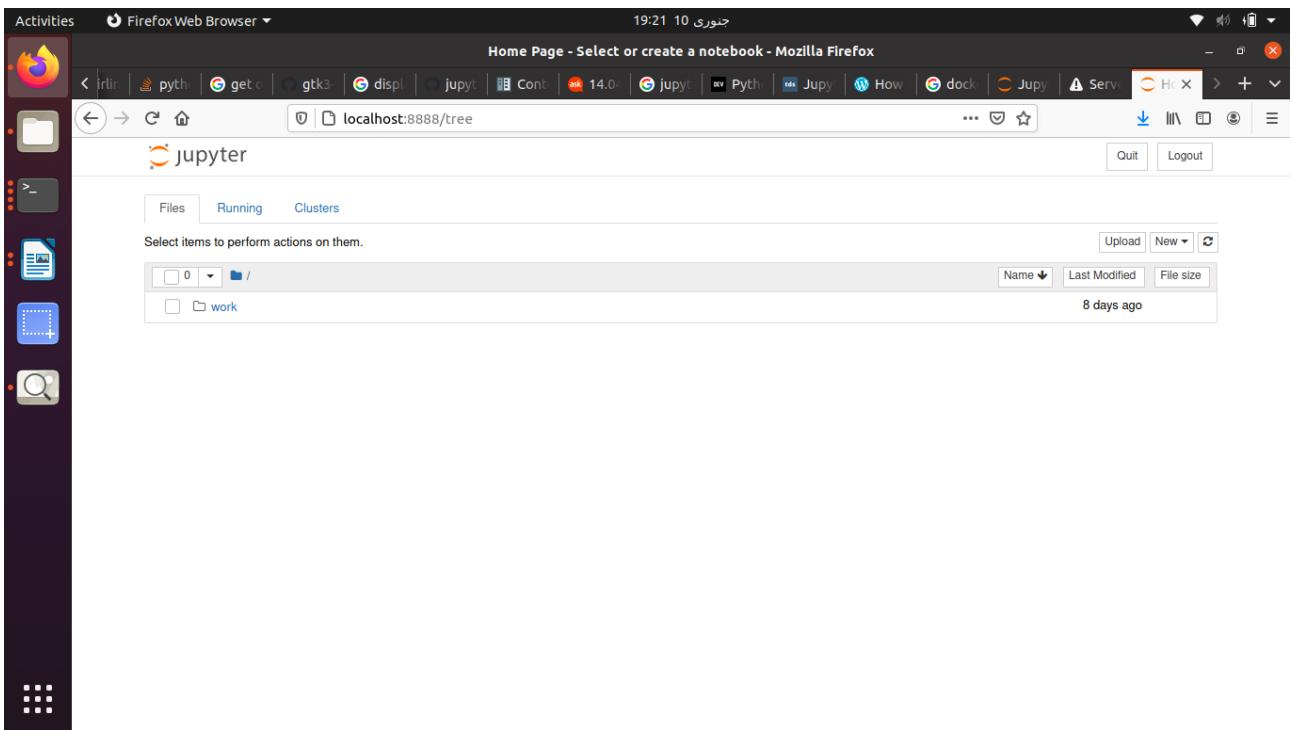
Successfully built f0732961deb
Successfully tagged spark:latest
Pulling datascience-notebook (jupyter/datascience-notebook):...
