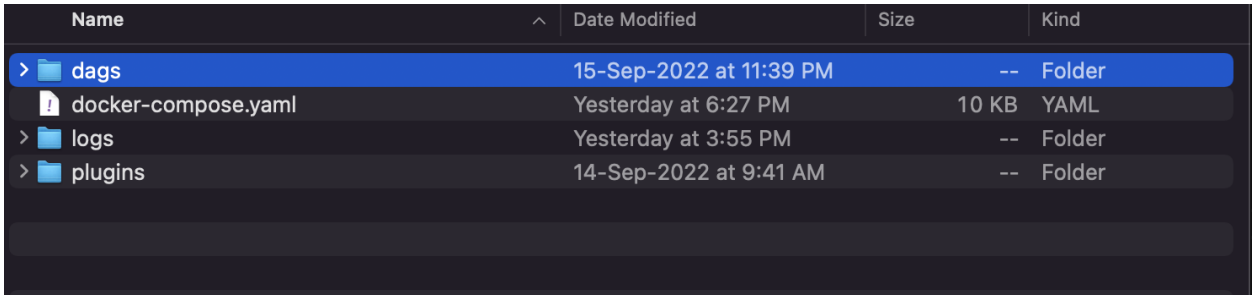
HIVE INSTALLATION :

Docker Installation :

Install Docker Desktop . Drag into applications. Open the docker

Create a repository with folders,files shown below

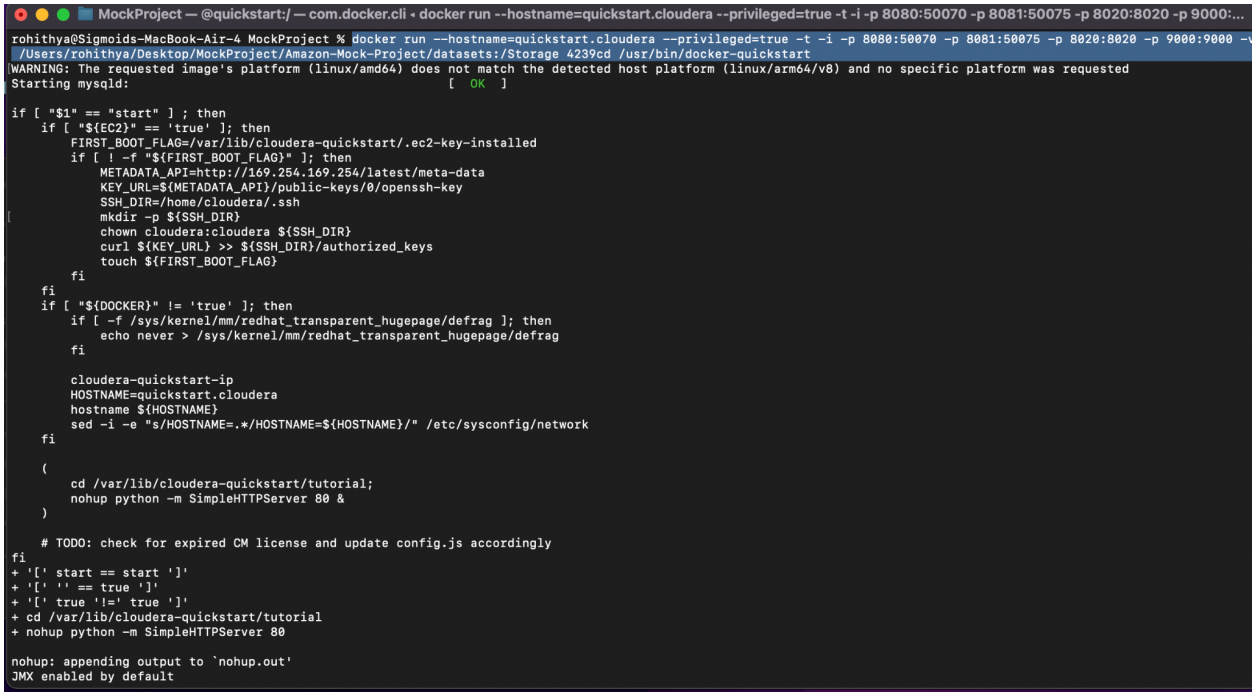


In terminal navigate to the above repository and type command
edit docker-compose.yaml with containers required

