LAB-1: MongoDB

1. Student Collection:

```
> use mySTUD

switched to db mySTUD


> db.getCollectionNames()

[ "Student" ]


> db.Student.insert({_id: 1, Name:"John", USN: "1B22CS001",Semester: 6,Dept_name: "CSE", CGPA: 9.6, Hobbies : ["Reading","Gardening"]})

WriteResult({ "nInserted" : 1 })

> db.Student.insert({_id: 2, Name:"Wick", USN: "1B22IS301",Semester: 4,Dept_name: "ISE", CGPA: 8.3, Hobbies : ["Reading","Gardening"]})

WriteResult({ "nInserted" : 1 })

> db.Student.insert({_id: 3, Name:"Horris", USN: "1B22EE021",Semester: 5,Dept_name: "EEE", CGPA: 9.3, Hobbies : ["eSports"]})

WriteResult({ "nInserted" : 1 })

> db.Student.insert({_id: 4, Name:"Arthur", USN: "1B22CS041",Semester: 6,Dept_name: "CSE", CGPA: 8.6, Hobbies : ["Novel Reading"]})

WriteResult({ "nInserted" : 1 })

> db.Student.insert({_id: 5, Name:"Tess", USN: "1B22ME011",Semester: 5,Dept_name: "ME", CGPA: 9.1, Hobbies : ["DIY"]})

WriteResult({ "nInserted" : 1 })

> db.Student.insert({_id: 6, Name:"Sylvia", USN: "1B22CS013",Semester: 5,Dept_name: "CSE", CGPA: 9.1, Hobbies : ["DIY"]})

WriteResult({ "nInserted" : 1 })

> db.Student.insert({_id: 7, Name:"Hritik", USN: "1B22CS014",Semester: 5,Dept_name: "CSE", CGPA: 8.7, Hobbies : ["Reading"]})

WriteResult({ "nInserted" : 1 })


> db.Student.find().pretty()

{

        "_id" : 1,

        "Name" : "John",

        "USN" : "1B22CS001",

        "Semester" : 6,
    "Dept_name" : "CSE",
    "CGPA" : 9.6,
    "Hobbies" : [
        "Reading",
        "Gardening"
    ]
}
{
    "_id" : 2,
    "Name" : "Wick",
    "USN" : "1B22IS301",
    "Semester" : 4,
    "Dept_name" : "ISE",
    "CGPA" : 8.3,
    "Hobbies" : [
        "Reading",
        "Gardening"
    ]
}
{
    "_id" : 3,
    "Name" : "Horris",
    "USN" : "1B22EE021",
    "Semester" : 5,
    "Dept_name" : "EEE",
    "CGPA" : 9.3,
    "Hobbies" : [
        "eSports"
    ]
}
{
    "_id" : 4,
    "Name" : "Arthur",
    "USN" : "1B22CS041",
    "Semester" : 6,
    "Dept_name" : "CSE",
    "CGPA" : 8.6,
    "Hobbies" : [
        "Novel Reading"
    ]
}
{
    "_id" : 5,
    "Name" : "Tess",
    "USN" : "1B22ME011",
    "Semester" : 5,
    "Dept_name" : "ME",
    "CGPA" : 9.1,
    "Hobbies" : [
        "DIY"
    ]
}
```

```
> db.Student.aggregate({$match :{Dept_name:"CSE"}},{$group: {_id:"$Semester",AvgCGPA:{$avg:"$CGPA"}}},{$match :{AvgCGPA:{$gt:7.5}}})
{ "_id" : 5, "AvgCGPA" : 8.899999999999999 }
{ "_id" : 6, "AvgCGPA" : 9.1 }
```