latest: Pulling from jupyter/datascience-notebook
da7391352a9b: Already exists
14428a6d4bcd: Already exists
2c2d948710f2: Already exists
ca7bf2d1592f: Pull complete
9e7a2fbed339: Pull complete
d8b4c85d66bd: Pull complete
773273f9a979: Pull complete
73968a1cce87: Extracting [=====] 5.368kB/5.368kB
b9bc7c3f9e37: Download complete
5c47e33bf3e4: Downloading [>] 2.127MB/62.9MB
0f66d494db95: Downloading [>] 3.233MB/96.36MB
02111c680dc9: Download complete
1165e39cb1e8: Download complete
0e36996fddd3: Waiting
f9c15206a9d3: Waiting
f79fdd95f721: Waiting
ec79857d12e8: Waiting
556c3a8978ae: Waiting
1696bccf31c7: Waiting
729adcb479ab: Waiting
2c1bda956783: Waiting
821bb95dab07: Waiting
340dabe791df: Waiting
2f0ef3706b64: Waiting
eabb550054b1: Waiting
38cb299bbe9d: Waiting
d85280d20762: Waiting
```

The jupyter docker container is up and running on the url provided below:



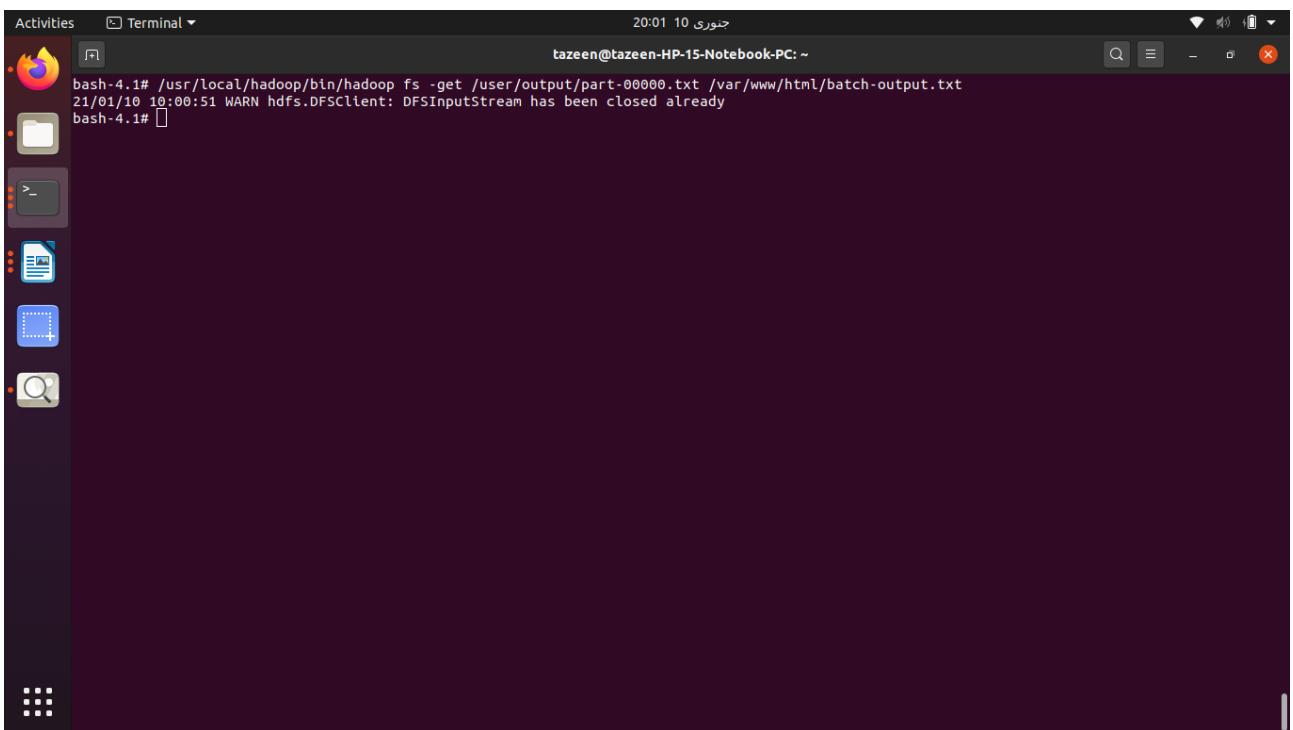
```
Activities Terminal ١٩:٢١ ١٠ جووى tazeen@tazeen-HP-15-Notebook-PC: ~
.
