

# INSTALLATION AND COMMANDS

# WORD COUNTS EXAMPLE

# PDC Project: Hadoop

# Name: Muhammad Allah Rakha

# Roll No: P19-0006 (BCS-6A)

# Name: Muhammad Aamir Khan

# Roll No: P18-0121 (BCS-6B)

**Source Link**

**Website: https://aaaastark.herokuapp.com**

**GitHub: https://github.com/aaaastark**

**INTRODUCTION TO HADOOP**

**What is Hadoop?**

Hadoop is an open source software programming framework for storing a large amount of data and performing the computation. Its framework is based on Java programming with some native code in C and shell scripts.

* Apache Hadoop is an open source framework intended to make interaction with Big Data.
* Hadoop has made its place in the industries and companies that need to work on large data sets which are sensitive and needs efficient handling.
* Hadoop is a framework that enables processing of large data sets which reside in the form of clusters.

**History of Hadoop?**

**Apache Software Foundation** is the developers of Hadoop, and its co-founders are **Doug Cutting** and **Mike Cafarella**. In October 2003 the first paper release was Google File System. In January 2006, MapReduce development started on the Apache Nutch which consisted of around 6000 lines coding for it and around 5000 lines coding for HDFS. In April 2006 Hadoop 0.1.0 was released.

**Some common frameworks of Hadoop?**

* **Hive**- It uses HiveQl for data structuring and for writing complicated MapReduce in HDFS.
* **Drill**- It consists of user-defined functions and is used for data exploration.
* **Storm**- It allows real-time processing and streaming of data.
* **Spark**- It contains a Machine Learning Library (MLlib) for providing enhanced machine learning and is widely used for data processing. It also supports Java, Python, and Scala.
* **Pig**- It has Pig Latin, a SQL-Like language and performs data transformation of unstructured data.
* **Tez**- It reduces the complexities of Hive and Pig and helps in the running of their codes faster.

**Advantages and Disadvantages of Hadoop?**

**Advantages:**

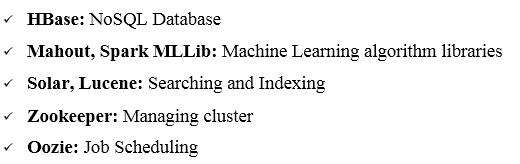
* Ability to store a large amount of data.
* High flexibility.
* Cost effective.
* High computational power.
* Tasks are independent.
* Linear scaling.

**Disadvantages:**

* Not very effective for small data.
* Hard cluster management.
* Has stability issues.
* Security concerns.

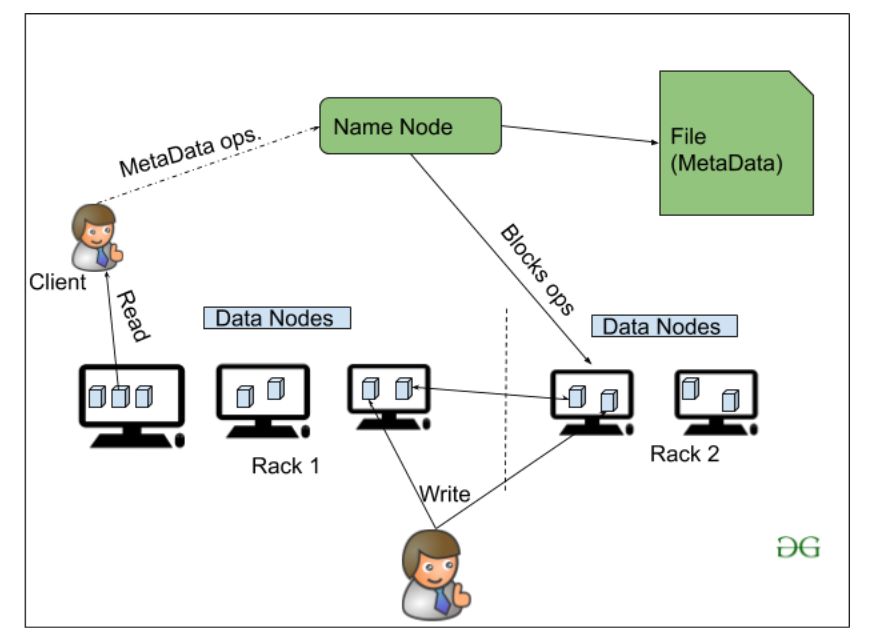
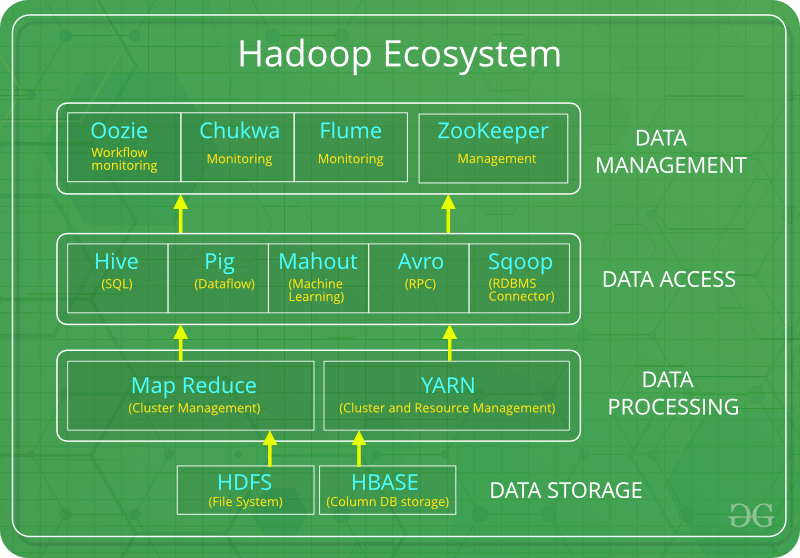
# Hadoop Ecosystem?

# Hadoop Ecosystem is a platform or a suite which provides various services to solve the big data problems.

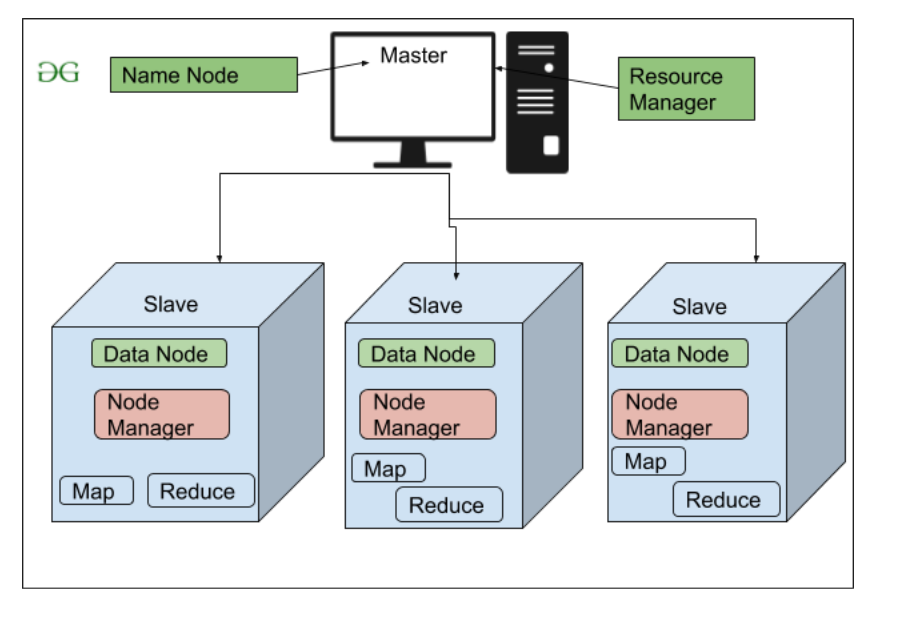
**Following are the components that collectively form a Hadoop ecosystem:**

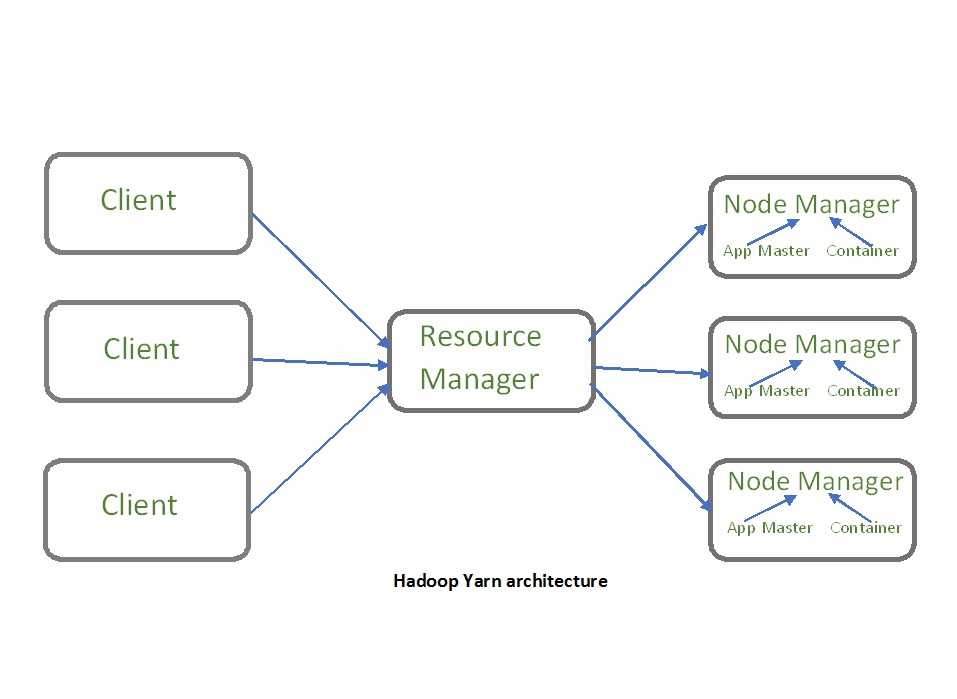
* **HDFS:** Hadoop Distributed File System
* **YARN:** Yet Another Resource Negotiator
* **MapReduce:** Programming based Data Processing
* **Spark:** In-Memory data processing
* **PIG, HIVE:** Query based processing of data services

All these tools work collectively to provide services such as absorption, analysis, storage and maintenance of data etc.

****

**HDFS:**

* HDFS is the primary or major component of Hadoop ecosystem and is responsible for storing large data sets of structured or unstructured data across various nodes and thereby maintaining the metadata in the form of log files.
* HDFS consists of two core components i.e.
  + Name node
  + Data Node
* Name Node is the prime node which contains metadata (data about data) requiring comparatively fewer resources than the data nodes that stores the actual data. These data nodes are commodity hardware in the distributed environment.