2. Bank Collection:

```
> use mySTUD
switched to db mySTUD

> db.createCollection("Bank")
{ "ok" : 1 }

> db.Bank.insert({name: "Arka", type:"savings", transactions: ["+1000", "-100", "+5000"], Balance:1400})
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({name: "Derek", type:"savings", transactions: ["-100", "+300", "+500"], Balance:5500})
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({name: "Shreastha", type:"savings", transactions: ["+200", "-300", "+60", "-70"], Balance:8000})
WriteResult({ "nInserted" : 1 })
```

```
> db.Bank.insert({name: "Harries", type:"savings", transactions: ["+600", "-7000"], Balance:11000})
WriteResult({ "nInserted" : 1 })




> db.Bank.update({name:"Derek"},{$pull:{transactions:{$in:["+500"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Bank.find().pretty()
{
        "_id" : ObjectId("626649b716596fe24c1442bb"),
        "name" : "Arka",
        "type" : "savings",
        "transactions" : [
                "+1000",
                "-100"
        ],
        "Balance" : 1400
}
{
        "_id" : ObjectId("626649d116596fe24c1442bc"),
        "name" : "Derek",
        "type" : "savings",
        "transactions" : [
                "-100",
                "+300"
        ],
        "Balance" : 5500
}
{
        "_id" : ObjectId("626649f116596fe24c1442bd"),
        "name" : "Shreastha",
        "type" : "savings",
        "transactions" : [
                "+200",
                "-300",
                "+60",
                "-70"
        ],
        "Balance" : 8000
}
{
        "_id" : ObjectId("62664a1216596fe24c1442be"),
        "name" : "Harries",
        "type" : "savings",
        "transactions" : [
                "+600",
                "-7000"
        ],
        "Balance" : 11000
}

> db.Bank.update({name: "Shreastha"},{$pop:{transactions:-1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Bank.find().pretty()
{
        "_id" : ObjectId("626649b716596fe24c1442bb"),
        "name" : "Arka",
        "type" : "savings",
        "transactions" : [
                "+1000",
                "-100"
        ],
        "Balance" : 1400
}
{
        "_id" : ObjectId("626649d116596fe24c1442bc"),
        "name" : "Derek",
        "type" : "savings",
        "transactions" : [
                "-100",
                "+300"
        ],
        "Balance" : 5500
}
{
        "_id" : ObjectId("626649f116596fe24c1442bd"),
        "name" : "Shreastha",
        "type" : "savings",
        "transactions" : [
                "-300",
                "+60",
                "-70"
        ],
        "Balance" : 8000
}
{
        "_id" : ObjectId("62664a1216596fe24c1442be"),
        "name" : "Harries",
        "type" : "savings",
        "transactions" : [
                "+600",
                "-7000"
        ],
        "Balance" : 11000
}
```

LAB-2 & 3: Cassandra

1. Employee Keyspace:

```
cqlsh:system> CREATE KEYSPACE employee with replication = {
          ... 'class':'SimpleStrategy','replication_factor':1};


cqlsh:system> USE employee;


cqlsh:employee> CREATE TABLE employee_info (
            ... emp_id text,
            ... emp_name text,
            ... designation text,
            ... date_of_joining date,
            ... salary float,
            ... dept_name text,
            ... PRIMARY KEY(emp_id)
            ... );


cqlsh:employee> DESC TABLES

employee_info

cqlsh:employee> BEGIN BATCH
            ... INSERT INTO employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
            ... VALUES ('120','John','Software Engineering','2020-01-01',80000,'Development')
            ... INSERT INTO employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
            ... VALUES ('121','Harry','Debugger','2020-04-11',60000,'Development')
            ... INSERT INTO employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
            ... VALUES ('122','Clark','Tester','2020-02-21',75000,'Testing')
            ... APPLY BATCH;


cqlsh:employee> SELECT * FROM employee_info;

 emp_id | date_of_joining | dept_name   | designation          | emp_name | salary
--------+-----------------+-------------+----------------------+----------+--------
    120 |      2020-01-01 | Development | Software Engineering |     John |  80000
    121 |      2020-04-11 | Development |             Debugger |    Harry |  60000
    122 |      2020-02-21 |     Testing |               Tester |    Clark |  75000

(3 rows)


cqlsh:employee> UPDATE employee_info SET emp_name='Potter',dept_name='Testing'
            ... WHERE emp_id='121';


cqlsh:employee> ALTER TABLE employee_info ADD Projects set<text>;
```

```
cqlsh:employee> UPDATE employee_info SET Projects={'SQL','QT'} WHERE emp_id='120';

cqlsh:employee> UPDATE employee_info SET Projects={'UI/UX','PYPY3'} WHERE emp_id='121';

cqlsh:employee> UPDATE employee_info SET Projects={'Voice Module','DATACENTER'} WHERE emp_id='122';

cqlsh:employee> SELECT * FROM employee_info;

 emp_id | date_of_joining | dept_name   | designation          | emp_name | projects                          | salary
--------+-----------------+-------------+----------------------+----------+-----------------------------------+--------
    120 |      2020-01-01 | Development | Software Engineering |     John |                   {'QT', 'SQL'} |  80000
    121 |      2020-04-11 |     Testing |             Debugger |   Potter |              {'PYPY3', 'UI/UX'} |  60000
    122 |      2020-02-21 |     Testing |               Tester |    Clark | {'DATACENTER', 'Voice Module'} |  75000

(3 rows)


cqlsh:employee> INSERT INTO employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
            ... VALUES ('123','James','System Design Lead','2020-03-02',90000,'Development') USING TTL 15;


cqlsh:employee> SELECT TTL(emp_name) FROM employee_info WHERE emp_id='123' ;

 ttl(emp_name)
---------------
             9

(1 rows)


cqlsh:employee> SELECT TTL(emp_name) FROM employee_info WHERE emp_id='123' ;

 ttl(emp_name)
---------------
             6

(1 rows)


cqlsh:employee> SELECT * FROM employee_info ;

 emp_id | date_of_joining | dept_name   | designation          | emp_name | projects                          | salary
--------+-----------------+-------------+----------------------+----------+-----------------------------------+--------
    123 |      2020-03-02 | Development |   System Design Lead |    James |                              null |  90000
    120 |      2020-01-01 | Development | Software Engineering |     John |                   {'QT', 'SQL'} |  80000
    121 |      2020-04-11 |     Testing |             Debugger |   Potter |              {'PYPY3', 'UI/UX'} |  60000
    122 |      2020-02-21 |     Testing |               Tester |    Clark | {'DATACENTER', 'Voice Module'} |  75000


(4 rows)

cqlsh:employee> SELECT TTL(emp_name) FROM employee_info WHERE emp_id='123' ;

 ttl(emp_name)
---------------


(0 rows)
cqlsh:employee> SELECT * FROM employee_info ;

 emp_id | date_of_joining | dept_name   | designation          | emp_name | projects                          | salary
--------+-----------------+-------------+----------------------+----------+-----------------------------------+--------
    120 |      2020-01-01 | Development | Software Engineering |     John |                   {'QT', 'SQL'} |  80000
    121 |      2020-04-11 |     Testing |             Debugger |   Potter |              {'PYPY3', 'UI/UX'} |  60000
    122 |      2020-02-21 |     Testing |               Tester |    Clark | {'DATACENTER', 'Voice Module'} |  75000

(3 rows)
```