me/notebook_cookie_secret
datascience-notebook-container | [I 14:07:36.853 NotebookApp] JupyterLab extension loaded from /opt/conda/lib/python3.8/site-packages/jupyterlab
ab
datascience-notebook-container | [I 14:07:36.853 NotebookApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
datascience-notebook-container | [I 14:07:36.857 NotebookApp] Serving notebooks from local directory: /home/jovyan
datascience-notebook-container | [I 14:07:36.857 NotebookApp] Jupyter Notebook 6.1.6 is running at:
datascience-notebook-container | [I 14:07:36.858 NotebookApp] http://e587c9a78ffc:8888/?token=00dc82fbda7910a4692abcb6f67abbbe150440c49517c5f1
datascience-notebook-container | [I 14:07:36.858 NotebookApp] or http://127.0.0.1:8888/?token=00dc82fbda7910a4692abcb6f67abbbe150440c49517c5f1
1
datascience-notebook-container | [I 14:07:36.858 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
datascience-notebook-container | [C 14:07:36.876 NotebookApp]
datascience-notebook-container | To access the notebook, open this file in a browser:
datascience-notebook-container | file:///home/jovyan/.local/share/jupyter/runtime/nbserver-7-open.html
datascience-notebook-container | Or copy and paste one of these URLs:
datascience-notebook-container | http://e587c9a78ffc:8888/?token=00dc82fbda7910a4692abcb6f67abbbe150440c49517c5f1
datascience-notebook-container | or http://127.0.0.1:8888/?token=00dc82fbda7910a4692abcb6f67abbbe150440c49517c5f1
spark-master | [I 21/01/10 14:07:39 INFO Master: Started daemon with process name: 1@spark-master
spark-worker | [I 21/01/10 14:07:39 INFO Worker: Started daemon with process name: 1@spark-worker
spark-master | [I 21/01/10 14:07:39 INFO SignalUtils: Registered signal handler for TERM
spark-worker | [I 21/01/10 14:07:39 INFO SignalUtils: Registered signal handler for TERM
spark-master | [I 21/01/10 14:07:39 INFO SignalUtils: Registered signal handler for HUP
spark-worker | [I 21/01/10 14:07:39 INFO SignalUtils: Registered signal handler for HUP
spark-worker | [I 21/01/10 14:07:39 INFO SignalUtils: Registered signal handler for INT
spark-worker | [I 21/01/10 14:07:39 INFO SignalUtils: Registered signal handler for INT
spark-worker | [I 21/01/10 14:07:43 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
a classes where applicable
spark-master | [I 21/01/10 14:07:43 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
a classes where applicable
spark-master | [I 21/01/10 14:07:44 INFO SecurityManager: Changing view acls to: root
spark-worker | [I 21/01/10 14:07:44 INFO SecurityManager: Changing view acls to: root
spark-worker | [I 21/01/10 14:07:44 INFO SecurityManager: Changing modify acls to: root
spark-master | [I 21/01/10 14:07:44 INFO SecurityManager: Changing modify acls to: root
spark-worker | [I 21/01/10 14:07:44 INFO SecurityManager: Changing view acls groups to:
spark-master | [I 21/01/10 14:07:44 INFO SecurityManager: Changing view acls groups to:
spark-master | [I 21/01/10 14:07:44 INFO SecurityManager: Changing modify acls groups to:
```

Navigating to the jupyter notebook on the url provided.

