

**Programming Assignement-2**

Readme file



Riten chhatrala- [rchhatarala@hawk.iit.edu](mailto:rchhatarala@hawk.iit.edu)

Preyang shah- [pshah115@hawk.iit.edu](mailto:pshah115@hawk.iit.edu)

**Shared-Memory:**

You can see many folders containing scripts, screenshots, Source code etc. under Shared\_128GB folder.

You can find the Shared Memory external code under Shared\_sort\_128GB and Shared\_sort\_1TB. Below are the description of the files.

1. **ExternalSortMultiThreaded.java**

This file has main method which calls two methods 1) divideSorAndSave and 2) mergeAllFiles. In first phase i.e divideSorAndSave we are dividing single input file into small chunks and then each thread will call DivideSortAndSave class which will sort that chunk using merge sort and store sorted chunk data into temporary files and store those temporary file into global list which we will use in mergeAllFile phase.

In Second Phase we are mergin all Temporary file in single go. We are opening all the files and start reading then and merging them into single file.

1. **DivideSortAndSave.java**

This class extends Thread class to support multithread functionality. In this file we implemented merge file which can run by multiple thread.

You can find screenshot under Shared\_sort\_128GB folder.

**#################Single Node Setup(128GB) ##################**

**To run the Program:**

1. You need to log in to amazon AWS instance and search for EC2. Click on EC2 and it will open the EC2 Dashboard in which you need to click on launch instance button.
2. you need to select the OS first. choose "**ubuntu**". Click on the next and you are in instance page where you need to choose instance type. As per instruction given in assignment choose **i3.large** instance. You can see pop up for adding public key after click on several next buttons, so create new key and download it to your local drive. Click on launch instance button to launch the instance. (we do not require to change anything else as our assignment-2 satisfies within above steps, no further configuration needed to add in instance.)
3. Now write name of your instance and you can see status "**running**" for instance. Now click on connect and you can find "**ssh command**" written at the end just copy that part. Now open command prompt and go to the path where you saved your public key. Then you just paste the command you copied from EC2 dashboard.
4. Now you need to write **sudo su** command to go to root. Now using scp command copy all files (attached in Shared\_sort\_128GB folder) from the Shared\_sort\_128GB folder to i3.large instance.
5. Now write "**ls**" in the instance and you can see script and all the files you copied form the Shared\_sort\_128GB folder. Give permission to all the file except .pem key you generated while creating instance. Using "**chmod 777 file\_name**" you can give permission to the file.
6. Now run script **./install\_hadoop.sh** and it will download **java version 8** and **hadoop 2.7.1** to the instance. you need to click enter when script ask you about RSA generation. After pressing enter script start running again and after completing it you need to write "**source ~/.bashrc**" .
7. After that write "**hadoop --version**" to check hadoop is installed or not. Using that command, you can see the version of hadoop which is 2.7.1.
8. Now run the **./genrate\_raid.sh** script which mount data to the disk. after mounting the data, you can call gensort to create 128GB data. you need to follow below command.

**./gensort -a 1374389534(128gb) /mnt/raid/input\_128gb**(file name you want to generate data in).

9. edit and add below two lines and relogging back to command prompt.

**\* hard nofile 65536**

**\* soft nofile 65536**

10. After relog in back to putty. you need to run below command.

**ulimit -n 65536**

11. While program in merge phase it will open more than 1000 files from temp and it will show error like "too many files open". To overcome this Problem, we are enhancing this file limit to 65536.

12. Now run the below command.

**java -Xms256m -Xmx29000m ExternalSort.jar input\_128Gb output\_128GB 2(NoOfThreads)**

13.In above command you can set minimum and maximum limit of memory and it have 3 parameter. First parameter should be input file, second parameter should be output file and third parameter is the number of threads. Number of threads is used in program to use parallelization concept for multicore systems. We put 2 as number of thread as our i3.large instance has 2VCPU.

14. Now run the **./valsort /mnt/raid/output\_128GB** to valsort the output data and it will give result as sorted data.

You can find screenshot under Shared\_sort\_1TB folder.