2. Library Keyspace:

```
cqlsh> CREATE KEYSPACE Library with replication={ 'class':'SimpleStrategy', 'replication_factor':1};

cqlsh> USE Library

cqlsh:library> CREATE TABLE library_info(
           ... stud_id int,
           ... stud_name text,
           ... book_id int,
           ... book_name text,
           ... counter_value counter,
           ... date_of_issue date,
           ... PRIMARY KEY((stud_id,book_id),stud_name,book_name,date_of_issue)
           ... );

cqlsh:library> DESC library_info;

CREATE TABLE library.library_info (
    stud_id int,
    book_id int,
    stud_name text,
    book_name text,
    date_of_issue date,
    counter_value counter,
    PRIMARY KEY ((stud_id, book_id), stud_name, book_name, date_of_issue)
) WITH CLUSTERING ORDER BY (stud_name ASC, book_name ASC, date_of_issue ASC)
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy',
                     'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

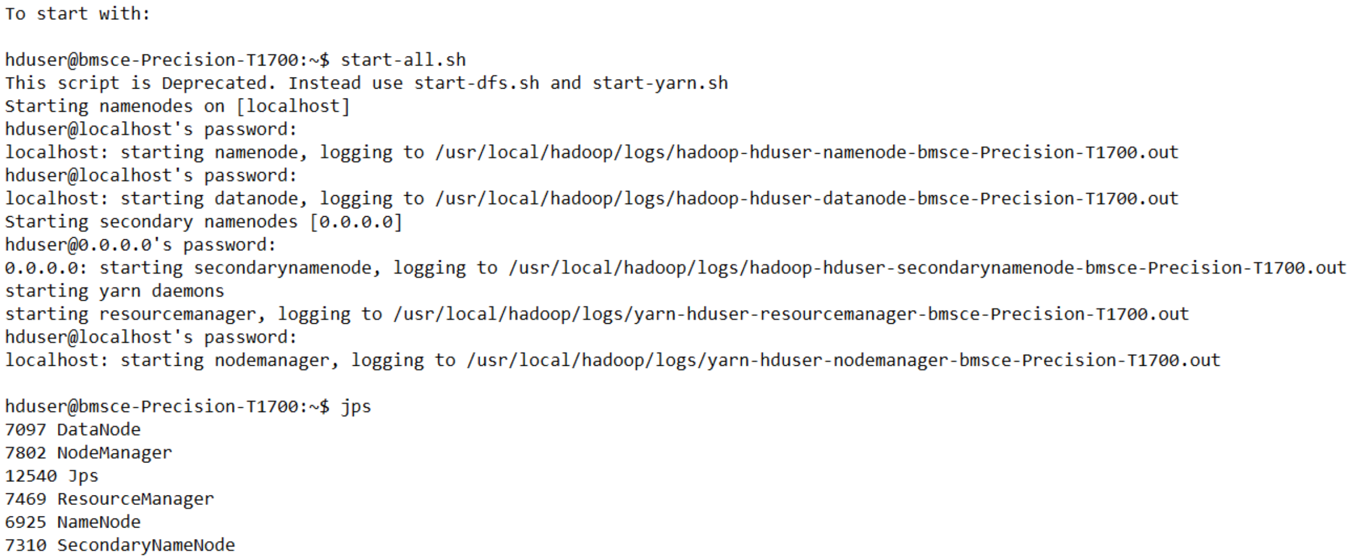
```
cqlsh:library> UPDATE library_info SET counter_value = counter_value + 1  WHERE book_id=100 and stud_id=112
                and stud_name='Krishna' and book_name='BDA'and date_of_issue='2020-02-02' ;
cqlsh:library> UPDATE library_info SET counter_value = counter_value + 1  WHERE book_id=100 and stud_id=112
                and stud_name='Krishna' and book_name='BDA'and date_of_issue='2020-02-02' ;
cqlsh:library> UPDATE library_info SET counter_value = counter_value + 1  WHERE book_id=201 and stud_id=132
                and stud_name='Arthur' and book_name='CNS'and date_of_issue='2020-02-05' ;
cqlsh:library> UPDATE library_info SET counter_value = counter_value + 1  WHERE book_id=111 and stud_id=202
                and stud_name='Alan' and book_name='OOMD'and date_of_issue='2020-09-12' ;


