Lab on Cloud Guide for Self Learning Hadoop Developer

Author(s)	Manju K
Authorized by	Infosys Ltd
Creation/Revision Date	Aug 2018
Version	1.1

Usage Guidelines

Do not forward this document to any non-Infosys mail ID. Forwarding this document to a non-Infosys mail ID may lead to disciplinary action against you, including termination of employment.

Contents of this material cannot be used in any other internal or external document without explicit permission from ETA@infosys.com.

COPYRIGHT NOTICE

Education, Training and Assessment Department
Infosys Limited
Electronics City
Hosur Road
Bangalore - 561 229, India.

Tel: 91 80 852 0261-270
Fax: 91 80 852 0362
www.infosys.com
mailto:ETA@infosys.com

Document Revision History

Version	Date	Author(s)	Reviewer(s)	Description
1.0	14-May-18	Manju K	Ajit_RavindranNair	Self-learning lab guide for Hadoop developer
1.1	17-Aug-18	Manju K	Ajit_RavindranNair	Updated for Lex Links

CONTENTS

Lab on Cloud Guide for Self Learning Hadoop Developer

Prelude

This document will help you to work on the various components of Hadoop Stack - Hadoop, MapReduce, Hive, Pig and Sqoop.

For learning Big Data Technology Landscape click here.

To learn the Hadoop Architecture and its components click here.

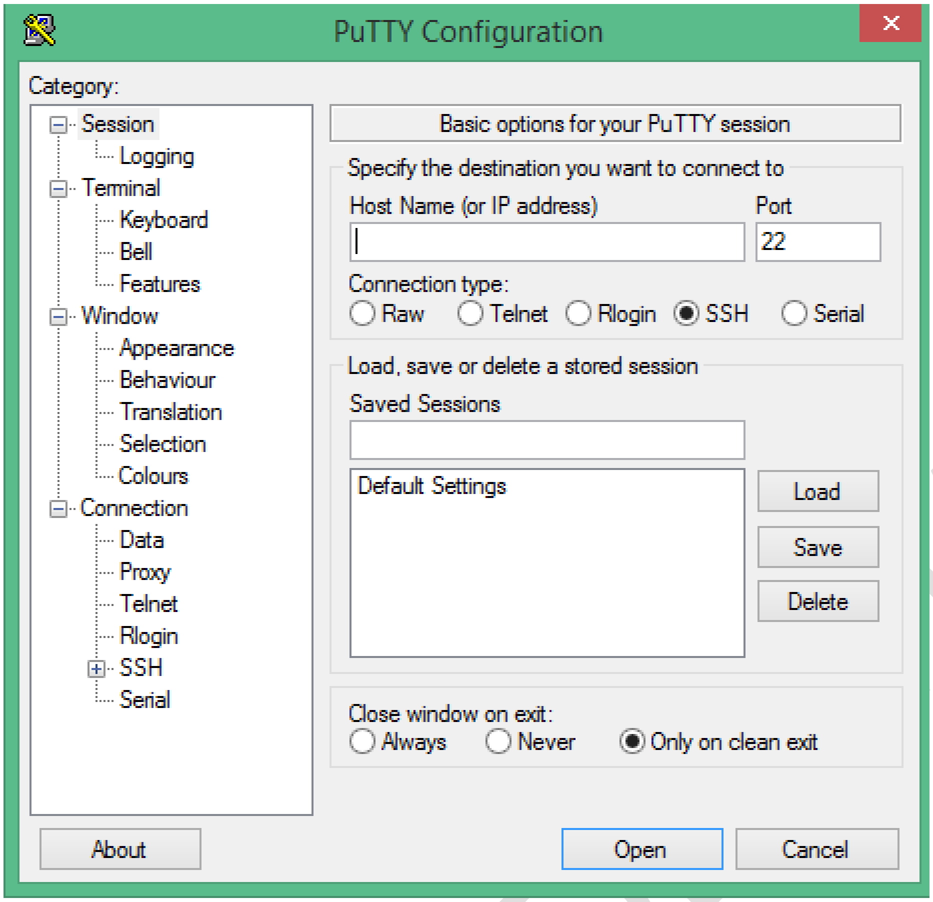
Below are the LEX courses suggested:

1. Big Data Technology Landscape
2. Introduction to Hadoop Framework
3. MapReduce Programming
4. Introduction to Apache Hive

1. Connecting to the Hadoop cluster

1.Connecting to the Lab on Cloud Hadoop Cluster from Windows/Mac

Download putty from Sparsh Downloads link. Use putty to connect using the credentials shared. Snapshot of putty given for reference.