**#####################Single Node Setup(1TB) #######################**

**To run the Program:**

1. You need to log in to amazon AWS instance and search for EC2. Click on EC2 and it will open the EC2 Dashboard in which you need to click on launch instance button.
2. you need to select the OS first. choose "**ubuntu**". Click on the next and you are in instance page where you need to choose instance type. As per instruction given in assignment choose **i3.large** instance. You can see pop up for adding public key after click on several next buttons, so create new key and download it to your local drive. Click on launch instance button to launch the instance. (we do not require to change anything else as our assignment-2 satisfies within above steps, no further configuration needed to add in instance.)
3. Now write name of your instance and you can see status "**running**" for instance. Now click on connect and you can find "**ssh command**" written at the end just copy that part. Now open command prompt and go to the path where you saved your public key. Then you just paste the command you copied from EC2 dashboard.
4. Now you need to write **sudo su** command to go to root. Now using scp command copy all files (attached in Shared\_sort\_1TB folder) from the Shared\_sort\_1TB folder to i3.4xlarge instance.
5. Now write "**ls**" in the instance and you can see script and all the files you copied form the Shared\_sort\_1TB folder. Give permission to all the file except .pem key you generated while creating instance. Using "**chmod 777 file\_name**" you can give permission to the file.
6. Now run script **./install\_hadoop.sh** and it will download **java version 8** and **hadoop 2.7.1** to the instance. you need to click enter when script ask you about RSA generation. After pressing enter script start running again and after completing it you need to write "**source ~/.bashrc**" .
7. After that write "**hadoop --version**" to check hadoop is installed or not. Using that command, you can see the version of hadoop which is 2.7.1.
8. Now run the **./genrate\_raid.sh** script which mount data to the disk. after mounting the data, you can call gensort to create 1TB data. you need to follow below command.

**./gensort -a 10995116227(1TB) /mnt/raid/input\_1TB**(file name you want to generate data in).

9. edit and add below two lines and relogging back to command prompt.

**\* hard nofile 65536**

**\* soft nofile 65536**

10. After relog in back to putty. you need to run below command.

**ulimit -n 65536**

11. While program in merge phase it will open more than 1000 files from temp and it will show error like "too many files open". To overcome this Problem, we are enhancing this file limit to 65536.

12. Now run the below command.

**java -Xms256m -Xmx29000m ExternalSort.jar input\_1Tb output\_1TB 2(NoOfThreads)**

13.In above command you can set minimum and maximum limit of memory and it have 3 parameter. First parameter should be input file, second parameter should be output file and third parameter is the number of threads. Number of threads is used in program to use parallelization concept for multicore systems. We put 2 as number of thread as our i3.large instance has 16VCPUs.

14. Now run the **./valsort /mnt/raid/output\_1TB** to valsort the output data and it will give result as sorted data.

**Hadoop:**

You can see many folders containing scripts, screenshots, Source code etc. under Hadoop\_128GB folder.

You can find TerasortHadoop folder in the Hadoop\_128GB folder in which you can find code for sorting input data. The folder contains below files.

1) TeraHadoop.java - This file has the main class which called mapper and reducer class. You need to pass parameter here. You need to pass input parameter (input file with data) first and add output parameter (output file that you want to generate) as second parameter in this order. We called methods like mapperclass, combinerclass, reducerclass etc.

2) SortMap.java - This class is the mapper class which take first 10 bit of the input line from input file.

3) Sort Reducer.java - This class is the reducer class which takes output from the mapper and write it to output file.

You can find screenshot under Hadoop\_128GB folder.