cqlsh:library> SELECT * FROM library_info;

 stud_id | book_id | stud_name | book_name | date_of_issue | counter_value
---------+---------+-----------+-----------+---------------+---------------
     132 |     201 |    Arthur |       CNS |    2020-02-05 |             1
     112 |     100 |   Krishna |       BDA |    2020-02-02 |             2
     202 |     111 |      Alan |      OOMD |    2020-09-12 |             1

(3 rows)


cqlsh:library> SELECT * from library_info WHERE stud_id=112 and book_id=100;

 stud_id | book_id | stud_name | book_name | date_of_issue | counter_value
---------+---------+-----------+-----------+---------------+---------------
     112 |     100 |   Krishna |       BDA |    2020-02-02 |             2


cqlsh:library> COPY library_info(stud_id,book_id,stud_name,book_name,date_of_issue,counter_value) TO 'LIB.csv';
Using 1 child processes

Starting copy of library.library_info with columns [stud_id, book_id, stud_name, book_name, date_of_issue, counter_value].
Processed: 3 rows; Rate:       1 rows/s; Avg. rate:       1 rows/s
3 rows exported to 1 files in 2.418 seconds.


cqlsh:library> TRUNCATE library_info;
cqlsh:library> SELECT * FROM library_info;
```

```
 stud_id | book_id | stud_name | book_name | date_of_issue | counter_value
---------+---------+-----------+-----------+---------------+---------------

(0 rows)


cqlsh:library> COPY library_info(stud_id,book_id,stud_name,book_name,date_of_issue,counter_value) FROM 'LIB.csv';
Using 1 child processes

Starting copy of library.library_info with columns [stud_id, book_id, stud_name, book_name, date_of_issue, counter_value].
Processed: 3 rows; Rate:       3 rows/s; Avg. rate:       5 rows/s
3 rows imported from 1 files in 0.593 seconds (0 skipped).
cqlsh:library> SELECT * FROM library_info;

 stud_id | book_id | stud_name | book_name | date_of_issue | counter_value
---------+---------+-----------+-----------+---------------+---------------
     132 |     201 |    Arthur |       CNS |    2020-02-05 |             1
     112 |     100 |   Krishna |       BDA |    2020-02-02 |             2
     202 |     111 |      Alan |      OOMD |    2020-09-12 |             1