Host Name (or IP address):

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

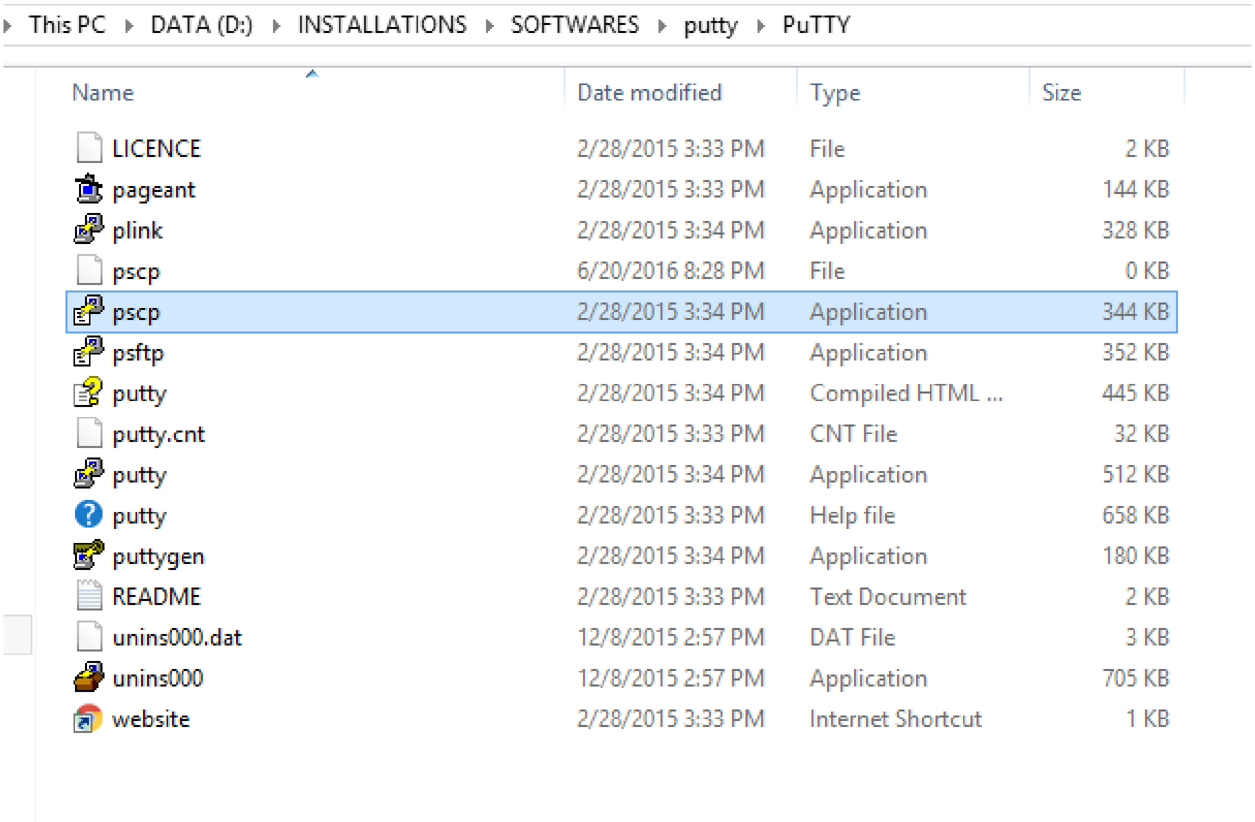
To transfer files from local system to the cluster machine:

Open command prompt and navigate to the path where PSCP is present in Putty installation folder as shown below. The command for file transfer is:

pscp <<File to be transferred in our PCs>> <<Linux_username>>@<<LinuxHost>>:<<Linux folder path where the file is to be transferred>>