**#################Single Node Setup(128GB) #################**

**To run the Program:**

1. You need to log in to amazon AWS instance and search for EC2. Click on EC2 and it will open the EC2 Dashboard in which you need to click on launch instance button.
2. you need to select the OS first. choose "**ubuntu**". Click on the next and you are in instance page where you need to choose instance type. As per instruction given in assignment choose **i3.large** instance. You can see pop up for adding public key after click on several next buttons, so create new key and download it to your local drive. Click on launch instance button to launch the instance. (we do not require to change anything else as our assignment-2 satisfies within above steps, no further configuration needed to add in instance.)
3. Now write name of your instance and you can see status "**running**" for instance. Now click on connect and you can find "**ssh command**" written at the end just copy that part. Now open command prompt and go to the path where you saved your public key. Then you just paste the command you copied from EC2 dashboard.

1. Now you need to write **sudo su** command to go to root. Now using scp command copy all files (attached in Hadoop\_128GB folder) from the Hadoop\_128GB folder to i3.large instance.
2. Now write "**ls**" in the instance and you can see script and all the files you copied form the Hadoop\_128GB folder. Give permission to all the file except .pem key you generated while creating instance. Using "**chmod 777 file\_name**" you can give permission to the file.
3. Now run script **./install\_hadoop.sh** and it will download **java version 8** and **hadoop 2.7.1** to the instance. you need to click enter when script ask you about RSA generation. After pressing enter script start running again and after completing it you need to write "**source ~/.bashrc**" .
4. After that write "**hadoop --version**" to check hadoop is installed or not. Using that command, you can see the version of hadoop which is 2.7.1.
5. Now run the **./genrate\_raid.sh** script which mount data to the disk. after mounting the data, you can call gensort to create 128GB data. you need to follow below command.

**./gensort -a 1374389534(128gb) /mnt/raid/input\_128gb**(file name you want to generate data in)

1. you can see data is generating or not using "**lsblk**" command. After generating data, you need to format namenode using below command.

"**hdfs namenode -format**" this command to format the namenode and after that write "start-all.sh" command to start namenode and datanode. now you need to write "**JPS**" to check if all things are running or not.

1. Now you need to write "**hadoop fs -put /mnt/raid/input\_128gb /input\_128gb**" file name to put generated data input file into hdfs.
2. After putting the input file in hdfs you can run the jar using below command to run the program.

**hadoop jar "TerasortHadoop.jar" /input\_128gb /output\_128gb**

1. After running the jar file, you can get all details about the hadoop job.
2. Now run this command to get output file outside hdfs.

**hadoop fs -get /output\_128gb /mnt/raid/output\_128gb**

1. Now run the **./valsort /mnt/raid/output\_128gb** to valsort the output data and it will give result as sorted data.

You can find screenshot for single-node(1TB) under Hadoop\_1TB folder.

**##################Single - Node Setup(1TB) ##################**

**To run the Program:**

1. You need to log in to amazon AWS instance and search for EC2. Click on EC2 and it will open the EC2 Dashboard in which you need to click on launch instance button.
2. you need to select the OS first. choose "**ubuntu**". Click on the next and you are in instance page where you need to choose instance type. As per instruction given in assignment choose **i3.4xlarge** instance. You can see pop up for adding public key after click on several next buttons, so create new key and download it to your local drive. Click on launch instance button to launch the instance. (we do not require to change anything else as our assignment-2 satisfies within above steps, no further configuration needed to add in instance.)
3. Now write name of your instance and you can see status "**running**" for instance. Now click on connect and you can find "**ssh command**" written at the end just copy that part. Now open command prompt and go to the path where you saved your public key. Then you just paste the command you copied from EC2 dashboard.

1. Now you need to write **sudo su** command to go to root. Now using scp command copy all file (attached in Hadoop\_1TB folder) from the Hadoop\_1TB folder to i3.large instance.
2. Now write "**ls**" in the instance and you can see script and all the files you copied form the Hadoop\_1TB folder. Give permission to all the file except .pem key you generated while creating instance. Using "**chmod 777 file\_name**" you can give permission to the file.
3. Now run script **./install\_hadoop.sh** and it will download **java version 8** and **hadoop 2.7.1** to the instance. you need to click enter when script ask you about RSA generation. After pressing enter script start running again and after completing it you need to write "**source ~/.bashrc**" .
4. After that write "**hadoop --version**" to check hadoop is installed or not. Using that command, you can see the version of hadoop which is 2.7.1.
5. Now run the **./genrate\_raid.sh** script which mount data to the disk. after mounting the data, you can call gensort to create 1TB data. you need to follow below command.