**YARN:**

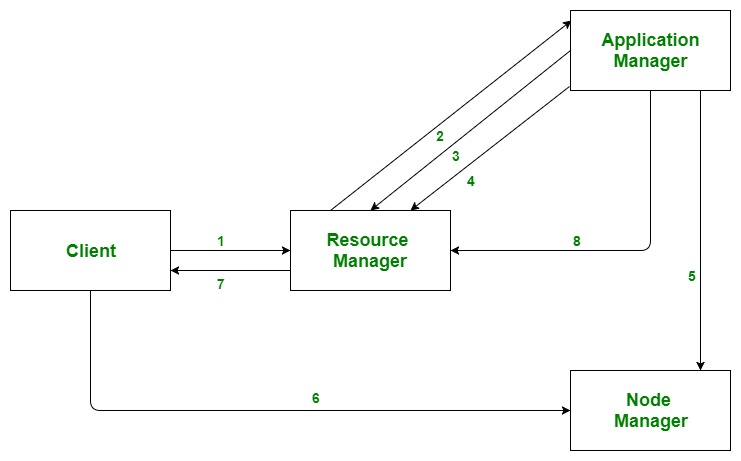
* Yet Another Resource Negotiator, as the name implies, YARN is the one who helps to manage the resources across the clusters. In short, it performs scheduling and resource allocation for the Hadoop System.
* Consists of three major components i.e.
  + Resource Manager
  + Nodes Manager
  + Application Manager
* Resource manager has the privilege of allocating resources for the applications in a system whereas Node managers work on the allocation of resources such as CPU, memory, bandwidth per machine and later on acknowledges the resource manager. Application manager works as an interface between the resource manager and node manager and performs negotiations as per the requirement of the two.

**Features of YARN:**

* **Multi-Tenancy:** It allows multiple engine access thus giving organizations a benefit of multi-tenancy.
* **Scalability:** The scheduler in Resource manager of YARN architecture allows Hadoop to extend and manage thousands of nodes and clusters.
* **Cluster-Utilization:** YARN supports Dynamic utilization of cluster in Hadoop, which enables optimized Cluster Utilization.
* **Compatibility:** YARN supports the existing map-reduce applications without disruptions thus making it compatible with Hadoop 1.0 as well.

**The main components of YARN architecture include:**

* **Client:** It submits map-reduce jobs.
* **Resource Manager:** Whenever it receives a processing request, it forwards it to the corresponding node manager and allocates resources for the completion of the request accordingly.
* **Node Manager:** Manages application and workflow and that particular node. It is also responsible for creating the container process and start it on the request of Application master. It registers with the Resource Manager and sends heartbeats with the health status of the node.
* **Application Master:** The application master is responsible for negotiating resources with the resource manager, tracking the status and monitoring progress of a single application. The application master requests the container from the node manager by sending a Container Launch Context(CLC) which includes everything an application needs to run. Once the application is started, it sends the health report to the resource manager from time-to-time.
* **Container:** It is a collection of physical resources such as RAM, CPU cores and disk on a single node. The containers are invoked by Container Launch Context(CLC) which is a record that contains information such as environment variables, security tokens, dependencies etc.

****

**MapReduce:**

MapReduce is a programming model used for efficient processing in parallel over large data-sets in a distributed manner. The data is first split and then combined to produce the final result.

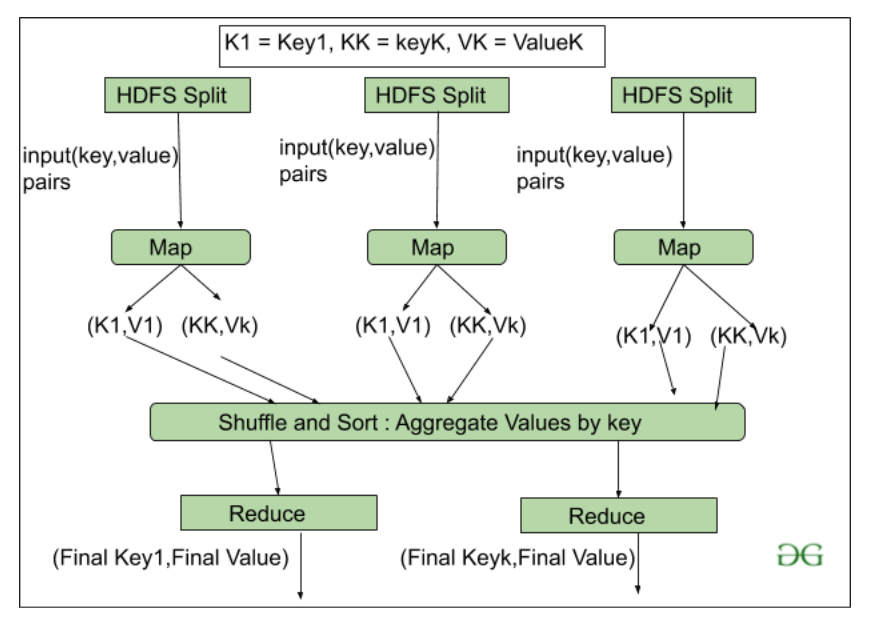
* Map(): Performs sorting and filtering of data and thereby organizing them in the form of group. Map generates a key-value pair based result which is later on processed by the Reduce() method.
* Reduce(): Takes the output generated by Map() as input and combines those tuples into smaller set of tuples.

**Let’s understand the Map Task and Reduce Task in detail.**

**Map Task:**

* **RecordReader** The purpose of *recordreader* is to break the records. It is responsible for providing key-value pairs in a Map() function. The key is actually is its locational information and value is the data associated with it.
* **Map:**  Is to process the Tuples obtained from record reader. The Map() function either does not generate any key-value pair or generate multiple pairs of these tuples.
* **Combiner:** Combiner is used for grouping the data in the Map workflow.
* Partitional is responsible for fetching key-value pairs generated in the Mapper Phases. Hashcode of each key is also fetched by this partition. Then partitioner performs it’s (Hashcode) modulus with the number of reducers (*key.hashcode()%(number of reducers)).*

**Reduce Task:**

* **Shuffle and Sort:** The Task of Reducer starts with this step, the process in which the Mapper generates the intermediate key-value and transfers them to the Reducer task is known as *Shuffling*. Using the Shuffling process the system can sort the data using its key value.
* **Reduce:** The main function or task of the Reduce is to gather the Tuple generated from Map and then perform some sorting and aggregation sort of process on those key-value depending on its key element.
* **OutputFormat:** Once all the operations are performed, the key-value pairs are written into the file with the help of record writer, each record in a new line, and the key and value in a space-separated manner.

**PIG:**

Pig was basically developed by Yahoo which works on a pig Latin language, which is Query based language similar to SQL.

It is a platform for structuring the data flow, processing and analyzing huge data sets.

**HIVE:**

HIVE performs reading and writing of large data sets. However, its query language is called as HQL (Hive Query Language).

It is highly scalable as it allows real-time processing and batch processing both. HIVE too comes with two components: ***JDBC******Drivers*** and ***HIVE******Command******Line***.

**Mahout:**

Mahout, allows Machine Learnability to a system or application. It provides various libraries or functionalities such as collaborative filtering, clustering, and classification which are nothing but concepts of Machine learning.

**Apache Spark:**

Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. The main feature of Spark is its **in-memory cluster computing** that increases the processing speed of an application. Spark is designed to cover a wide range of workloads such as batch applications, iterative algorithms, interactive queries and streaming.

**Apache HBase:**

It’s a NoSQL database which supports all kinds of data and thus capable of handling anything of Hadoop Database. It provides capabilities of Google’s BigTable, thus able to work on Big Data sets effectively.

**Solr, Lucene:**

These are the two services that perform the task of searching and indexing with the help of some java libraries, especially Lucene is based on Java which allows spell check mechanism, as well. However, Lucene is driven by Solr.

**Zookeeper:**

There was a huge issue of management of coordination and synchronization among the resources or the components of Hadoop which resulted in inconsistency, often.

Zookeeper overcame all the problems by performing synchronization, inter-component based communication, grouping, and maintenance.

**Oozie:**

Oozie simply performs the task of a scheduler, thus scheduling jobs and binding them together as a single unit. There is two kinds of jobs .i.e. **Oozie workflow** and **Oozie coordinator jobs**.

* Oozie workflow is the jobs that need to be executed in a sequentially ordered manner.
* Oozie Coordinator jobs are those that are triggered when some data or external stimulus is given to it.

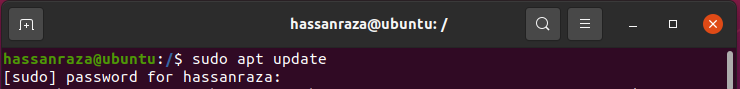
**INSTALLATION OF HADOOP**

# Admin User: hassanraza User Password: Hassan

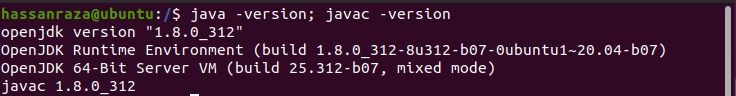
# Hadoop User: hdoop User Password: hdoop

# NOTE: Install the new Ubuntu using the Dual Boot or VMware.

# COMMANDS: Following these commands for Hadoop Installation.



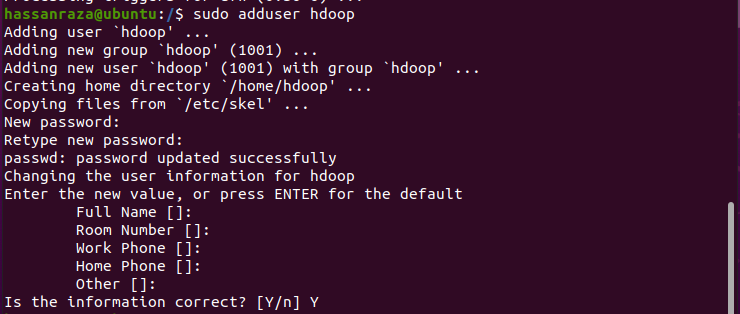
Install the Java JDK 8. Required to Hadoop installation process.

Check the version of Java and Java Compiler.

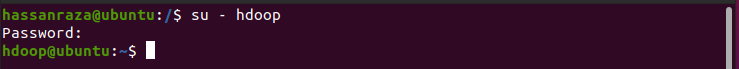
OpenSSH provides a large suite of secure tunneling capabilities, several authentication methods, and sophisticated configuration options. Secure Shell (SSH) protocol family of tools for **remotely controlling, or transferring files between, computers. Install for both Client and Server.**



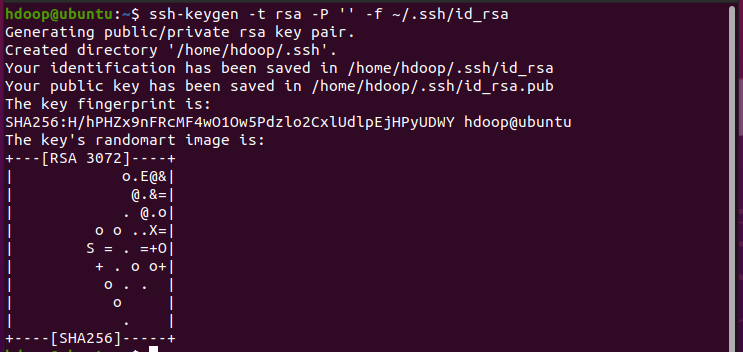
Add the new user to Hadoop Setup.