Example: pscp C:\Users\userfile.txt clusteruser@myhost:/home/user_trng/Myfiles/
Where:

C:\Users\userfile.txt is the file in our PC → Source Path
/home/user_trng/Myfiles/ → Destination Path



This PC ▸ DATA (D:) ▸ INSTALLATIONS ▸ SOFTWARES ▸ putty ▸ PuTTY

Name	Date modified	Type	Size
LICENCE	2/28/2015 3:33 PM	File	2 KB
pageant	2/28/2015 3:33 PM	Application	144 KB
plink	2/28/2015 3:34 PM	Application	328 KB
pscp	6/20/2016 8:28 PM	File	0 KB
pscp	2/28/2015 3:34 PM	Application	344 KB
psftp	2/28/2015 3:34 PM	Application	352 KB
putty	2/28/2015 3:34 PM	Compiled HTML ...	445 KB
putty.cnt	2/28/2015 3:33 PM	CNT File	32 KB
putty	2/28/2015 3:34 PM	Application	512 KB
putty	2/28/2015 3:33 PM	Help file	658 KB
puttygen	2/28/2015 3:34 PM	Application	180 KB
README	2/28/2015 3:33 PM	Text Document	2 KB
unins000.dat	12/8/2015 2:57 PM	DAT File	3 KB
unins000	12/8/2015 2:57 PM	Application	705 KB
website	2/28/2015 3:33 PM	Internet Shortcut	1 KB

2. Working on Hadoop HDFS commands

Refer to the entire list of Hadoop shell commands available here

Assignment : Solve the exercise given in the LEX link here

Please note, you will be given access only to the hdfs folder /user/LabOnCloud in the cluster node connected. Create a directory by name empum_empname (eg 1111_MyName) in the hdfs at /user/LabOnCloud while working on Hadoop commands.

The jar files to be used for MapReduce Programming and Hive UDFs are available at **/user/LabOnCloud/Jars** HDFS path.

Estimated Time: 60 minutes

3. MapReduce Programming

Learn about MapReduce ,its architecture and programming concepts from here .

Assignment: Write a MapReduce program to implement a WordCount application.

Click here for solving the exercise.

Estimated Time: 150 minutes

Learn about **Combiners** in MapReduce from here .

Assignment : Modify the above assignment by including the Combiner class

Click here for solving the exercise.

Estimated Time: 60 minutes

Learn about **Partitioners** in MapReduce from here.

Assignment : Write a MapReduce program that makes use of partitioners to optimize performance .

Click here for solving the exercise.

Estimated Time: 120 minutes

Case Study: Implement a retail use case scenario using MapReduce.

Click here for solving the case study.

Estimated Time: 4 Hours

4. Data Analysis using Hive

Hive is the ecosystem tool in Hadoop used for structured data analysis. Learn about Hive from here .

Creation and modification of Databases in Hive : Try the sample queries for the below database operations from here

a. Show Databases
b. Create Database
c. Describe Database
d. Use Database
e. Drop Database

Download the sample data sets for Hive assignments from here.

Assignment**:** Working with databases in Hive.

Click here for solving the exercise.

Estimated Time: 60 minutes

Create and Load data to Hive tables : Try the sample queries for the below table operations from here

a. Create Table - using various Hive data types
b. Describe Table
c. Show Tables
d. Drop Table
e. Loading Data to the table

Assignment: Create and manipulate hive tables.

Click here for solving the exercise.

Estimated Time: 60 minutes

Altering Tables and Databases : Try the sample queries for the below table and database operations from here .

a. Modify database properties
b. Modify table properties
c. Change column name
d. Change column type
e. Rename column

Assignment: Create and manipulate database and table properties

Click here for solving the exercise.

Estimated Time: 60 minutes

Table Manipulation commands : Try the sample queries for the below table manipulations from here.

a. Select queries
b. Joins operations
c. Index creation
d. Working with views

Assignment: Create and manipulate indexes and views in Hive

Click here for solving the exercise.

Estimated Time: 60 minutes

Creation and usage of partitions & Buckets in Hive : Try the sample queries for the below operations on tables from here .

a. Creating Static partitions
b. Creating Dynamic partitions
c. Listing partitions
d. Creating buckets

Assignment: Create and manipulate partitions & buckets

Click here for solving the exercise.