**./gensort -a 10995116227(1TB) /mnt/raid/input\_128gb**(file name you want to generate data in)

1. you can see data is generating or not using "**lsblk**" command. After generating data, you need to format namenode using below command.

"**hdfs namenode -format**" this command to format the namenode and after that write "start-all.sh" command to start namenode and datanode. now you need to write "**JPS**" to check if all things are running or not.

1. Now you need to write "**hadoop fs -put /mnt/raid/input\_1TB /input\_1TB**" file name to put generated data input file into hdfs.
2. After putting the input file in hdfs you can run the jar using below command to run the program.

**hadoop jar "TerasortHadoop.jar" /input\_1TB /output\_1TB**

1. After running the jar file, you can get all details about the hadoop job.
2. Now run this command to get output file outside hdfs.

**hadoop fs -get /output\_1TB /mnt/raid/output\_1TB**

1. Now run the **./valsort /mnt/raid/output\_1TB** to valsort the output data and it will give result as sorted data.

**####################Multi - Node Setup(1TB) ####################**

**To run the Program:**

1. You need to log in to amazon AWS instance and search for EC2. Click on EC2 and it will open the EC2 Dashboard in which you need to click on launch instance button.
2. you need to select the os first. choose "**ubuntu**" here. Click on next and you are in instance page where you need to choose instance type. As per instruction given in assignment choose **i3.large** instance. Press next and you can see how many instances you want to make change the value to
3. Now click next again and you can see pop up for adding public key after click on next button. Now you can use same key that you create before or you can create new key and download it to your local drive. click on launch instance button to launch the instance.(we do not require to change anything else as our assignment-2 needs satisfy within above steps, no further configuration needed to add in instance.)
4. Now write name of your instances one as **hadoop-master** and other as **hadoop-slave-1**,etc. and you can see "**running**" status for instances.

1. Now click on hadoop-master instance and click on connect. you can find "**ssh command**" written at the end just copy that part. Now open command prompt and go to the path where you saved your public key. Then you just paste the command you copied from EC2 dashboard. After that you need to go to the Hadoop-multinode folder in local drive and edit **hosts\_tmp** file and write private IP for all the instances one by one with its name and now coming back to command prompt.
2. Now you need to write sudo su command to go to root. Now using scp command copy all file from the Hadoop-multinode folder to i3.4xlarge hadoop-master instance.
3. Now write "ls" in the instance and you can see script and all the files you copied form the Hadoop-multinode folder. Give permission to all the file except .pem key you generated while creating instance. Using "**chmod 777 file\_name**" you can give permission to the file.
4. Now run script **./master.sh** and it will download java version 8 and hadoop 2.7.1 to all the instances. you need to click enter when script ask you about RSA generation. After pressing enter script start running again. It also creates password less ssh to the slave instances.
5. You can check the hadoop install or not in all instance by going to that instance using ssh hadoop-slave-1, etc. and it will take you to that slave instance.
6. After that write "hadoop --version" to check hadoop is installed or not. Using that command, you can see the version of hadoop which is 2.7.1.
7. Now run the ./genrate\_raid\_clusters.sh script in master instance which send genrate\_raid.sh file(which is used to mount data to the disk)to all the slave instance. after mounting the data, you can call gensort to create 1TB data. you need to follow below command.

**./gensort -a 10995116227(1TB) /input\_1TB**(file name you want to generate data in)

1. you can see data is generating using "**lsblk**" command. After generating data you need to format namenode using below command.

"**hdfs namenode -format**" is the command to format the namenode and after that write "start-all.sh" command to start namenode and datanode. now you need to write

"**JPS**" to check if all things are running or not. you can see namenode,resouce manager running on master instance while slave datanode,nodemanager in slave instance.

