

HADOOP

①

Exp 1 :- Configuration → check / verify all the softwares are in good health or not.

2] * to check java path.

\$ cd /usr/lib/jvm.

\$ ls.

→ go to browser

→ click on cloudera Manager

→ Enter username &

password : Cloudera

3] * to check all components

\$ sudo jps

→ click login

→ check hdfs & map
reduce is in good

health.

4] * to check hadoop location

\$ cd /usr/lib/hadoop-0.20-mapreduce → if not, then

\$ ls

refresh,

5] * to verify hadoop installation file.

\$ cd conf

\$ ls.

6] * open a file.

gedit core-site.xml.

file name.

edit hdfs-site.xml.

Exp 2 :- Loading a file from local file system to hadoop file system.

- Open the terminal
- to verify whether all daemons are running or not.
- \$ sudo jps
- create a folder or directory in hadoop.
\$ hadoop fs -mkdir /user/cloudera/nh001b
(directory/filename)
- if name node is in safe mode then give below command
:- \$ sudo -u hdfs hdfs dfsadmin -safemode leave.
- To verify whether or not the folder is created.
\$ hadoop fs -ls /user/cloudera.
- to note down the current directory location before creating the file command.
\$ pwd (present working directory)
o/p:- /usr/lib/hadoop-0.20-mapreduce/conf
- open minimize the present terminal & open new terminal.
- to create a file named 'nh001b' in local file system.
\$ gedit nh001b.nh156
 - ↳ editor using a file can read, edit..
 - ↳ automatic file will be opened, write few same statements, save & close.
- to verify whether or not file is created.
\$ ls

→ to put the local file into the hadoop file system

\$ hadoop fs -put /home/cloudera/nh156 space
/user/cloudera/nh001b (file name)
(directory name)

→ to verify whether or not the local file is loaded into hadoop file.

\$ hadoop fs -ls /user/cloudera/nh001b
%

→ to check the content of loaded file

\$ hadoop fs -cat /user/cloudera/packagethh001b/nh156

%

to verify the loaded file from browser

→ open the firefox browser pro in cloudera

↓ click
hdfs Name Node

↓ click

browse file system

↓ click

User

↓ click

cloudera

↓ click

nh001b

↓ click

nh156

Exp 3:- Perform analysis on loaded files using hadoop mapreduce programs & verify the output using hadoop commands as well as browser inbuilt command.

a) Count

b) grep.

→ to see the list of jar files available in hadoop
\$ cd /usr/lib/hadoop-0.20-mapreduce
\$ ls
(red colour files are jar files)

→ to see the content of jar file.

\$ hadoop jar /usr/lib/hadoop-0.20-mapreduce/hadoop-examples-2.0.0-mr1-cdh4.4.0.jar

→ Wordcount.

(grep)

\$ (above command i.e hadoop jarjar) wordcount
/user/cloudera/nh001b/nh156 user/cloudera/nh001b/
disc name file name disc name

summy.

new file name to save output word count.

new file name to save output of grep

→ to verify output files.

\$ hadoop fs -ls /user/cloudera/nh001b

\$ hadoop fs -cat /user/cloudera/nh001b/summy

→ to see the content of output file.

\$ hadoop fs -cat /user/cloudera/nh001b/summy/part*

O/p:- E-10, Z-10 etc output of word count.

→ to view same output in browser.

→ hdfs Namenode → browser file system → user → cloudera → nh001b

→ summy → part-1... → O/p will be displayed.

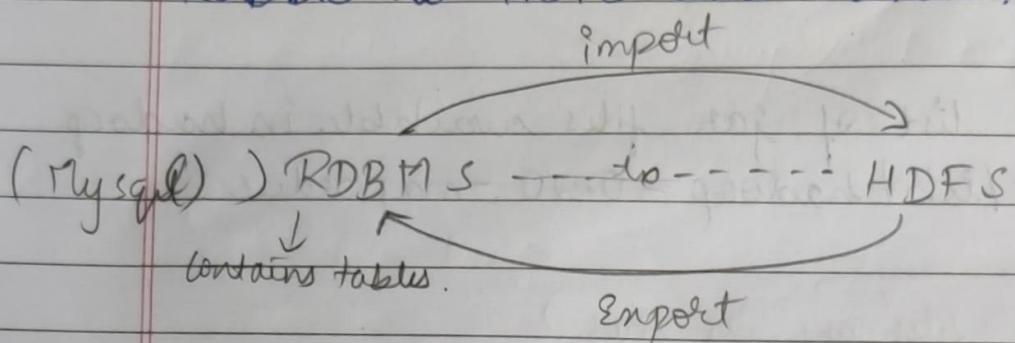
(summy)

↳ ~~group~~

SQL + Hadoop = Sqoop.

which helps

Sqoop is the intermediately, to transfer the files from RDBMS to HDFS and vice versa.



SQOOP

Exp 1:- Verifying Sqoop status through cloudera Manager.