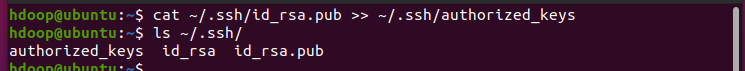
Change the Root user to Hadoop user.



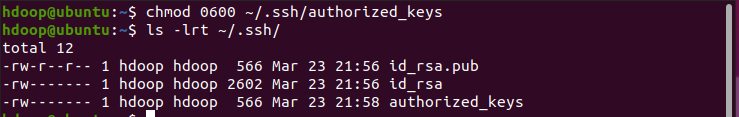
Generate the Key RSA then save into “.ssh” Directory. Two files (id\_rsa and id\_rsa.pub).

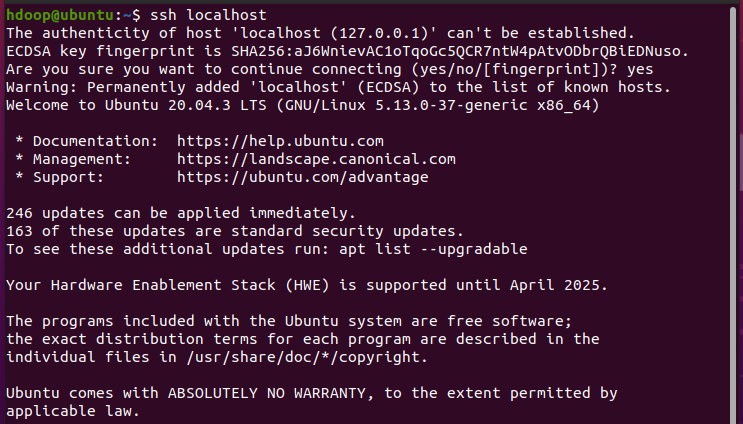


Create and Write the content of “id\_rsa.pub” key file into “authorized\_keys” file.



Change the Mode of key file “authorized\_keys”. Assign a “Read” and “Write” mode.



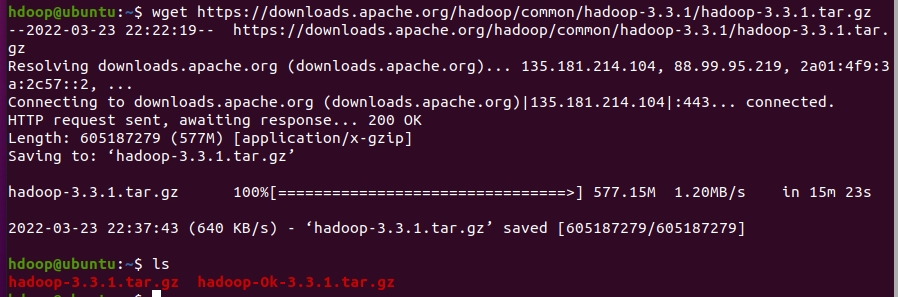
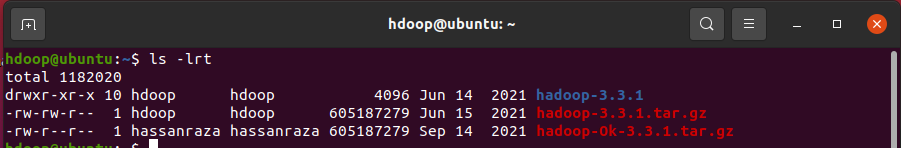
Open the Secure Shell (SSH) into local-host port.

## **First Way:**

If you have already download Hadoop file. Want to install into Hadoop Home Root (hadoop). Then follow these process. The download file placed on Home Directory (hassanraza). Move Directory (hassanraza) to Directory (hdoop).

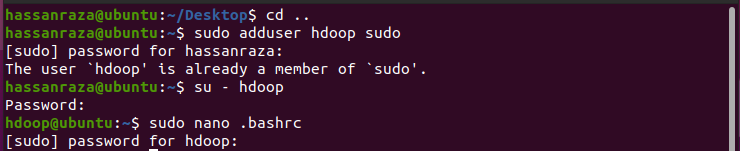


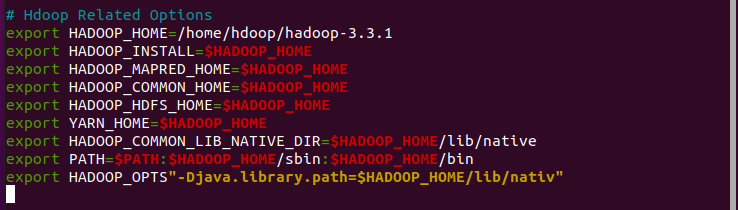
## **Second Way:**

Then follow these process for Hadoop download Online. The download file placed on Home Directory (hdoop).

Allow the “sudo” permission to Hadoop Home Root (hdoop). Then edit the “.bashrc” file for Global/Local paths variables. Here export Hadoop and Java paths (installation and access location etc). Assign the code at end of file.

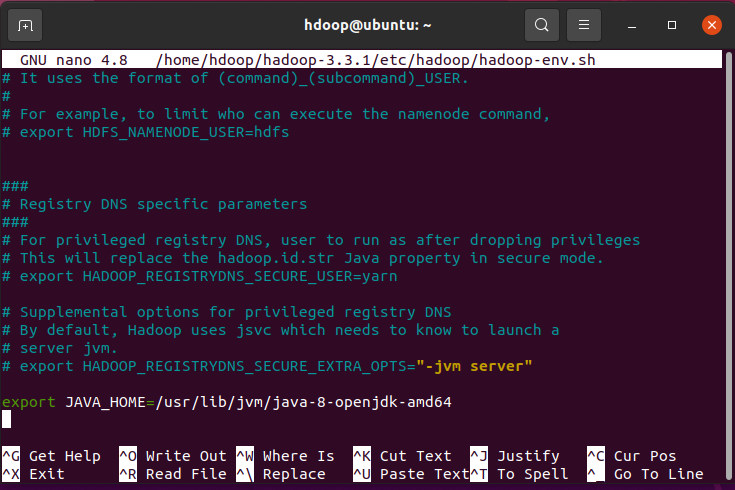
Some changing into file, then these changing make available for execution to OS Kernel. Using “source ~/.bashrc”



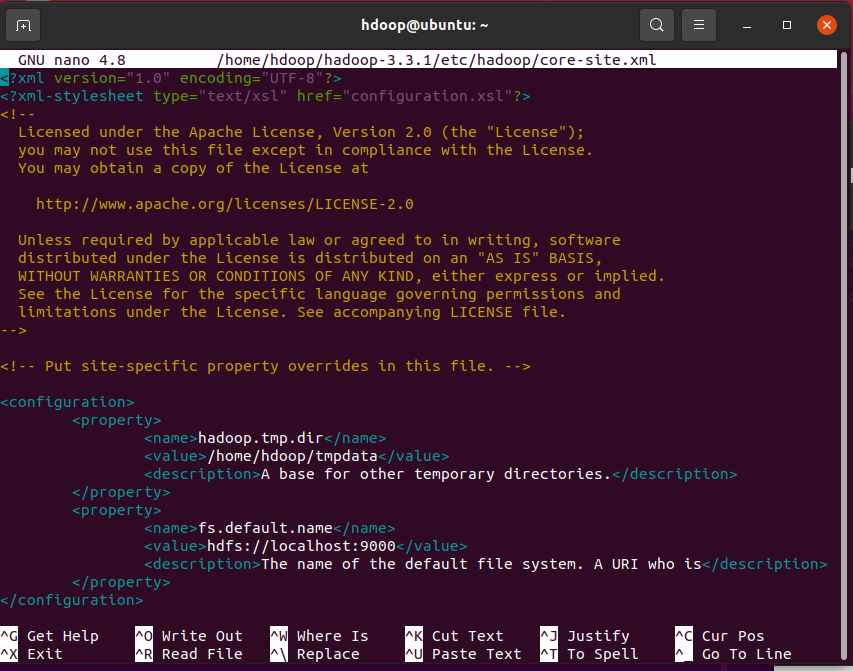




Add the path of Java JDK (JAVA\_HOME) into Hadoop Environment shell file.