1. Now you need to make directory name /input using command **hadoop fs -mkdir /input.**
2. Now you need to put 8 copy of input file(created using gensort) to hdfs using below command.
3. "**hadoop fs -put /mnt/raid/input\_1TB /input/input\_1TB\_1**(you need to make 8 copies to do it 1 TB as 128\*8= 1024Gb= 1TB)".
4. After putting the input file in hdfs you can run the jar using below command to run the program. You need to pass input folder as input parameter.

**hadoop jar "TerasortHadoop.jar" /input /output\_1TB**

1. After running the jar file, you can get all details about the hadoop job. Now you can check one file using below 2 steps. you need to follow same steps for all output file. Now run this command to get output file outside hdfs.

**hadoop fs -get /output\_1TB /mnt/raid/output\_1TB**

1. Now run the **./valsort /mnt/raid/output\_1TB** to valsort the output data and it will give result as sorted data.

**Spark:**

You can see many folders containing scripts, screenshots, Source code etc. under Spark\_128GB folder.

You can find spark\_sort.txt file under spark folder in which you can find code for sort input data.

* In this text file we firstly take nanosecond from system to store current time in nanosecond.
* Then we are reading data from input file which is stored in hdfs.
* Then we are taking key as first 10 data of the line from the file.
* Then we do sorting on that key and iterate loop to store it in the output file.
* We are again calculating time in nanosecond.
* In last statement we are calculating the time with start and end time of program labeled as Elapsed Time.

You can see many folders containing scripts,screenshots,Source code etc. under Spark\_128GB folder.

**#####################Single Node Setup(128GB) ###################**

**To run the Program:**

1. You need to log in to amazon AWS instance and search for EC2. Click on EC2 and it will open the EC2 Dashboard in which you need to click on launch instance button.
2. you need to select the os first. choose "**ubuntu**" here. Click on the next and you are in instance page where you need to choose instance type. As per instruction given in assignment choose **i3.large** instance. You can see pop up for adding public key after click on next button so create new key and download it to your local drive. click on launch instance button to launch the instance.(we do not require to change anything else as our assignment-2 needs satisfies within above steps, no further configuration needed to add in instance.)
3. Now write name of your instance and you can see status "**running**" for instance. Now click on connect and you can find "**ssh command**" written at the end just copy that part. Now open command prompt and go to the path where you saved your public key. Then you just paste the command you copied from EC2 dashboard.
4. Now you need to write **sudo su** command to go to root. Now using scp command copy all file from the Spark\_128GB folder to i3.large instance.
5. Now write "**ls**" in the instance and you can see script and all the files you copied form the spark folder under Spark\_128GB folder. Give permission to all the file except .pem key you generated while creating instance. Using "**chmod 777 file\_name**" you can give permission to the file.
6. Now run script **./install\_hadoop.sh** and it will download java version 8 and hadoop 2.7.1 to the instance. you need to click enter when script ask you about RSA generation. After pressing enter script start running again and after completing it you need to write "**source ~/.bashrc**" .
7. After that write "hadoop --version" to check hadoop is installed or not. Using that command, you can see the version of hadoop which is 2.7.1.
8. Now run the ./spark\_hadoop.sh script to install spark. we used spark 2.2.0.You can check spark version using spark-submit --version. Scala can also have downloaded while you run this script. Now you need to make changes to 1 property of 1 files in spark. You need to make changes in the /opt/spark/conf/log4j.properties.template and change the log4j.rootCategory=INFO, console to the log4j.rootCategory=**ERROR**, console .
9. You need to check one more property for in /opt/spark/conf/spark-defaults.conf.template file or /opt/spark/conf/spark-defaults.conf file. you need to check last line **spark.local.dir/mnt/raid/spark\_temp** if it’s not seen in the file then you need to add it.
10. Now run the ./genrate\_raid.sh script which mount data to the disk. after mounting the data you can call gensort to create 128GB data. you need to follow below command.