→ Open the terminal.

* to Start mysql service.

\$ sudo service mysqld start.

* To connect to mysql.

\$ mysql -u root.

→ to create database.

\$ > create database ramyaashree;

database name.

> show databases;

> exit.

→ if Name node is in safemode then use the query.

\$ sudo -u hdfs hdfs dfsadmin -safemode leave

Output:- Safe mode is off

Since we exited from mysql & we are in cloudera, so again type command.

→ sudo service mysqld start

→ mysql -u root

to enter into mysql.

mysql> use ramyashree;

mysql> create table sty(cno int);

mysql> create table teach(id int, name char);

mysql> insert into sty values (101), (102), (103), (104), (105);

mysql> exit;insert into teach values(10, 'a'), (20, 'b'),
(30, 'c'), (40, 'd'), (50, 'e');

mysql> exit

^{"Eval"}
Exp 2:- List databases, tables & Execute using command.

→ list databases

\$ sqoop list-databases --connect

\$ sqoop list-databases --connect "jdbc:mysql://localhost" --username root

→ list databases

\$ sqoop list-databases --connect "jdbc:mysql://localhost" --username root

→ list tables.

\$ sqoop list-tables --connect "jdbc:mysql://localhost/ramyashree" --username root

db name

→ Eval command

\$ sqoop eval --connect "jdbc:mysql://localhost/ramyashree"
--username root --query "Select * from sty"

Note :- No need to add semicolon in sqoop.

Count no of rows in teach table:-

```
$ sgoop eval --connect "jdbc:mysql://localhost/ramyashree"
--username root --query "select count(*) from teach"
```

```
$ sgoop eval -- --query "insert into teach values(60,
'f')"
```

```
$ sgoop eval -- --query "select * from teach"
```

Exp 3:- importing

all tables

subset of tables - where cond.

specific tables

incremental import

```
// hadoop fs -mkdir /user/cloudera/ramyasgoop
```

i) import all subset of tables.

```
$ sgoop import --connect "jdbc:mysql://localhost/ramyashree"
```

```
--username root --table teach --where "id = '20'"
```

```
--target-dir /user/cloudera/ramyasgoop/sgooplabs156
```

```
-m 1
```

0/p:-

retreived "1" records

to check the imported table in the target directory.

```
$ hadoop fs -ls /user/cloudera/ramyasgoop
```

```
$ hadoop fs -ls /user/cloudera/ramyasgoop/sgooplabs156
```

```
$ hadoop fs -cat /user/cloudera/ramyasgoop/sgooplabs156/
part*
```

i) import specific table to default directory.

```
$ sgoop import --connect "jdbc:mysql://localhost/ramyashree"
--username root --table teach -m 1
```

mapreduce default value

to check the imported table in the default directory.

```
$ hadoop fs -ls /user/cloudera/
```

```
$ hadoop fs -ls /user/cloudera/teach
```

\$ hadoop fs -cat /user/cloudera/teach/part-m-00000

iii) import all tables.

\$ sqoop import-all-tables --connect "jdbc:mysql://localhost/^{use db}/ramyashru" --username root -m 1

(if error file already exists then open new terminal connect mysql & drop the existing tables & create new tables & insert values). & then come back to the existing terminal & write import-all-tables -- command.

→ to check the imported table.

\$ hadoop fs -ls /user/cloudera

hadoop fs -ls /user/cloudera/newsty

hadoop fs -ls /user/cloudera/newteacher/part*

hadoop fs -cat /user/cloudera/newsty/part*

iv) incremental import

insert data: \$ sqoop eval --connect "jdbc:mysql://localhost/ramyashru" --username root --query "insert into sty values(106),(107)"

→ incremental import

\$ sqoop import --connect "jdbc:mysql://localhost/ramyashru" --username root --table sty --target-dir /user/cloudera/ramyashru/sqooplab156 --incremental append --check-column sno --last-value 105 -m 1

put = to move

classmate

Date _____

Page _____

Exp 4:- Export hdfs files to mysql.

Create a file

→ \$ gedit exp4156
file name

↳ automatically file opens, write, save & close.

// in file

1, a

2, b

3, c

4, d.

- move file from home to user

leave safe mode

\$ sudo -u hdfs hdfs dfsadmin --safemode leave

move file from home to user (check whether we are in home or user by using "pwd" command).