Estimated Time: 120 minutes

Creating custom UDFs : Try the sample scripts available here to create Custom UDFs.

a. Create a UDF with Eclipse and Java

Assignment: Create a custom UDF in Hive

Click here for solving the exercise.

Estimated Time: 60 minutes

Case Study : Implement the real-life use case described here using Hive.

Estimated Time: 4 hours

5. Data Analysis using Pig

You can access the Pig study materials which is part of Hadoop Framework Part 2 from http://icp.ad.infosys.com/project/OS_TVMZ/Hadoop/Forms/AllItems.aspx

Create a sample file with the below contents to test PigLatin commands.

Dataset: student.tsv

1001	John	45.0
1002	James	85.0
1003	John	45.0
1004	James	85.0
1005	Smith	60.0
1006	Scott	70.0
1007	Shoba	80.0
1008	Taanu	90.0
1009	Anbu	95.0
1010	Aruna	85.0

Dataset: department.tsv

1001	101	B.E
1002	102	B.Tech
1003	103	M.Tech
1004	101	B.E
1005	103	M.Tech
1006	103	M.Tech
1007	101	B.E
1008	102	B.Tech
1009	101	B.E
1010	102	B.Tech

Working with Pig's grunt shell: Load the datasets in HDFS and connect to grunt shell.

a. Copy the sample files created to the HDFS directory named /user/LabOnCloud/

```
hadoop fs  -put  student.tsv  /user/LabOnCloud/emno_name/student.tsv
hadoop fs  -put  department.tsv
/user/LabOnCloud/emno_name/department.tsv
```

b. Access grunt shell by typing the below command from the prompt

```
$>   pig  –x
```

Each processing step in Pig results in a new data set, or relation.

Working with Basic operators: Basic operators in Pig are – Load, Store and Dump.

a. Load: To specify the input data to be processed

```
A = load 'hdfsdir/student.tsv' as (rollno:int, name:chararray, gpa:float);
```

b. Store: To store the processed data or relation by a name

```
A = load 'student.tsv' as (rollno:int, name:chararray, gpa:float);
store A into 'result';
```

c. Dump: To display the contents of a relation on screen

```
Dump A;
```

Assignment: Create a sample data set HDFS and load it to a Pig relation.

Store it in a specific name and display the contents on screen

Estimated Time : 45minutes

Working with Relational operators: Relational operators in Pig allow you to transform datasets by sorting, grouping, joining, projecting, and filtering.

a. filter: to filter records based on a condition

```
A = load 'student.tsv' as (rollno:int, name:chararray, gpa:float);
B = filter A by gpa > 80.0;
C = filter B BY rollno > 1004;
DUMP B;
```

b. foreach : to apply an operation on every elements in a relation

```
A = load 'student.tsv' as (rollno:int, name:chararray, gpa:float);
B = foreach A generate UPPER (name);
```

c. distinct : to display unique elements in a relation

```
A = load 'student.tsv' as (rollno:int, name:chararray, gpa:float);
B = DISTINCT A;
DUMP B;
```

d. limit: to limit the number of results listed

```
A = load 'student.tsv' as (rollno:int, name:chararray, gpa:float);
B = LIMIT A 3;
DUMP B;
```

e. order by: to sort the dataset in a relation

```
A = load 'student.tsv' as (rollno:int, name:chararray, gpa:float);
B = ORDER A BY name;
DUMP B;
```

f. join: to join two relations

```
A = load 'student.tsv' as (rollno:int, name:chararray, gpa:float);
B = load 'department.tsv' as (rollno:int,
        deptno:int,deptname:chararray);
C = JOIN A BY rollno, B BY rollno;
DUMP C;
```

g. union : to get Union of two relations

```
A = load 'student.tsv' as (rollno, name, gp);
B = load 'department.tsv' as (rollno, deptno,deptname);
C = UNION A,B;
DUMP C;
```

h. split : to split a relation into multiple relations based on certain conditions

```
A = load 'student.tsv' as (rollno:int, name:chararray, gpa:float);
SPLIT A INTO X IF gpa<50, Y IF (gpa >=50 AND gpa < 75), Z IF
        (gpa >= 75 OR name == 'Smith');
DUMP X;
DUMP Y;
```