**./gensort -a 1374389534(128gb) /input\_128gb**(file name you want to generate data in)

1. you can see data is generating or not using "**lsblk**" command. After generating data you need to format namenode using below command. "**hdfs namenode -forma**t" is the command to format the namenode and after that write "start-all.sh" command to start namenode and datanode. now you need to write "JPS" to check if all things are running or not.
2. Now you need to write "**hadoop fs -put /mnt/raid/input\_128gb /input\_128gb**" file name to put generated data input file into hdfs.
3. After putting the input file in hdfs you need to run the spark\_sort.txt file using below command.

**spark-shell -i spark\_sort.txt**

1. After running the file, you can get all details about time it takes to run the program in spark. After that you need to go out of spark-shell to get output and do valsort on output.
2. Now run this command to get output file outside hdfs.

hadoop fs -get /output\_128gb /mnt/raid/output\_128gb

1. Now run the **./valsort /mnt/raid/output\_128gb** to valsort the output data and it will give result as sorted data.

You can see many folders containing scripts,screenshots,Source code etc. under Spark\_1TB folder.

**####################Single Node Setup(1TB) ####################**

**To run the Program:**

1. You need to log in to amazon AWS instance and search for EC2. Click on EC2 and it will open the EC2 Dashboard in which you need to click on launch instance button.
2. you need to select the os first. choose "**ubuntu**" here. Click on the next and you are in instance page where you need to choose instance type. As per instruction given in assignment choose **i3.4xlarge** instance. You can see pop up for adding public key after click on next button so create new key and download it to your local drive. click on launch instance button to launch the instance.(we do not require to change anything else as our assignment-2 needs satisfies within above steps, no further configuration needed to add in instance.)
3. Now write name of your instance and you can see status "**running**" for instance. Now click on connect and you can find "**ssh command**" written at the end just copy that part. Now open command prompt and go to the path where you saved your public key. Then you just paste the command you copied from EC2 dashboard.
4. Now you need to write **sudo su** command to go to root. Now using scp command copy all file from the Spark\_1TB folder to i3.4xlarge instance.
5. Now write "**ls**" in the instance and you can see script and all the files you copied form the spark folder under Spark\_1TB folder. Give permission to all the file except .pem key you generated while creating instance. Using "**chmod 777 file\_name**" you can give permission to the file.
6. Now run script **./install\_hadoop.sh** and it will download java version 8 and hadoop 2.7.1 to the instance. you need to click enter when script ask you about RSA generation. After pressing enter script start running again and after completing it you need to write "**source ~/.bashrc**" .
7. After that write "hadoop --version" to check hadoop is installed or not. Using that command, you can see the version of hadoop which is 2.7.1.
8. Now run the ./spark\_hadoop.sh script to install spark. we used spark 2.2.0.You can check spark version using spark-submit --version. Scala can also have downloaded while you run this script. Now you need to make changes to 1 property of 1 files in spark. You need to make changes in the /opt/spark/conf/log4j.properties.template and change the log4j.rootCategory=INFO, console to the log4j.rootCategory=**ERROR**, console .
9. You need to check one more property for in /opt/spark/conf/spark-defaults.conf.template file or /opt/spark/conf/spark-defaults.conf file. you need to check last line **spark.local.dir/mnt/raid/spark\_temp** if it’s not seen in the file then you need to add it.
10. Now run the ./genrate\_raid.sh script which mount data to the disk. after mounting the data, you can call gensort to create 128GB data. you need to follow below command.

**./gensort -a 10995116227(1TB) /input\_1TB** (file name you want to generate data in)

1. you can see data is generating or not using "**lsblk**" command. After generating data you need to format namenode using below command. "**hdfs namenode -forma**t" is the command to format the namenode and after that write "start-all.sh" command to start namenode and datanode. now you need to write "JPS" to check if all things are running or not.
2. Now you need to write "**hadoop fs -put /mnt/raid/input\_1TB /input\_1TB**" file name to put generated data input file into hdfs.
3. After putting the input file in hdfs you need to run the spark\_sort.txt file using below command.