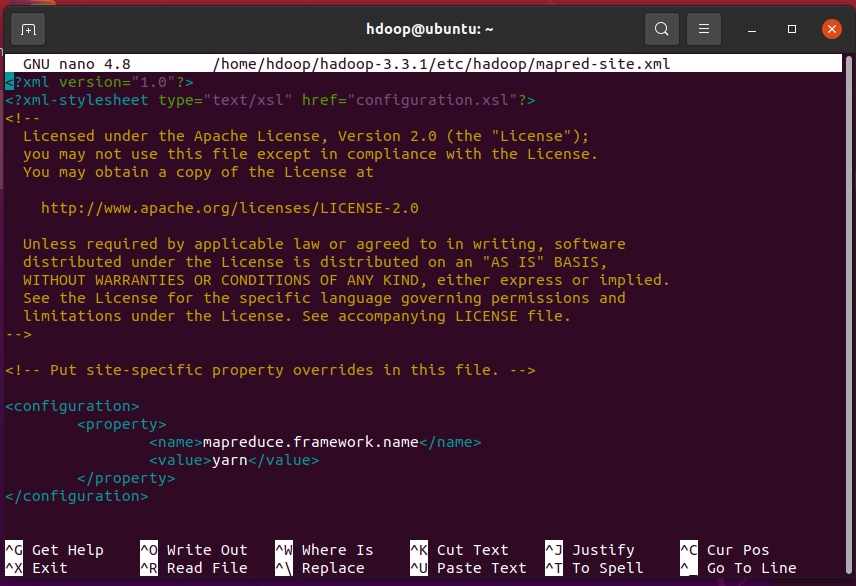
**1: Core-Site.xml**



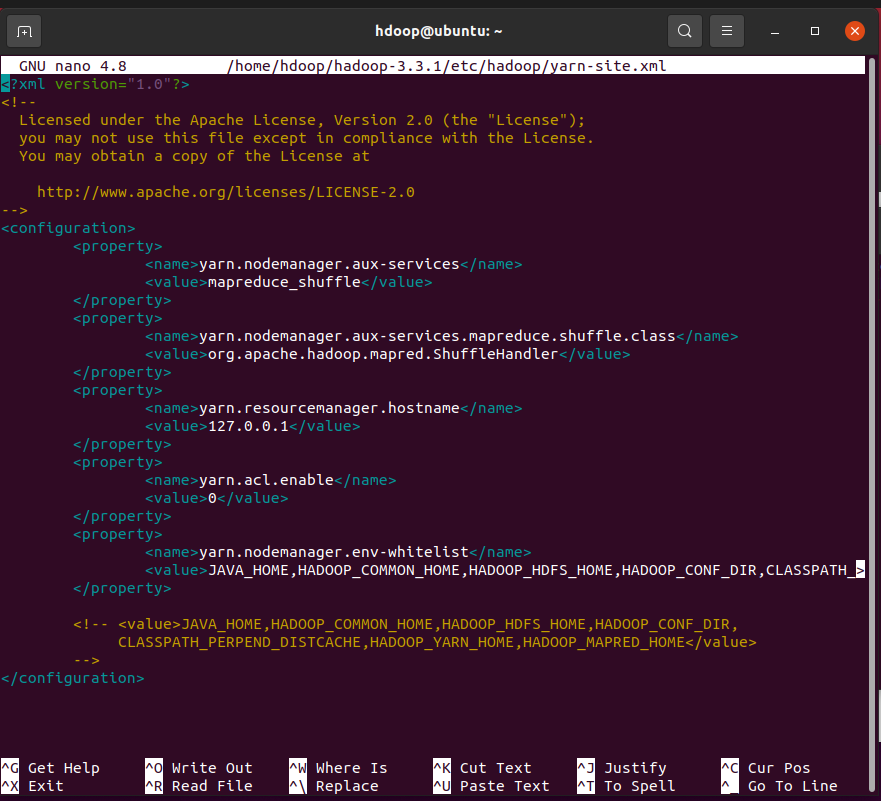
**2: hdfs-site.xml**

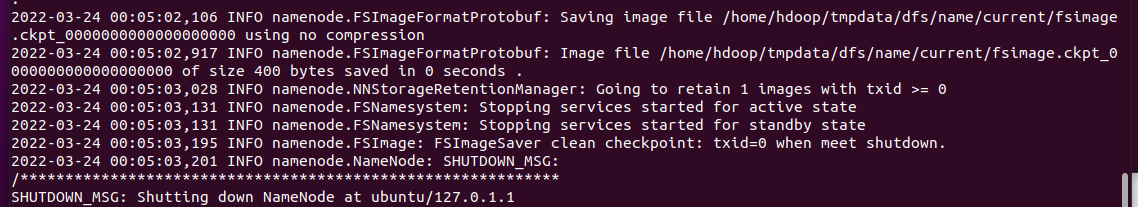
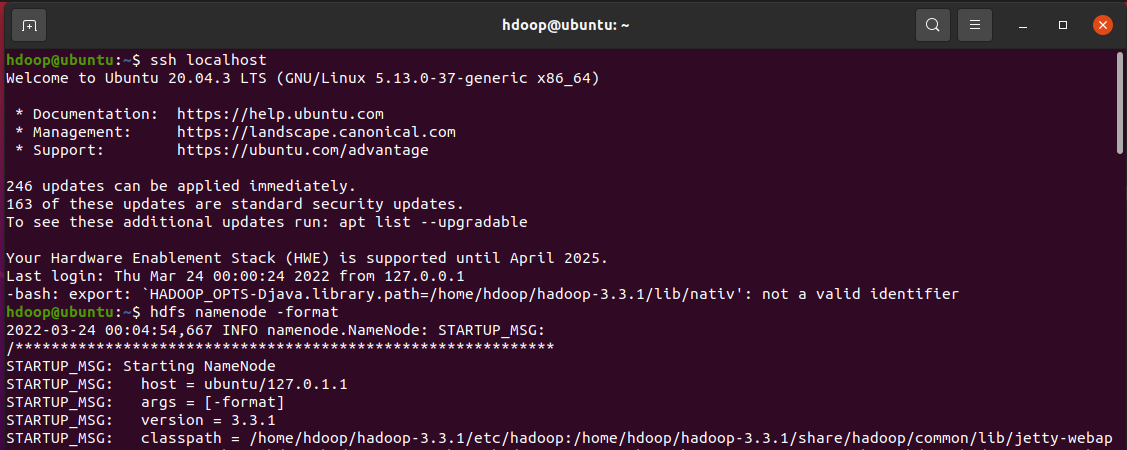


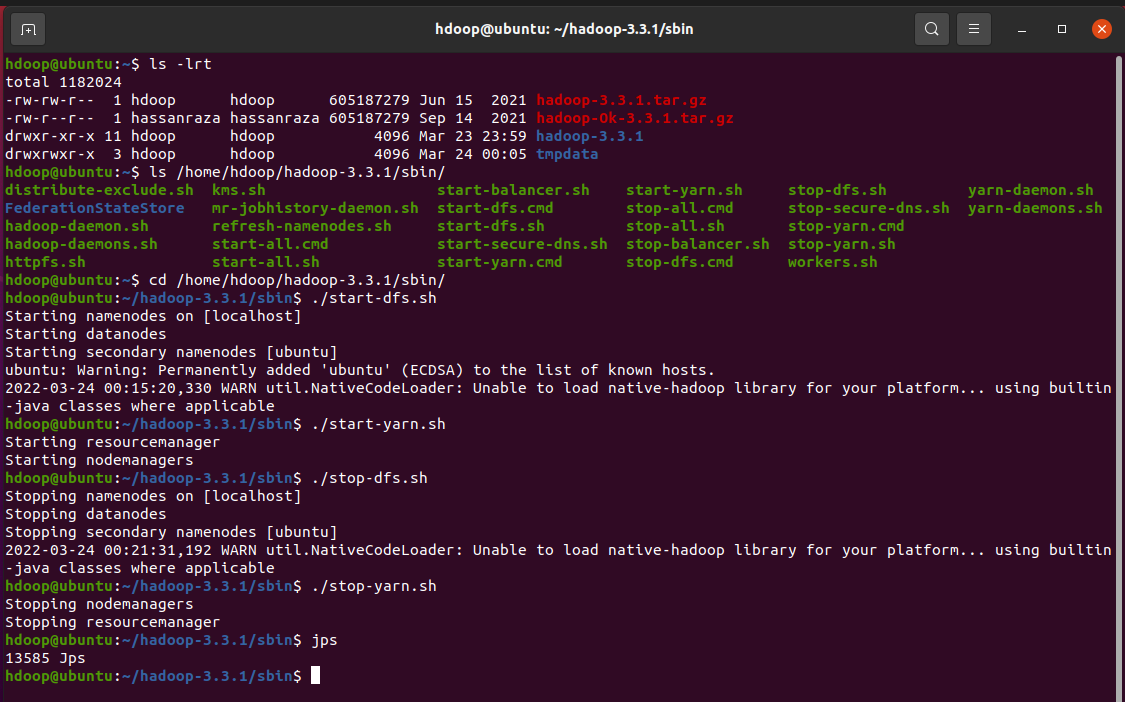
**3: mapred-site.xml**



**4: yarn-site.xml**

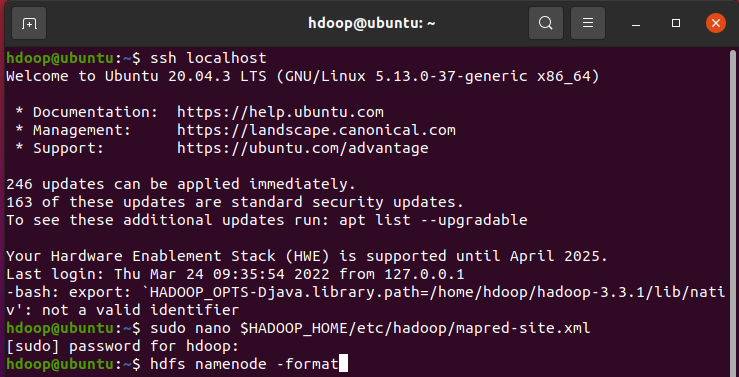
****

Open the SSH Local-Host. Make the format of Hadoop (XML) files. That you have modified change according to yourself.

Change the root of Hadoop “hadoop-3.3.1/sbin”. Start/Stop the services of Hadoop Distributed Files and Yarn.

# Local Host Web Server Access

# Open the new terminal then run “su – hdoop” and follow these process.



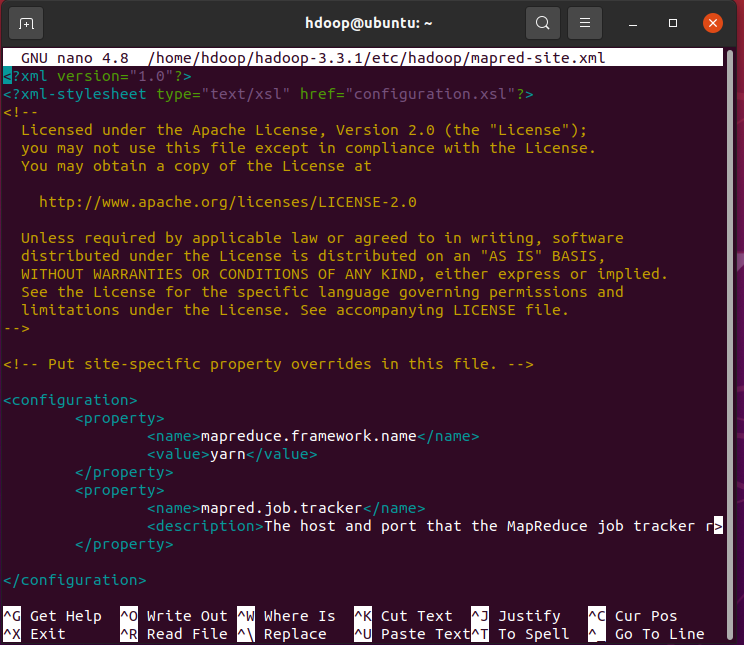
Just add the new “Property” in “mapred-site.xml” file.