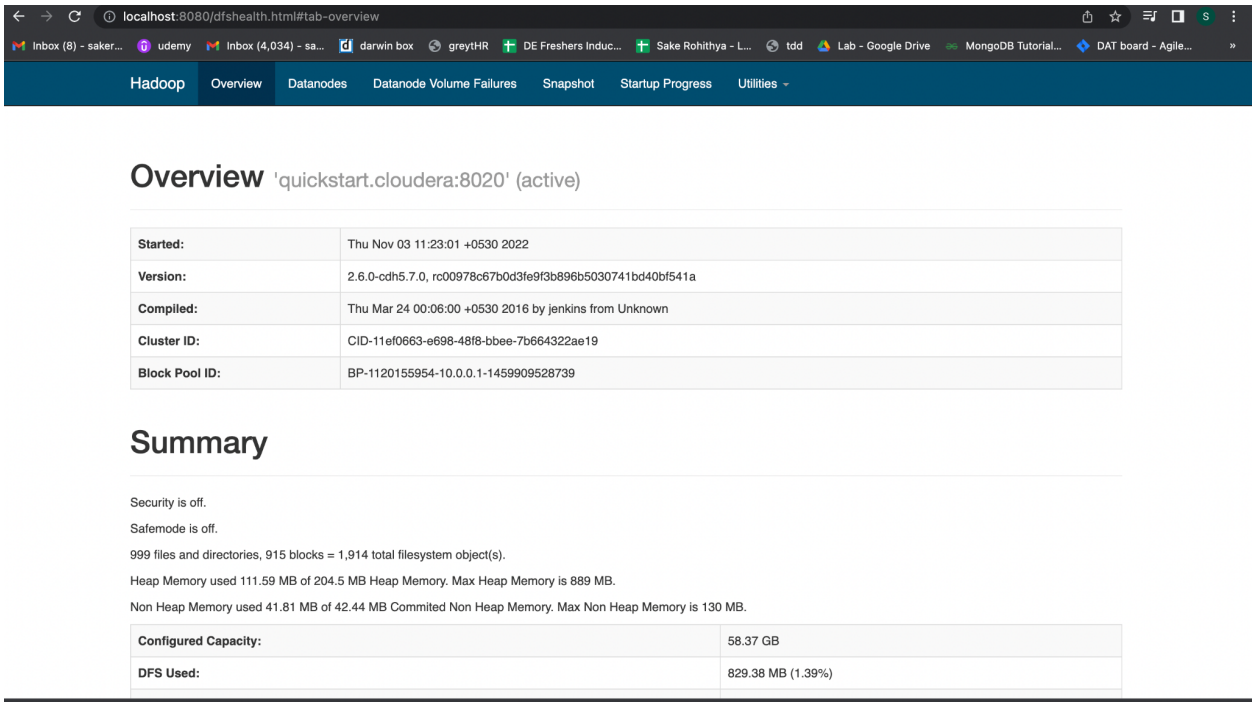
```
services:
  frontend:
    image: awesome/webapp
    ports:
      - "443:8043"
    networks:
      - front-tier
      - back-tier
    configs:
      - httpd-config
    secrets:
      - server-certificate

  backend:
    image: awesome/database
    volumes:
      - db-data:/etc/data
    networks:
      - back-tier

volumes:
  db-data:
    driver: flocker
    driver_opts:
      size: "10GiB"
```

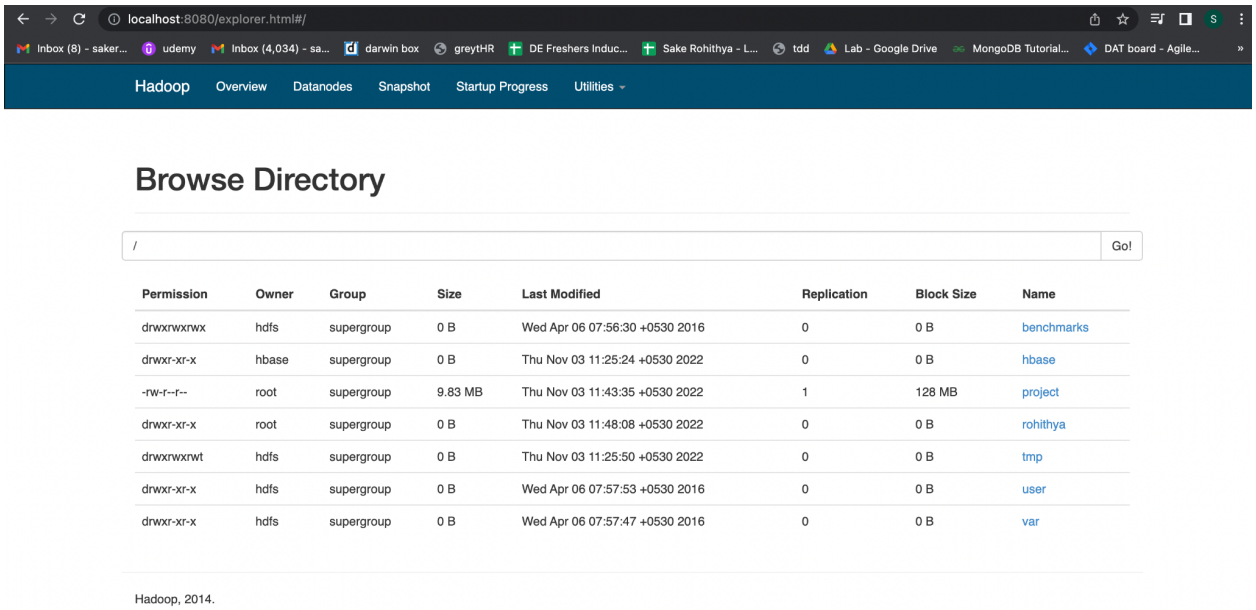
```
configs:
  httpd-config:
    external: true

secrets:
  server-certificate:
    external: true

networks:
  # The presence of these objects is sufficient to define them
  front-tier: {}
  back-tier: {}
```

```

**Starting Docker and Pulling Cloudera image:**

(Type below command in terminal)
docker-compose up -d —- Starts the container

docker pull cloudera/quickstart:latest —- Pulls Cloudera image where hive is inbuilt

docker images —- Check if cloudera image is successfully pulled

**Starting Hadoop environment :**

docker run --hostname=quickstart.cloudera --privileged=true -t -i -v [LocalPathAbs]:[ImagePath] [IMAGEID_First 6 Chararacters] /usr/bin/docker-quickstart

P.S : Before running above command make sure you empty containers
```