**spark-shell -i spark\_sort.txt**

1. After running the file, you can get all details about time it takes to run the program in spark. After that you need to go out of spark-shell to get output and do valsort on output.
2. Now run this command to get output file outside hdfs.

hadoop fs -get /output\_1TB /mnt/raid/output\_1TB

1. Now run the **./valsort /mnt/raid/output\_1TB** to valsort the output data and it will give result as sorted data.

**You can see many folders containing scripts,screenshots,Source code etc. under Spark\_multinode folder.**

**######################Multi - Node Setup(1TB) ######################**

**To run the Program:**

1. You need to log in to amazon AWS instance and search for EC2. Click on EC2 and it will open the EC2 Dashboard in which you need to click on launch instance button.
2. you need to select the os first. choose "**ubuntu**" here. Click on next and you are in instance page where you need to choose instance type. As per instruction given in assignment choose **i3.large** instance. Press next and you can see how many instances you want to make change the value to 9.
3. Now click next again and You can see pop up for adding public key after click on next button. Now you can use same key that you create before or you can create new key and download it to your local drive. click on launch instance button to launch the instance.(we do not require to change anything else as our assignment-2 needs satisfies within above steps, no further configuration needed to add in instance.)
4. Now write name of your instances one as spark-master and other as spark-slave-1,etc. and you can see "**running**" status for instances.
5. Now click on spark-master instance and click on connect. You can find "ssh command" written at the end just copy that part. Now open command prompt and go to the path where you saved your public key. Then you just paste the command you copied from EC2 dashboard.
6. After that you need to go to the spark-multinode folder in local drive and you can see **hosts\_tmp** file and in which you need to edit private IP for all the instances one by one with its name and now coming back to command prompt.
7. Now you need to write sudo su command to go to root. Now using scp command copy all file from the spark-multinode folder to i3.large spark-master instance.
8. Now write "ls" in the instance and you can see script and all the files you copied form the spark-multinode folder. Give permission to all the file except .pem key you generated while creating instance. Using "chmod 777 file\_name" you can give permission to the file.
9. Now run script ./master.sh and it will download java version 8 and hadoop 2.7.1 to all the instances. you need to click enter when script ask you about RSA generation. After pressing enter script start running again. It also creates password less ssh to the slave instances.
10. You can check the hadoop install or not in all instance by going to that instance using ssh spark-slave-1, etc. and it will take you to that slave instance.
11. After that write "hadoop --version" to check hadoop is installed or not. Using that command, you can see the version of hadoop which is 2.7.1.
12. Now run the spark\_hadoop.sh command in the spark-master instance and changes the 2 property files in spark-defaults.conf and log4j property file. After changing the property file

send /usr/local/spark/ folder in all spark-slave-1,2,3.

1. Run ./generate\_raid.sh script from all instance after mounting the data you can call gensort to create 1TB data. you need to follow below command.

**./gensort -a 10995116227(1TB) /input\_1TB**(file name you want to generate data in).

1. you can see data is generating using "lsblk" command. After generating data you need to format namenode using below command.

"hdfs namenode -format" is the command to format the namenode and after that write "start-all.sh" command to start namenode and datanode.

Now you need to write "JPS" to check if all things are running or not. you can see namenode,resouce manager running on master instance while slave datanode,nodemanager in slave instance.

1. Now you need to make directory name /input\_dir using command hadoop fs -mkdir /input\_dir.

"**hadoop fs -put /mnt/raid/input\_dir/input\_1TB /input\_dir/input\_1TB**.

1. After putting the input file in hdfs you can run the jar using below command to run the program. You do not need to pass any parameter as we included input filename as parameter in the spark\_Sort.txt file.

**spark-shell -i spark\_sort.text**

1. After running the above command, you can get all details about the Spark job details.
2. Now run this command to get output file outside hdfs.

**hadoop fs -get /output\_1TB /mnt/raid/output\_1TB**

1. Now run the **./valsort /mnt/raid/output\_1TB** to valsort the output data and it will give result as sorted data.