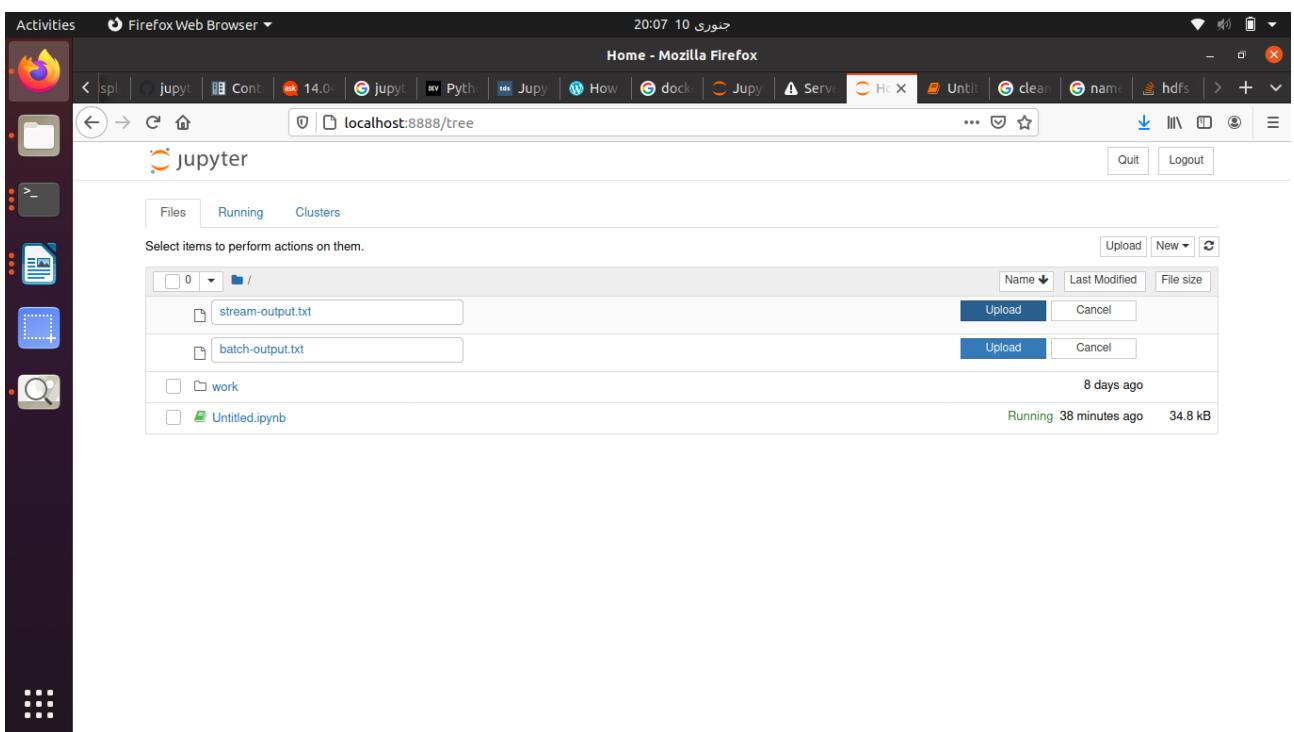
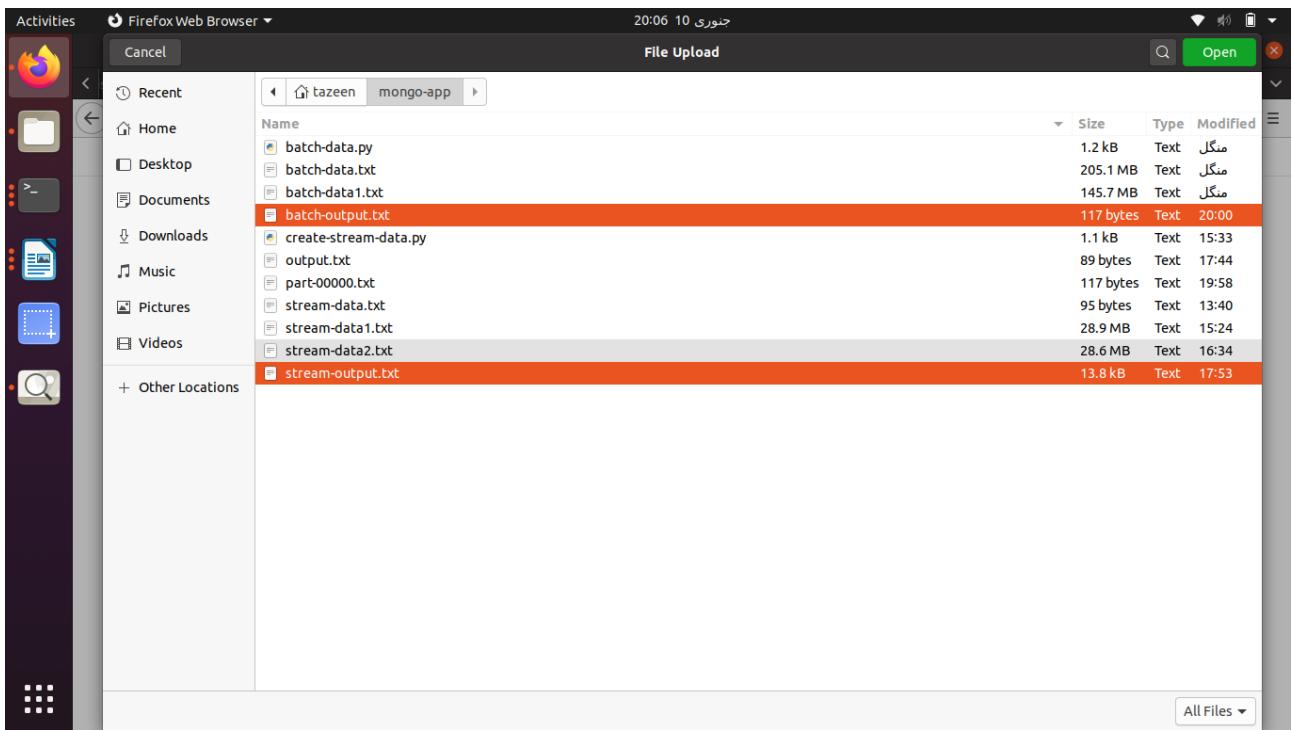


Now uploading output files in jupyter. These files are save in mongo-app folder which is mounted in the jupyter container.

First saving HDFS batch output file in mongo-app folder by:



The stream-output file is already saved in mongo-app folder as previously shown. Now uploading these files in jupyter notebook.



Finally creating new notebook to display results from the batch and stream layers.

Activities Firefox Web Browser ٢٠:١٦ ١٥ جورى

Serving Layer - Jupyter Notebook - Mozilla Firefox

localhost:8888/notebooks/Serving Layer.ipynb

Jupyter Serving Layer Last Checkpoint: an hour ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Kernel Logout

Results from Hadoop file System after batch data analysis

```
In [2]: myfile = open("batch-output.txt")
txt = myfile.read()
print(txt)
myfile.close()

Average Delayed time in year 2017 is 37.647597526 minutes
Average Delayed time in year 2018 is 36.3042229856 minutes
```

Results from Spark after stream data analysis

```
In [4]: myfile = open("stream-output.txt")
txt = myfile.read()
print(txt)
myfile.close()

month
11-2018
count
1677
sum_of_delay
75396.0
month
```

The screenshot shows a Linux desktop environment with a dark theme. A Firefox browser window is open, displaying a Jupyter Notebook titled "Serving Layer". The notebook contains two code cells. The first cell reads data from "batch-output.txt" and prints the average delayed time for years 2017 and 2018. The second cell reads data from "stream-output.txt" and prints various metrics including month, count, sum_of_delay, and month again. The desktop interface includes a dock at the bottom with various icons like Spl, Jupy, Cont, 14.0, Jupyter, Python, Jupyter, How, dock, Jupyter, Serv, Home, and others. On the left, there's a vertical dock with icons for file operations like Open, Save, Print, and a search icon.