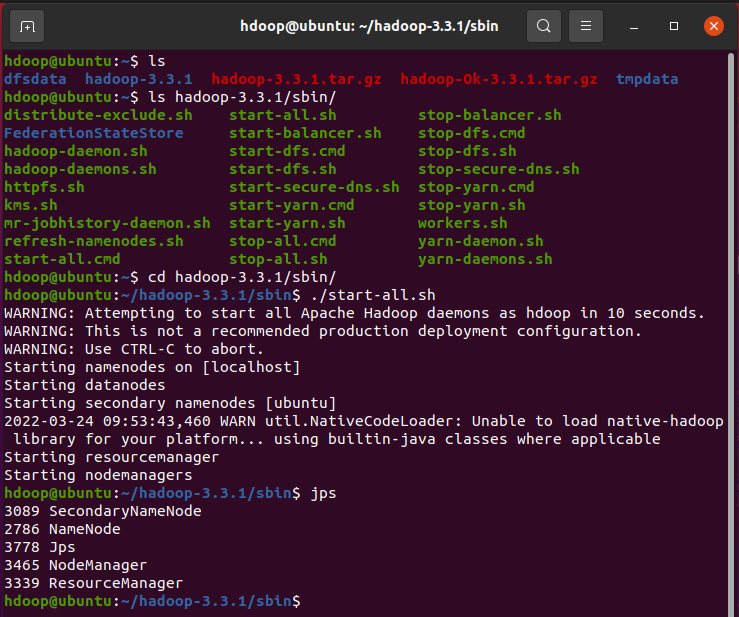
<property>

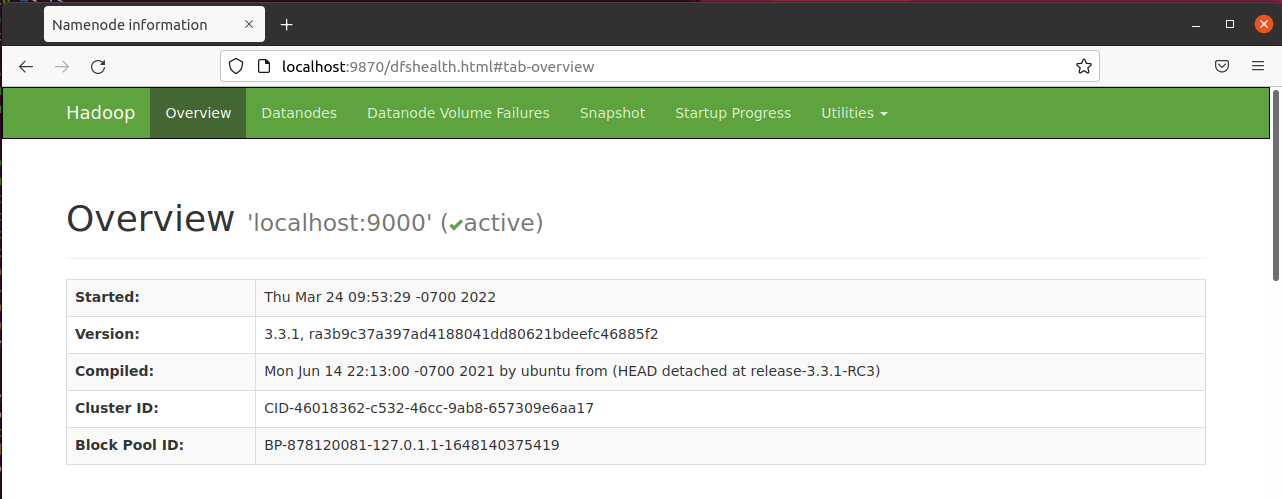
<name>mapred.job.tracker</name>

<description>The root and host start of Mapreduce Job Tracker at.</description>

</property>





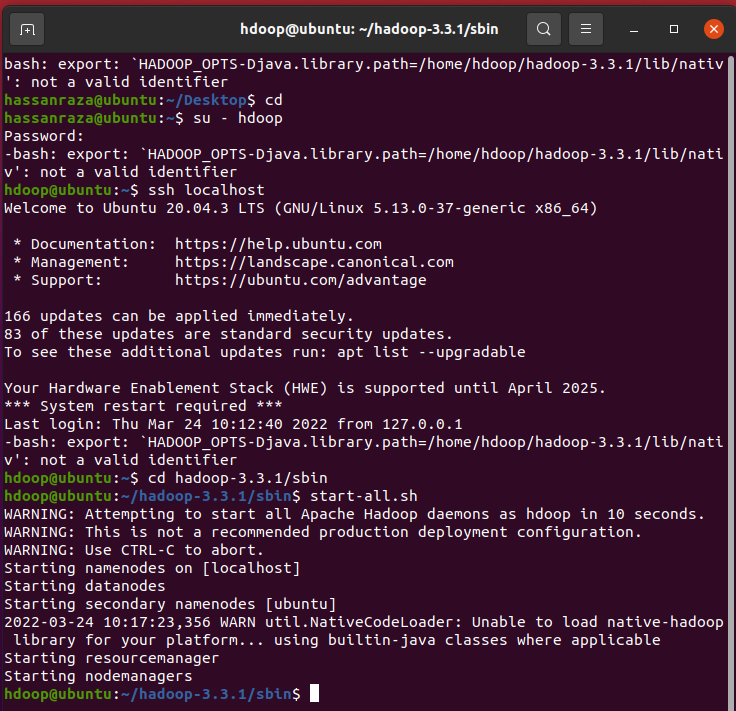


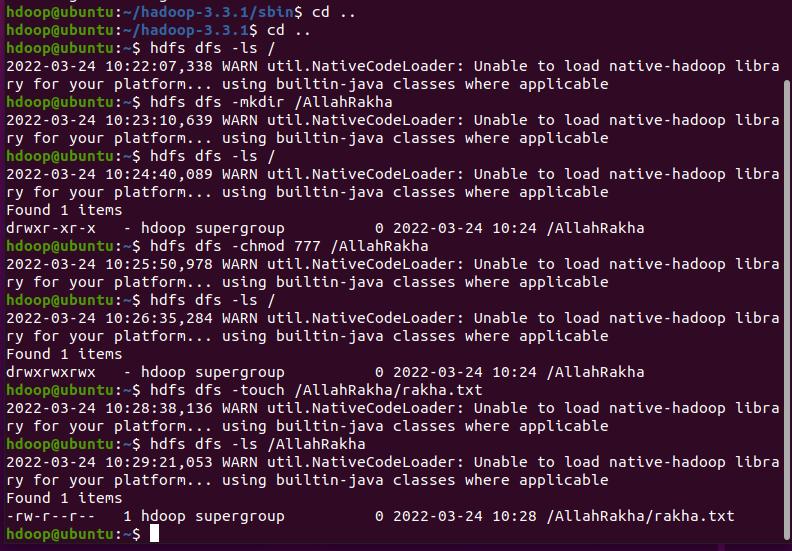
# HADOOP COMMANDS

Open the new terminal and then follow these process of commands.

Here, the all services start of Hadoop. Such as HDFS, YARN and MapReduce etc.

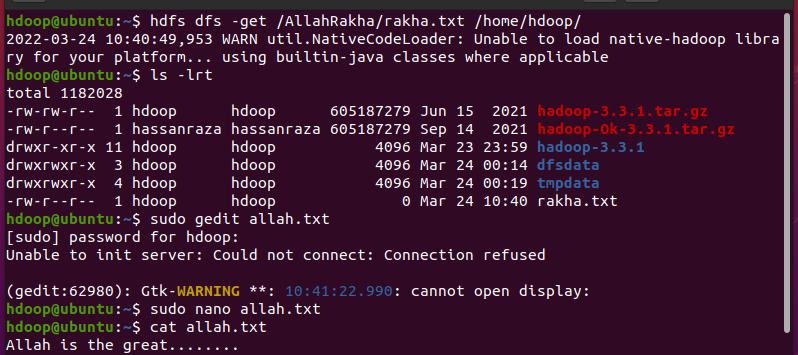
Folder, File create in Hadoop then change permission of folder.

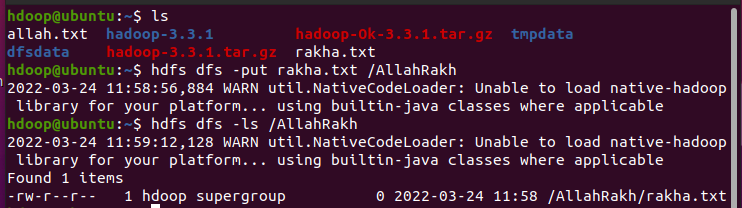


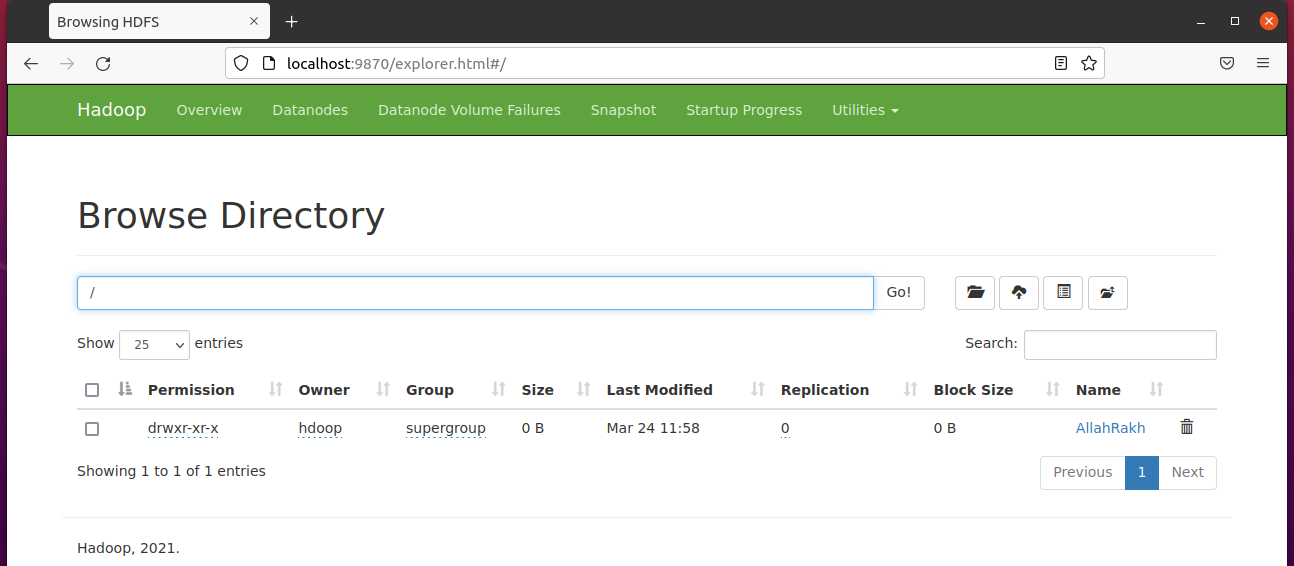


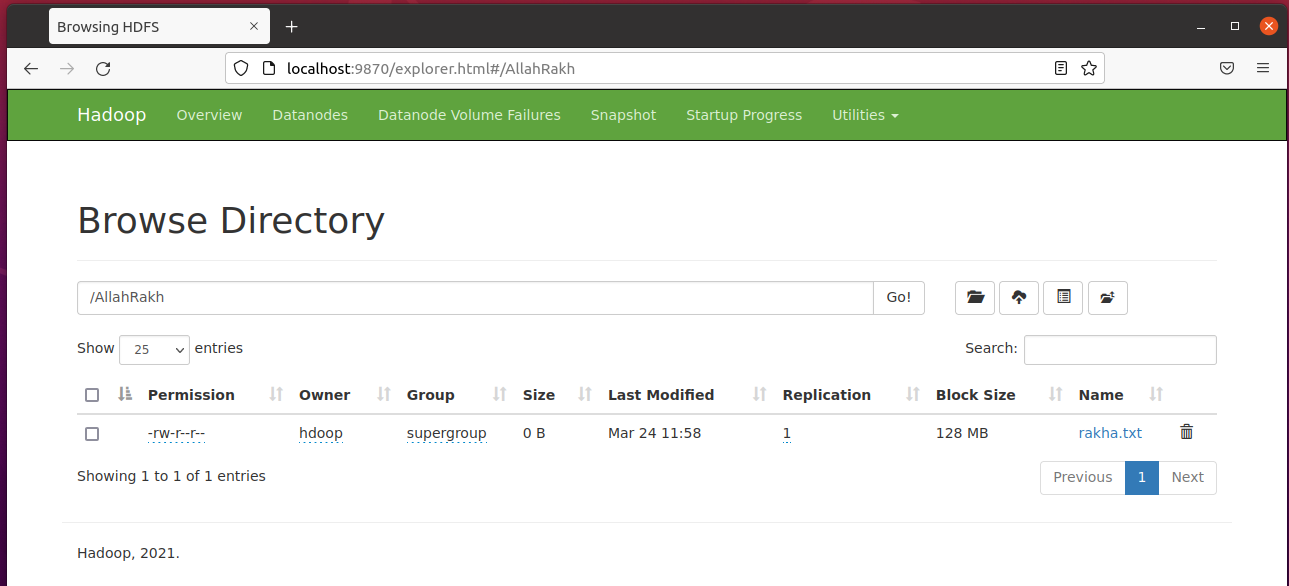
Get the Hadoop file from server “/AllahRakha” directory into your client local machine the root directory (hdoop).