Assignment: Create the below sample data (NYSE_Dividends.txt) and load it in a Pig relation.:

NYSE_Dividends.txt

Schema - exchange:chararray, symbol:chararray, date:chararray, dividends:float

NYSE	CPO	2009-12-30	0.14
NYSE	CPO	2002-09-28	0.14
NYSE	CPO	2009-06-26	0.14
NYSE	CPO	2004-03-27	0.14
NYSE	CPO	2009-01-06	0.14
NYSE	CCS	2004-10-28	0.414
NYSE	CCS	2009-07-29	0.414
NYSE	CCS	2003-04-29	0.414
NYSE	CCS	2009-01-28	0.414
NYSE	CIF	2009-12-09	0.029
NYSE	CIF	2006-11-10	0.019
NYSE	CIF	2009-10-13	0.019
NYSE	CIF	2003-09-10	0.019
NYSE	CIF	2009-08-10	0.02
NYSE	CIF	2008-07-13	0.02

Write pig scripts to do the listed operations

a. List the unique symbols in the given dataset

b. Split the dataset into two relations based on the value of dividends (relation1 – dividends>0.1, others - <0.1)

c. Sort the data based on the symbol

d. List the count of records in each symbol

Estimated Time: 120 minutes

Working with diagnostic operators: Diagnostic tools and operators make Pig development easy.

a. describe: to show the schema of a relation

```
A = load  'student.tsv' AS (name:chararray, age:int, gpa:float);
DESCRIBE A;
```

b. explain: to explain the Physical plan and Logical Plan of the script

```
A = load 'student.tsv' as (rollno:int, name:chararray, gpa:float);
B = GROUP A BY gpa;
DUMP B;
EXPLAIN B;
```

c. illustrate: to debug your code easily, it takes a sample of your data and runs your script on it.

```
A = load 'student.tsv' as (rollno:int, name:chararray, gpa:float);
B = GROUP A BY gpa;
DUMP B;
ILLUSTARTE B;
```

Assignment: Use the below diagnostic tools on the scripts generated in the previous assignment and analyze the result.

a. Describe
b. illustrate
c. explain

Estimated Time: 60 minutes

Working with Parameter Substitution: Helpful in executing scripts which has elements that need to change based on certain conditions. For example, a script that runs every day is likely to have a date component in its input files or filters. To avoid editing the script every day to change the date, parameter substitution can be used.

a. Create a pig script named parameterdemo.pig as below

```
A = load '$student' as (rollno:int, name:chararray, gpa:float);
DUMP A;
```

b. Execute the code using parameter substitution from the prompt. Here student is the parameter.

```
$>  pig –param  student =/pigdemo/student.tsv  parameterdemo.pig
```

Assignment: Write a pig script that makes use of parameter substitution to load different datasets used in the assignments (students.tsv. department,tsv , NYSE_Dividends.txt) . Use the dataset name as the parameter value while invoking the script.

Estimated Time : 60 minutes

6. Data ingestion using Sqoop

Sqoop is the data ingestion tool used to transfer data from RDBMS to HDFS. Here we are using Sqoop to transfer data from the MySQL database available in the cluster. User name and password for MySQL will be shared.

Sqoop course artifacts are available as part of Hadoop Framework Part 2 at http://icp.ad.infosys.com/project/OS_TVMZ/Hadoop/Forms/AllItems.aspx

Sample username and password is used in the scripts demo purpose.

Credentials for MySQL - user name : root , password : Infy@123

Table creation Connect to MySQL and create the below tables.

Tables : products, employee, department

Schema for products: [productId:int (primary key) , productcode:char(3), name:varchar2(30), qualtity:int, price:float]

Schema for employee: [empno: int (primary key), deptno :int (foreign key – referencing deptno of department), empname varchar(20),designation varchar(20)]

Schema for department:[deptno: int (primary key), deptname varchar (20)]

Data Insertion: Insert some sample records in to the tables created.