(3 rows)
```

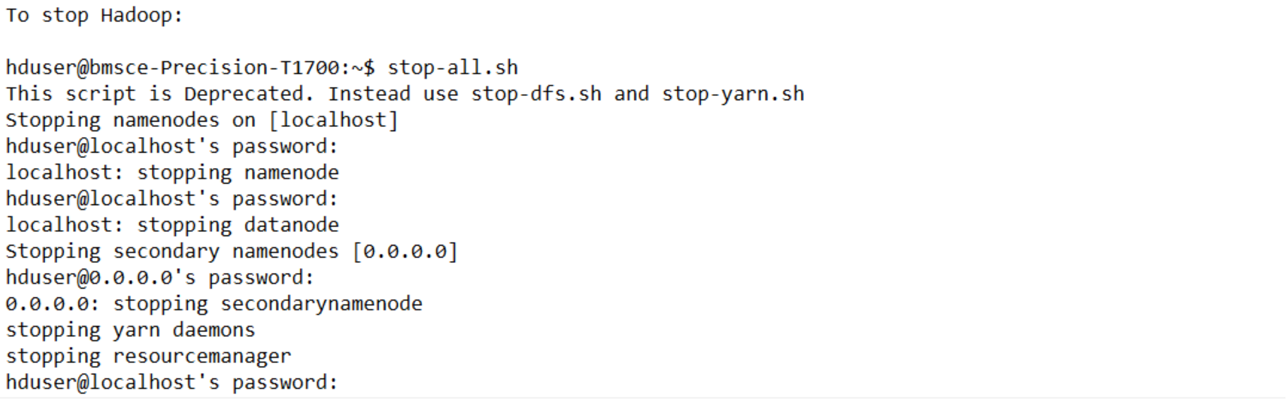
LAB-4: Hadoop Commands

```
1:
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /hadoop


2:
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 1 item
drwxr-xr-x   - hduser supergroup          0 2022-06-06 11:37 /hadoop


3:
hduser@bmsce-Precision-T1700:~$ hdfs dfs -put /home/hduser/Desktop/hadoop.txt /hadoop/hadoop.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /hadoop/hadoop.txt
Hello, I'm Hadoop


4:
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyFromLocal /home/hduser/Desktop/hadoop.txt /hadoop/hadoop2.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /hadoop/hadoop.txt
Hello, I'm Hadoop


5:
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /hadoop/hadoop1.txt /home/hduser/Desktop/hd.txt
hduser@bmsce-Precision-T1700:~$ ls Desktop/hd.txt
Desktop/hd.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -getmerge /hadoop/hadoop.txt /hadoop/hadoop2.txt /home/hduser/Desktop/hd_merge.txt
hduser@bmsce-Precision-T1700:~$ ls Desktop/hd_merge.txt
Desktop/hd_merge.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -getfacl /hadoop
# file: /hadoop
# owner: hduser
# group: supergroup
user::rwx
group::r-x
other::r-x
```

```
6:
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /hadoop/hadoop.txt /home/hduser/Desktop/hd2.txt

hduser@bmsce-Precision-T1700:~$ ls Desktop/hd2.txt
Desktop/hd2.txt


7:
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /hadoop/hadoop.txt
Hello, I'm Hadoop


8:
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /hadoop/AA
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mv /hadoop/hadoop.txt /hadoop/AA/hadoop.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /hadoop/AA
Found 1 items
-rw-r--r--   1 hduser supergroup         18 2022-06-06 11:41 /hadoop/AA/hadoop.txt


9:
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cp /hadoop/AA/hadoop.txt /hadoop/hadoop2.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /hadoop/hadoop2.txt
Hello, I'm Hadoop
```



```
To start with:

hduser@bmsce-Precision-T1700:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
hduser@localhost's password:
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-bmsce-Precision-T1700.out
hduser@localhost's password:
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-bmsce-Precision-T1700.out
Starting secondary namenodes [0.0.0.0]
hduser@0.0.0.0's password:
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hduser-secondarynamenode-bmsce-Precision-T1700.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-bmsce-Precision-T1700.out
hduser@localhost's password:
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-bmsce-Precision-T1700.out

hduser@bmsce-Precision-T1700:~$ jps
7097 DataNode
7802 NodeManager
12540 Jps
7469 ResourceManager
6925 NameNode
7310 SecondaryNameNode
```



```
To stop Hadoop:

hduser@bmsce-Precision-T1700:~$ stop-all.sh
This script is Deprecated. Instead use stop-dfs.sh and stop-yarn.sh
Stopping namenodes on [localhost]
hduser@localhost's password:
localhost: stopping namenode
hduser@localhost's password:
localhost: stopping datanode
Stopping secondary namenodes [0.0.0.0]
hduser@0.0.0.0's password:
0.0.0.0: stopping secondarynamenode
stopping yarn daemons
stopping resourcemanager
hduser@localhost's password:
```