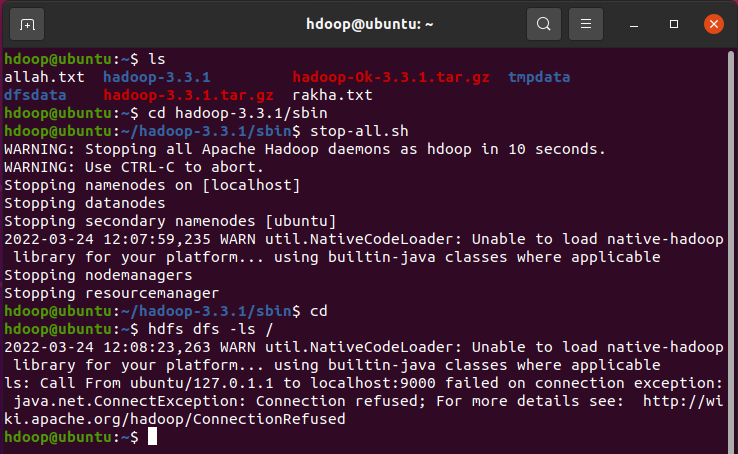
Put your client local machine the root directory (hdoop) into Hadoop server “/AllahRakha” directory.



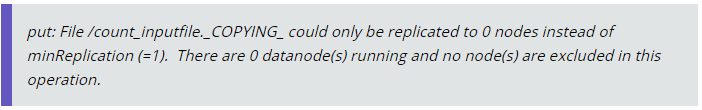


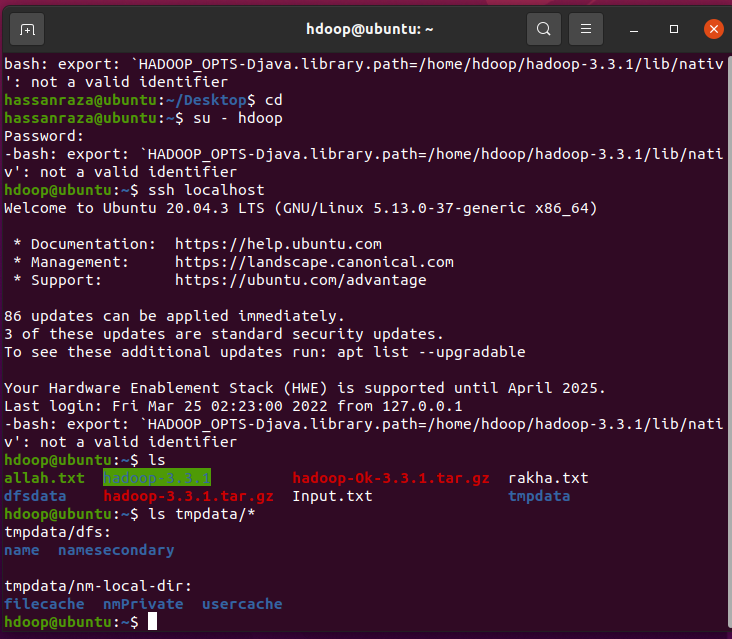


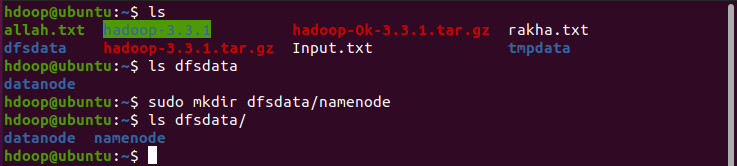


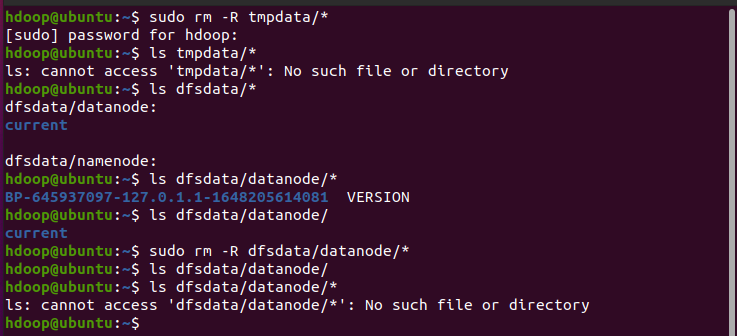


# PUT COMMAND ERROR AND RESOLVE (HADOOP FILE)

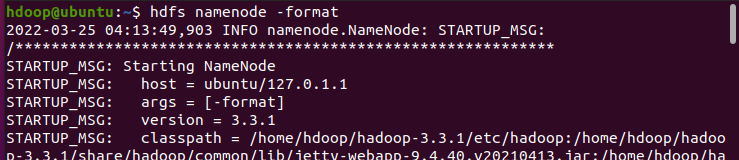
If any case this error appear at end of “put” command run in Hadoop. The Follow these process of commands to resolve error.

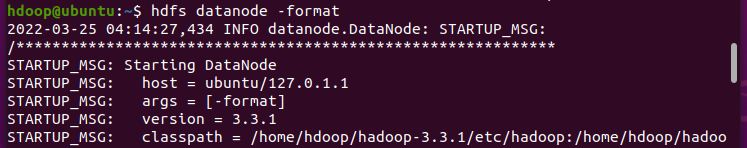


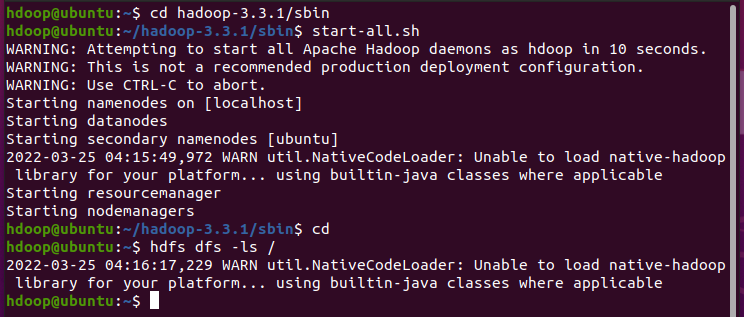
Here we create “namenode” directory in “dfsdata” folder.

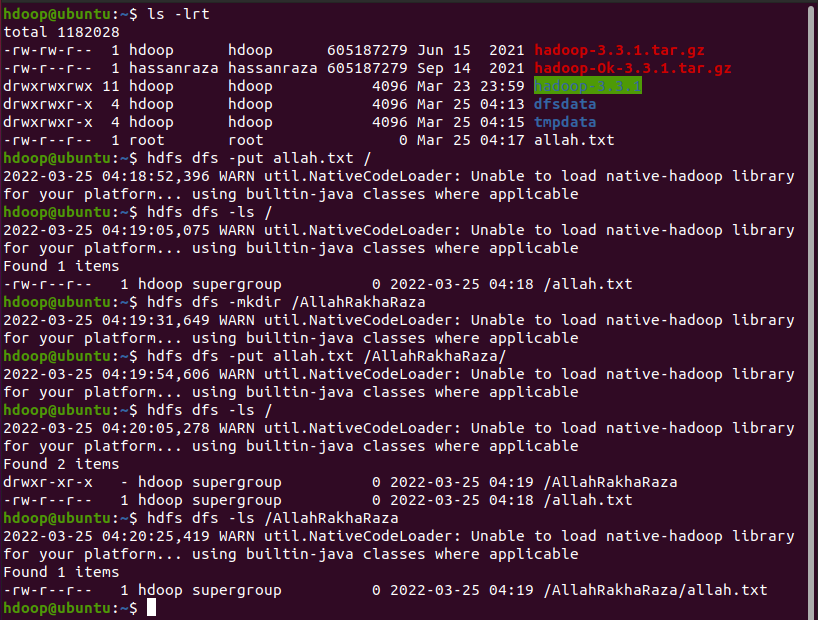
**Main Import Part.** Just remove all files and folder in “tmpdata” & “dfsdata/datanode”. Because new Hadoop installation in which “tmpdata” & “dfsdata/datanode” are store buffer record. The not accessible to “put” command into Hadoop Root Directory “/”.

**Point:** The “namenode” & “datanode” format is most import after remove some directory. Because, previous “cache” record delete and then create new empty record.