Assignment: Create the below database and tables in MySQL . Execute sqoop commands to perform the list of operations mentioned.

a. Create a database named TestDB
b. Create a table named EmployeeTab in the TestDB with fields emp_id, fname, lname, state, salary (Use appropriate data types)

Estimated Time: 60 minutes

Sqoop tools : Sqoop provides a set of tools to import and export data between RDBMS and Hadoop. Connect to the Hadoop cluster using credentials provided and execute the sqoop commands from the prompt.

1. list-databases: lists the databases available in the MySQL database

```
sqoop list-databases --connect \
jdbc:mysql://vimsmys-40:3306/ \
--username root \
--password sqlpwd
```

2. list-tables: Lists all the tables

```
sqoop list-tables \
--connect jdbc:mysql://<<hostname>>:<<port>>/<<dbname>> \
 --username root \
--password sqlpwd
```

3. sqoop import : Imports a table from the MySQL database

```
sqoop import \
--connect jdbc:mysql://<<hostname>>:<<port>>/<<dbname>> \
--username root \
--password sqlpwd \
--table employee
```

4. Import with target-dir option (it will create the target directory inside HDFS)

```
sqoop import  \
--connect jdbc:mysql://<<hostname>>:<<port>>/<<dbname>> \
--username root \
--password sqlpwd \
--table productstab \
--target-dir Myproductstab
```

5. warehouse-dir : Imports table data to a parent directory specified using the warehouse-dir option

```
sqoop import  \
--connect jdbc:mysql://<<hostname>>:<<port>>/<<dbname>> \
--username root \
--password sqlpwd \
--table products  \
--warehouse-dir parentDir
```

6. –m : import using –m option to set the number of mappers for imports

```
sqoop import \
--connect jdbc:mysql://<<hostname>>:<<port>>/<<dbname>>  \
--username root \
--password Infy@123 \
--table products \
-m 1 \
```

7. --query : Import with a free form query and where clause

```
sqoop import \
--connect jdbc:mysql://<<hostname>>:<<port>>/<<dbname>> \
--username hiveuser \
--password 1234 \
--query 'select productID,productCode,name,quantity,price from products where productID < 1003 AND $CONDITIONS' \
-m 1 \
--target-dir /user/LabOnCloud/productsquery1
```

8. --direct : Direct Connector option

```
sqoop import \
--connect jdbc:mysql://<<hostname>>:<<port>>/<<dbname>> \
--username hiveuser \
--password 1234 \
--query 'select  productID,productCode,name,quantity,price from products where productID < 1004 AND $CONDITIONS' \
-m 1 \
--direct \
--target-dir /user/LabOnCloud/productsquery1
```

9. sqoop eval : Inserting records with sqoop eval

```
sqoop eval --connect jdbc:mysql://<<hostname>>:<<port>>/<<dbname>> \
--username hiveuser \
--password 1234 \
-e "insert into products values(1007, 'PEC', 'Pencil NB',0,99.99)"
```

10. --hive-table : Import data from mysql into Hive:

```
sqoop import \
--connect jdbc:mysql://<<hostname>>:<<port>>/<<dbname>> \
--username root \
--password sqlpwd \
--table department \
--direct \
-m 1 \
--hive-import \
--create-hive-table \
--hive-table mysql_sqoop_example
```

11. export : Export data from HDFS to MySQL

```
sqoop export \
--connect jdbc:mysql://<<hostname>>:<<port>>/<<dbname>> \
--username root \
--table employee_export \
--export-dir /user/LabOnCloud/employee \
--batch
```

Assignment: Use the database and table created in the above assignment for the below sqoop operations.

a. List the databases present in MySQL.
b. Import the table EmployeeTab present in RDBMS to a given HDFS path. (/user/LabOnCloud/SqoopDemos)
c. Import only the details of those employees whose salary is greater than 45000
d. Write the sqoop command which imports the EmployeeTab table data from RDBMS to HDFS
e. Import the EmployeeTab table from RDBMS into Hive
f. Import the EmployeeTab table from RDBMS to /user/LabOnCloud/SqoopDemos path in HDFS using 3 mappers.
g. Using eval tool in sqoop, select only first 3 rows of the EmployeeTab table in RDBMS.
h. Using eval tool in Sqoop, insert a new row into the EmployeeTab table in RDBMS
i. Export selected columns of EmployeeTab to MySQL database from HDFS.

Estimated Time: 120 minutes