\$ hadoop fs -put /home/cloudera/exp4156 /user/cloudera

\$ hadoop fs -cat /user/cloudera/exp4156

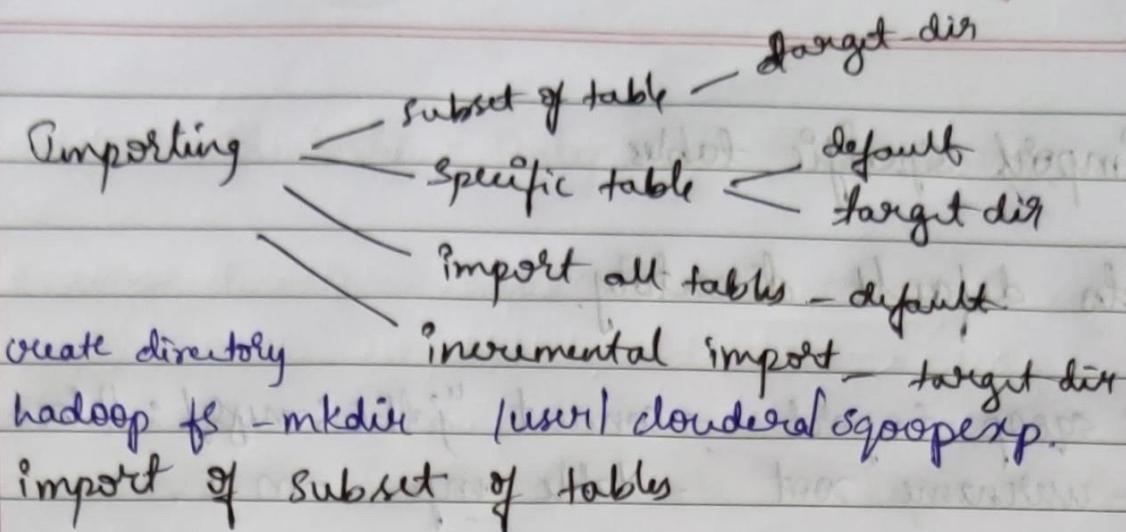
Creating → \$ sqoop eval --connect "jdbc:mysql://localhost/ramyashree" --username root --query "create table test_table (id int, name varchar)"

Export → \$ sqoop export --connect "jdbc:mysql://localhost/ramyashree" --username root --table test_table --import-dir /user/cloudera/test



to check.

\$ sqoop wal --connect "jdbc:mysql://localhost/bamyashree"
--username root --query "select * from test_table".



a) using where clause.

```
$ sqoop import --connect "jdbc:mysql://localhost/dbname"
--username root --table tablename teach --where "id = '2'"
--target-dir /user/cloudera/sqoopexp/op -m 1.
directory filename =
```

To check the imported table in target directory.

```
$ hadoop fs -ls /user/cloudera/sqoopexp
```

```
$ hadoop fs -ls /user/cloudera/sqoopexp/1 op
```

```
$ hadoop fs -cat /user/cloudera/sqoopexp/1 op/part*
```

b) using where clause & columns.

```
$ sqoop import --connect "jdbc:mysql://localhost/dbname"
--username root --table teach --columns "col1,col2"
--where "id > '2'" --target-dir /user/cloudera/
sqoopexp/2 op2 -m 1.
```

To check the imported tables in target directory.

```
$ hadoop fs -ls /user/cloudera/sqoopexp
```

```
$ hadoop fs -ls /user/cloudera/sqoopexp/2 op2
```

```
$ hadoop fs -cat /user/cloudera/sqoopexp/2 op2/part*
```

2) Import Specific tables

a) to default directory.

```
$ sqoop import --connect "jdbc:mysql://localhost/dbname"  
--username root --table emp -m 1
```

To check the imported tables in default directory.

```
$ hadoop fs -ls /user/cloudera/emp
```

```
-cat /user/cloudera/emp/part*
```

b) to target directory.

```
$ sqoop import --connect "jdbc:mysql://localhost/dbname"  
--username root --table emp --target-dir  
/user/cloudera/sqoopexp/ops -m 1
```

To check the imported table in target directory

```
$ hadoop fs -ls /user/cloudera/sqoopexp/ops
```

```
-cat /user/cloudera/sqoopexp/ops/part*
```

3] import All tables - Only default directory

```
$ sqoop import-all-tables --connect "jdbc:mysql://localhost/dbname" --username root -m 1.
```

To check the imported table in default directory.

```
$ hadoop fs -ls /user/cloudera
$ -l - /usr/cloudera/table name.
$ -cat /user/cloudera/table name/part*
```

4] Incremental import.

Insert new records into the table.

```
$ sqoop eval --connect "jdbc:mysql://localhost/dbname"
--username root --query "insert into tableemp values (106, 107)"
```

Import new records into existing hadoop op file.

```
$ sqoop import --connect "jdbc:mysql://localhost/dbname"
--username root --table emp --target-dir /user/cloudera
sqoopexp/01 --incremental append --check-column
eno --last-value 105 -m 1.
```

To check the imported table in,

```
$ hadoop fs -ls /user/cloudera/sqoopexp/01
$ -l - /sqoopexp/01
$ -cat /user/cloudera/sqoopexp/01/part*
```

NOTE

If we get error as file already exist while importing tables, then open new terminal connect mysql & drop existing tables & create new tables & insert values & then come back to existing terminal & write 'import-all-tables' command.