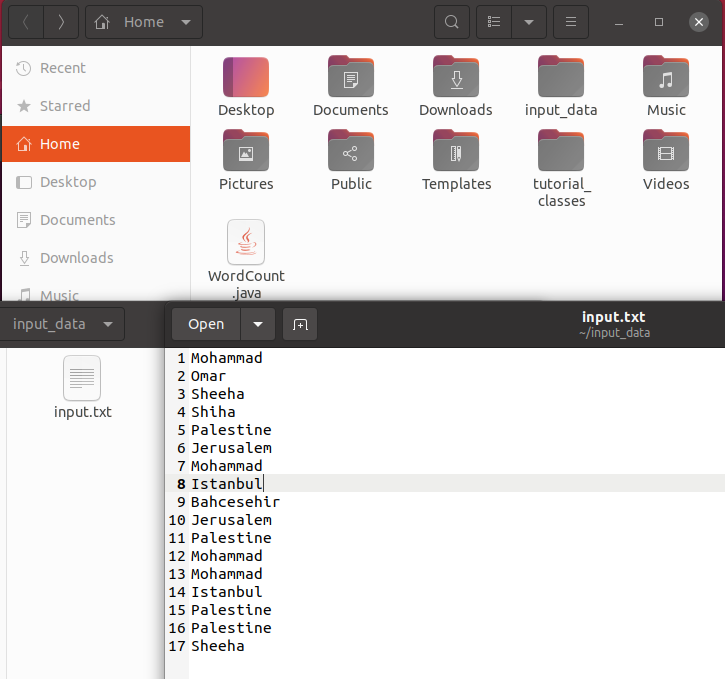


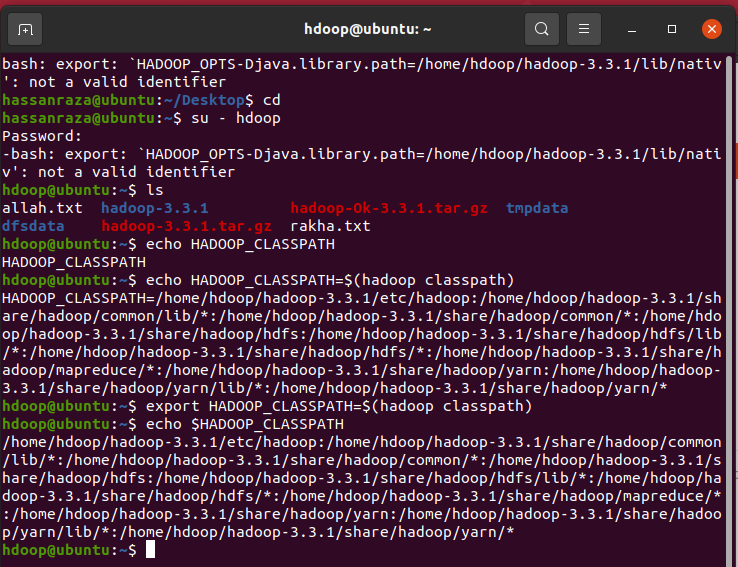


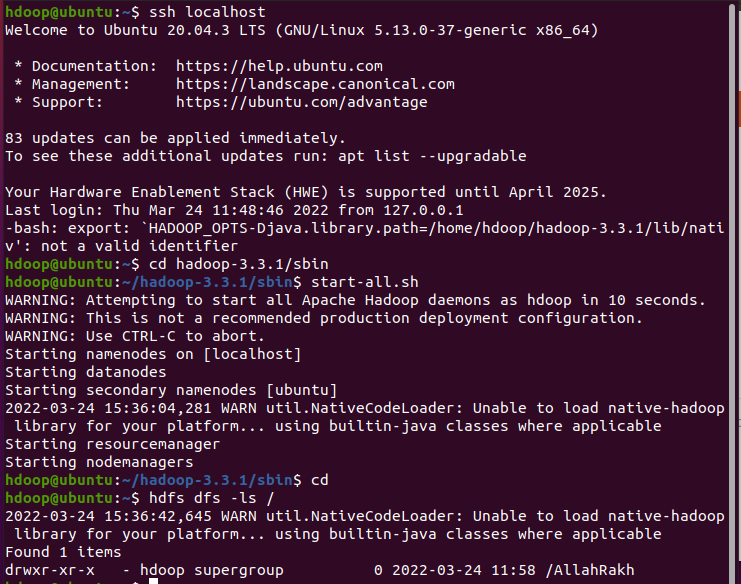


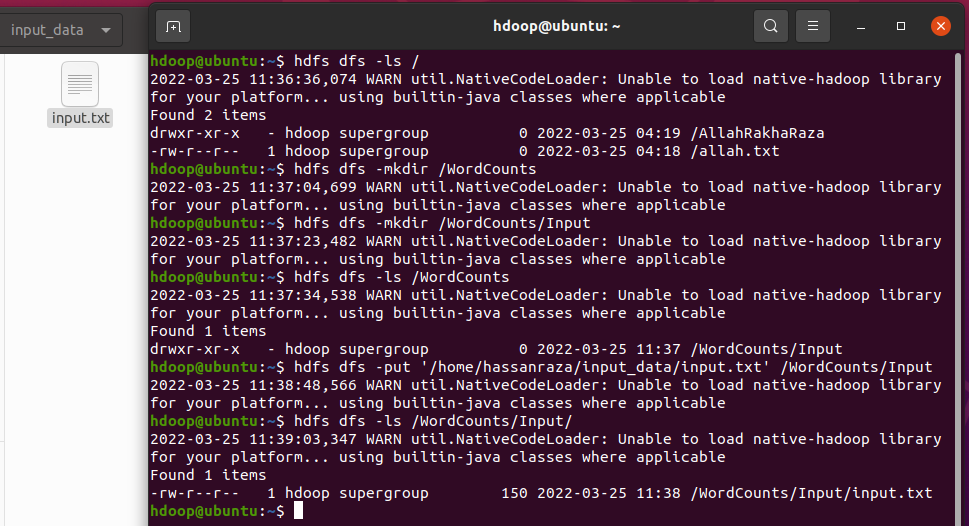
**Remember:** Here, we create a new file “allah.txt” in Hadoop Home Root Directory. Then put into Hadoop Server on Root Directory. Then create folder “/AllahRakhaRaza” and in which put “allah.txt” file.

# WORD COUNTS EXAMPLE

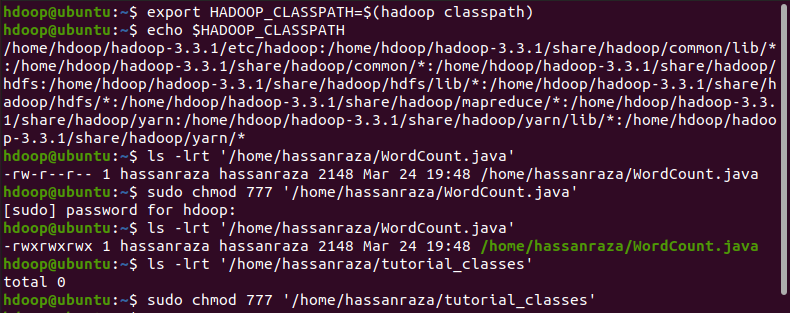
Create the two folder “input\_data” & “tutorial\_classes”. Then paste “WordCount.java” file in “/home/hassanraza” directory. File create “Input.txt” in which folder of “input\_data”.

Add “export” the hadoop classpath to global/local variable of HADOOP\_CLASSPATH and then echo or cat for testing.



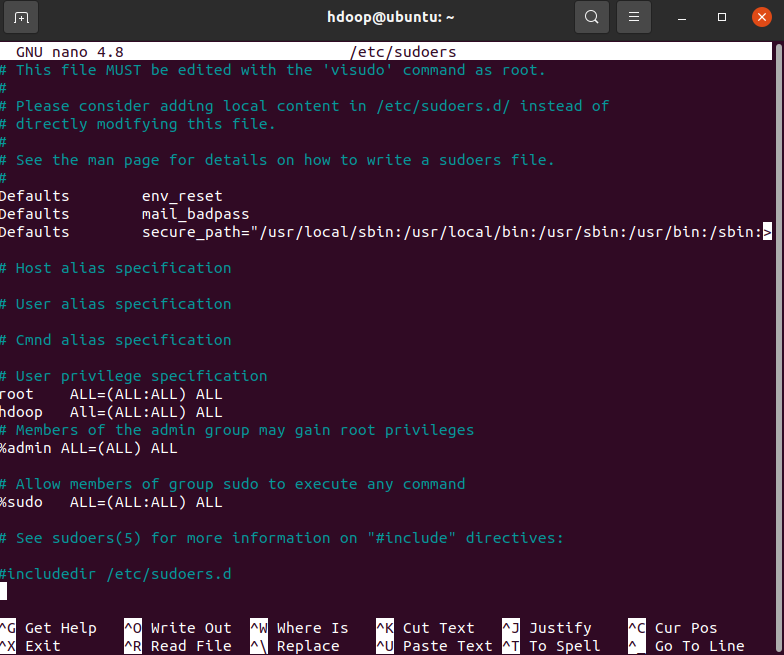
 Create “WordCounts/Input” directory in Hadoop. Then put file of “input.txt”.

Change the Mode of file and folder (WordCount.java, tutorial\_classes) with read\_write\_execute permission.

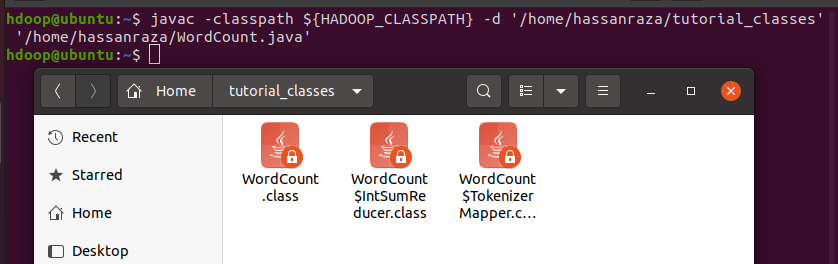


Add all privilege user specification permission to “hdoop” user.





Create “Word Counts” some files to counting a words in “tutorial\_classes” folder. Here, HADOOP\_CLASSPATH used for API access to create these files. First path use for Output files (tutorial\_classes) and Second path use for Input file (WordCount.java). “-d” here directory path show. Here, “-classpath” to assign “HADOOP\_CLASSPAHT” all list of paths that help out for “javac” execution.



Then change permission of all files “tutorial\_classes” folder. Assign read\_write\_execute permission to files.

