

1] Demonstration - Understanding MapReduce

- 1) Download attachment in **STAGING_AREA**
- 2) Pull it in **LABS_HOME/demos**
- 3) Open the lab manual
- 4) Wait for the instructions from an Instructor

Dataset:-

- data/constitution.txt

2] Writing down WordCount Program

- 1) Download the rar in **STAGING_AREA** location
- 2) Extract it
- 3) Pull the extracted folder in LABS_HOME/wordcount
- 4) Wait for the instructions from an Instructor

Code/Dataset:-

- data/wordcount.rar

3] Distributed GREP

Here attached the rar file. Perform following activities

- 1) Download the rar file in **STAGING_AREA**, extract it
- 2) Create a folder grep in **LABS_HOME**. Import the java files from step 1 to this folder
- 3) Create a project **grep** in Eclipse
- 4) Import the java file from step 2 in the same
- 5) Resolve the compilation errors by adding hadoop client side libraries in class path of the project
- 6) Implement all **TODOs** in the code
- 7) Look at the comments at the end of the file and create a jar and execute it
- 8) During implementation of the code refer to the API documentation.
- 9) What differences are you finding in this code and previous WordCount code?

Time Line = 45 Mins

Code/Dataset:-

- data/grep.rar
-

4] Inverted Index

Here attached the rar file. Perform following activities

- 1) Download the rar file in **STAGING_AREA** location
- 2) Create a folder invertedindex in **LABS_HOME**.
- 3) Create a project **invertedindex** in Eclipse
- 4) Import the java file in the same
- 5) Resolve the compilation errors by adding hadoop client side libraries in class path of the project

- 6) Implement all TODOs in the code
- 7) Look at the comments at the end of the file and create a jar and execute it
- 8) During implementation of the code refer to the API documentation.
- 9) Have you seen such types of applications?

Time Line = 45 Mins

Code/Dataset:-

- data/InvertedIndex.rar

Lab 5] Implementing Counter

- 1] Download an attachment in **STAGING_AREA** location
- 2] Extract it
- 3] Pull it to **LABS_HOME/counter**
- 4] Study the code
- 5] Create **counties** folder in your home on HDFS
- 6] Put all **counties_*.csv** files from local linux file system to **counties** folder on HDFS
- 7] Run the application with command as **yarn jar average.jar <Main Class>**
- 8] See the output of the job on terminal window and confirm whether counters you created are working or not

Dataset:- data/Average.rar

Time Line = 45 Mins

6] Compression

- 1) Download the rar file in **STAGING_AREA**
- 2) Extract it
- 3) Pull the extracted folder in **LABS_HOME**
- 4) Create a **logfiles** folder on HDFS and put the log files from project folder into it
- 5) Create an eclipse project **compression**
- 6) Import the java files from project folder in **STAGING_AREA** in to this project
- 7) Remove the compilation errors by adding hadoop client side libraries
- 8) Create a jar **compression.jar**
- 9) Have a first Run using **yarn jar compression.jar INFO** command
- 10) Wait for the instructions from the instructor

Code/Dataset:-

- data/Compression.rar

7] Streaming Job

- 1) Download RAR in **STAGING_AREA**
- 2) Extract it
- 3) Pull the extracted folder in **LABS_HOME**
- 4) Wait for the instructions from an Instructor

Code/Dataset:-

- data/MRStreaming.rar