(This starts the hadoop environment. Browser Namenode → localhost:8080)





Making directory in HDFS and copying data from local storage to HDFS dir

[root@quickstart /]# hadoop fs -mkdir /rohithya

[root@quickstart /]# hadoop fs -copyFromLocal Storage/data.csv /rohithya

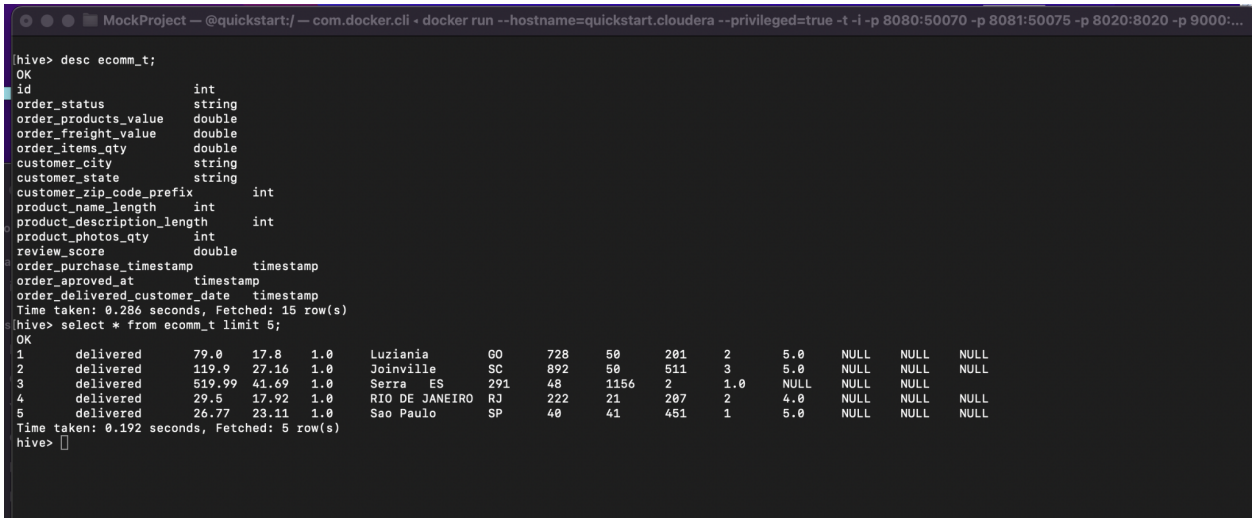
Start the hive environment

[root@quickstart /]# hive

Table creation and uploading data into table

hive> CREATE EXTERNAL TABLE ecomm_t (id int,order_status string,order_products_value double,order_freight_value double,order_items_qty double,customer_city string,customer_state string,customer_zip_code_prefix int,product_name_length int,product_description_length int,product_photos_qty int,review_score double,order_purchase_timestamp timestamp,order_aproved_at timestamp,order_delivered_customer_date timestamp) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION '/rohithya';

Querying the table to check if data is uploaded successfully or not :


MockProject — @quickstart:/ — com.docker.cli ◂ docker run --hostname=quickstart.cloudera --privileged=true -t -i -p 8080:50070 -p 8081:50075 -p 8020:8020 -p 9000:...
hive> desc ecomm_t;
OK
id int
order_status string
order_products_value double
order_freight_value double
order_items_qty double
customer_city string
customer_state string
customer_zip_code_prefix int
product_name_length int
product_description_length int
product_photos_qty int
review_score double
order_purchase_timestamp timestamp
order_aproved_at timestamp
order_delivered_customer_date timestamp
Time taken: 0.286 seconds, Fetched: 15 row(s)
hive> select * from ecomm_t limit 5;
OK
1 delivered 79.0 17.8 1.0 Luziania GO 728 50 201 2 5.0 NULL NULL NULL
2 delivered 119.9 27.16 1.0 Joinville SC 892 50 511 3 5.0 NULL NULL NULL
3 delivered 519.99 41.69 1.0 Serra ES 291 48 1156 2 1.0 NULL NULL NULL
4 delivered 29.5 17.92 1.0 RIO DE JANEIRO RJ 222 21 207 2 4.0 NULL NULL NULL
5 delivered 26.77 23.11 1.0 Sao Paulo SP 40 41 451 1 5.0 NULL NULL NULL
Time taken: 0.192 seconds, Fetched: 5 row(s)
hive>