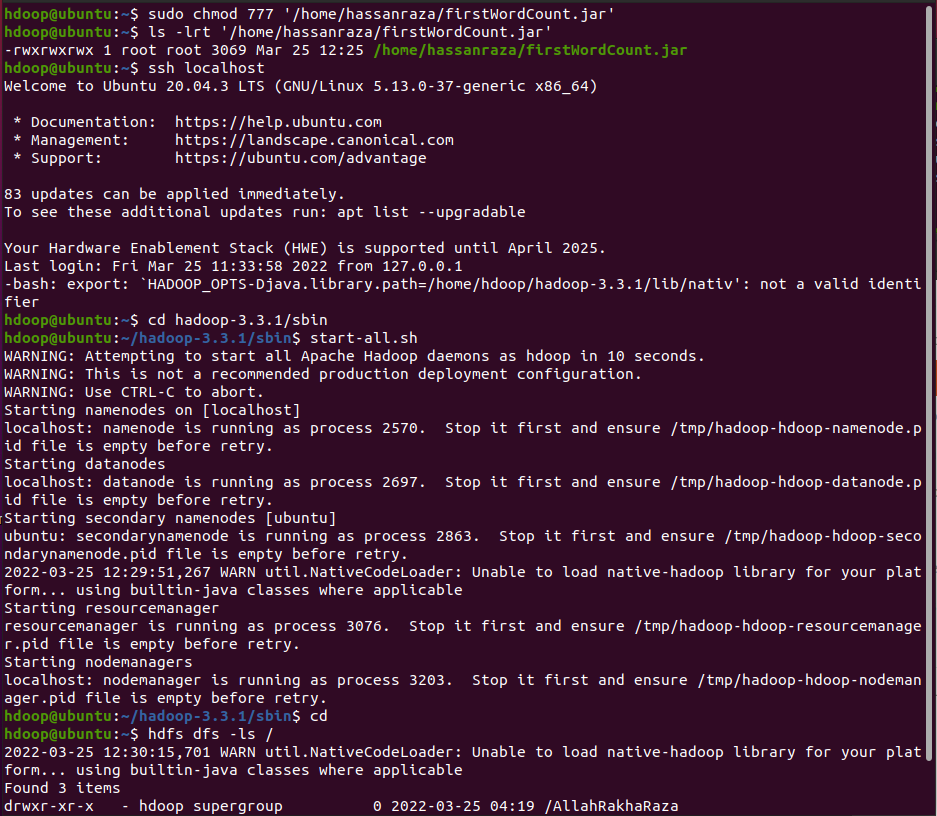
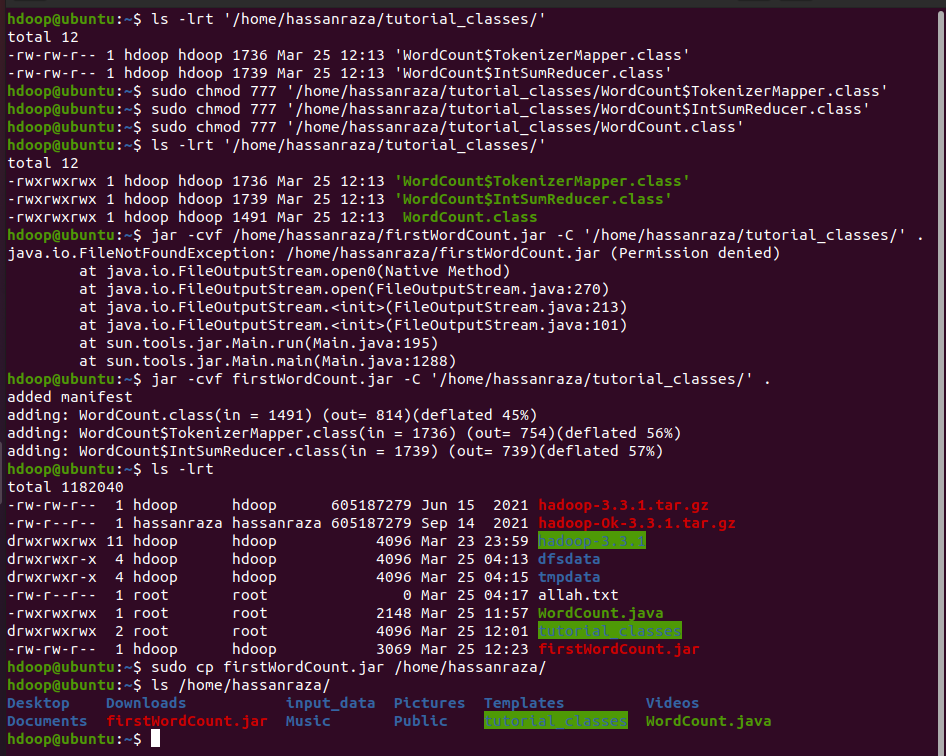
**Command**: jar -cvf firstWordCount.jar -C “/home/hassanraza/tutorial\_classes/” .

* Jar –cvf : used for compressed all files into single file. Location current directory root “/home/hdoop”.
* -C : use to access all files for compressing.
* “/home/hassanraza/tutorial\_classpath/” : Path (provide all files to compressing into single file).

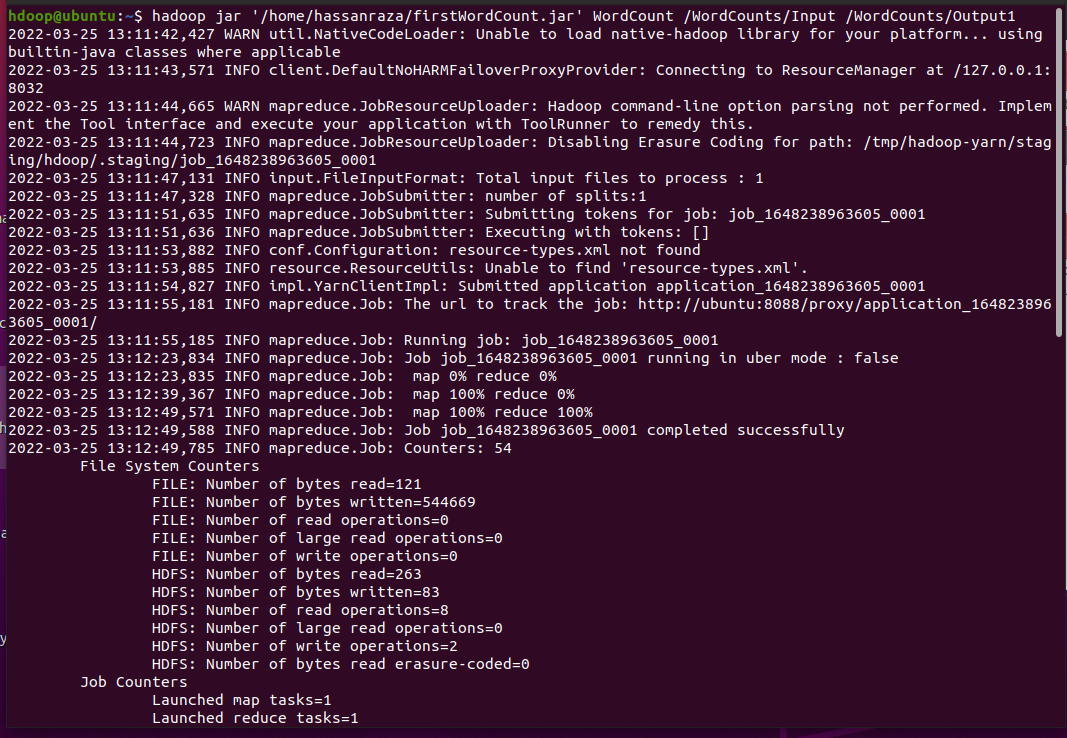
Then copy file “/home/hdoop/” to “/home/hassanraza/” directory root.

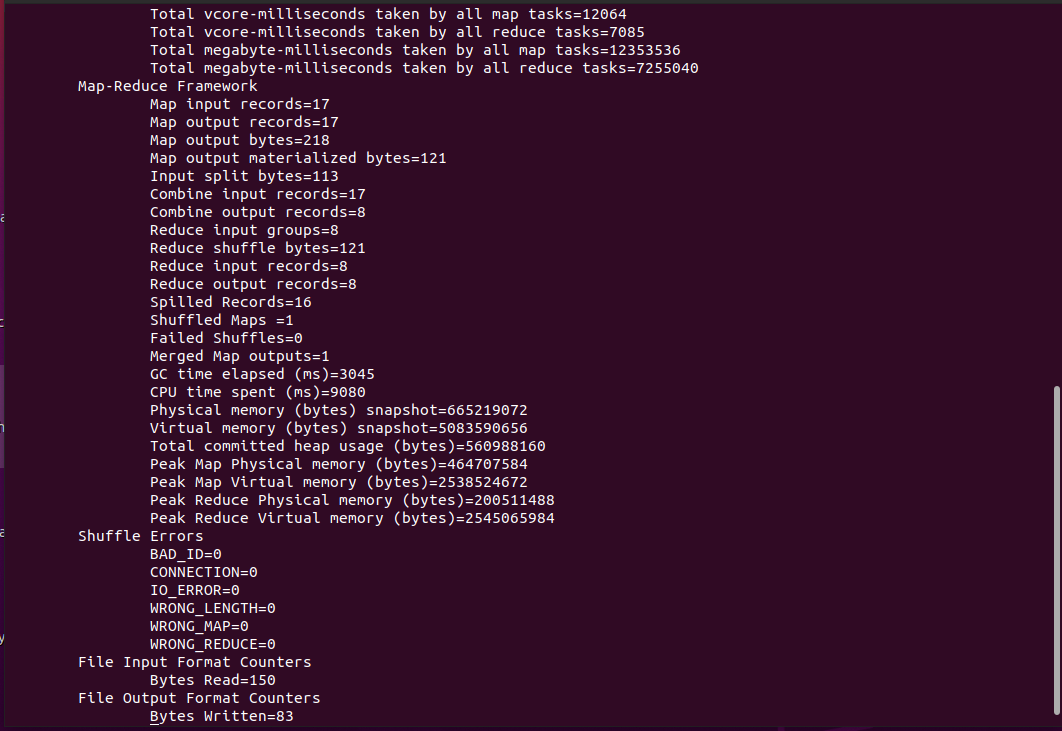
Then change permission of “firstWordCount.jar” file. Assign read\_write\_execute permission.

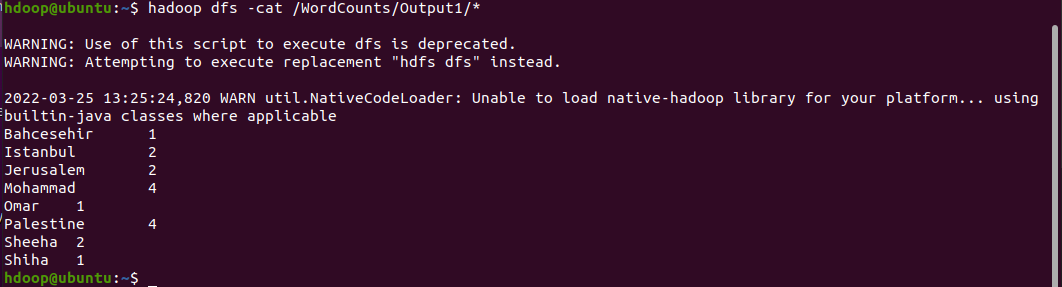
Then open “ssh localhost” and start all services of Hadoop.

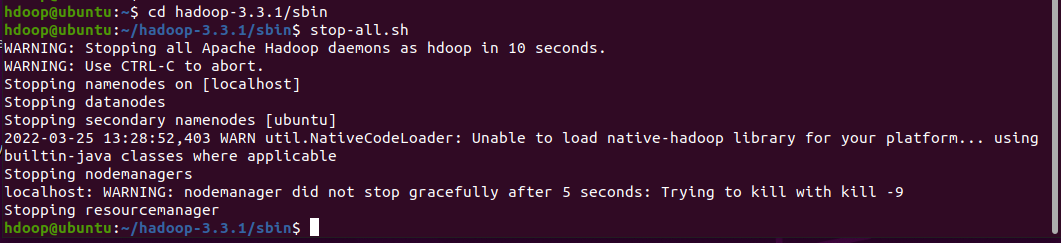


**Command**: hadoop jar ‘/home/hassanraza/firstWordCount.jar’ WordCount /WordCounts/Input /WordCounts/Output1

* hadoop jar : used for upload “firstWordCount.jar” into Hadoop Server and counting all words from file.
* ‘/home/hassanraza/firstWordCount.jar’ : used path of file that upload into Hadoop server.
* WordCount : class name of file original “WordCount.java”.
* /WordCounts/Input : path used to access “input.txt” file.
* ****/WordCounts/Output1 : path used to store all output record of relative counts word.



Show the output of record from “/WordCounts/Output1/\*” folder.

Shut down all services of Hadoop.