Setting Up Hadoop Environment

What is Big Data ?

Bernard Marr, defines Big Data as the digital trace that we are generating in this digital era. The Digital trace is made up of all the data that is captured when we use digital technology.

Ernst Young, refers Big Data to the dynamic large and disparate volume of data being created by people,tools and machines.

V's of Big Data

- **Velocity**-> Speed of data at which data accumulates.
- **Volume**-> Scale of data or increase in amount of data store.
- **Variety**-> Diversity of data means Structured, Unstructured.
- **Veracity**-> Conformity to facts and accuracy.

Characteristics of Big Data

1. Recommendation Engines are a common application of big data.
2. Companies like Amazon, Netflix use the Algorithm of Big Data.
3. SIRI personal assistants of Apple devices use Big Data to answer the infinite number of questions.
4. Google makes recommendations based on Big Data.

What is Hadoop

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

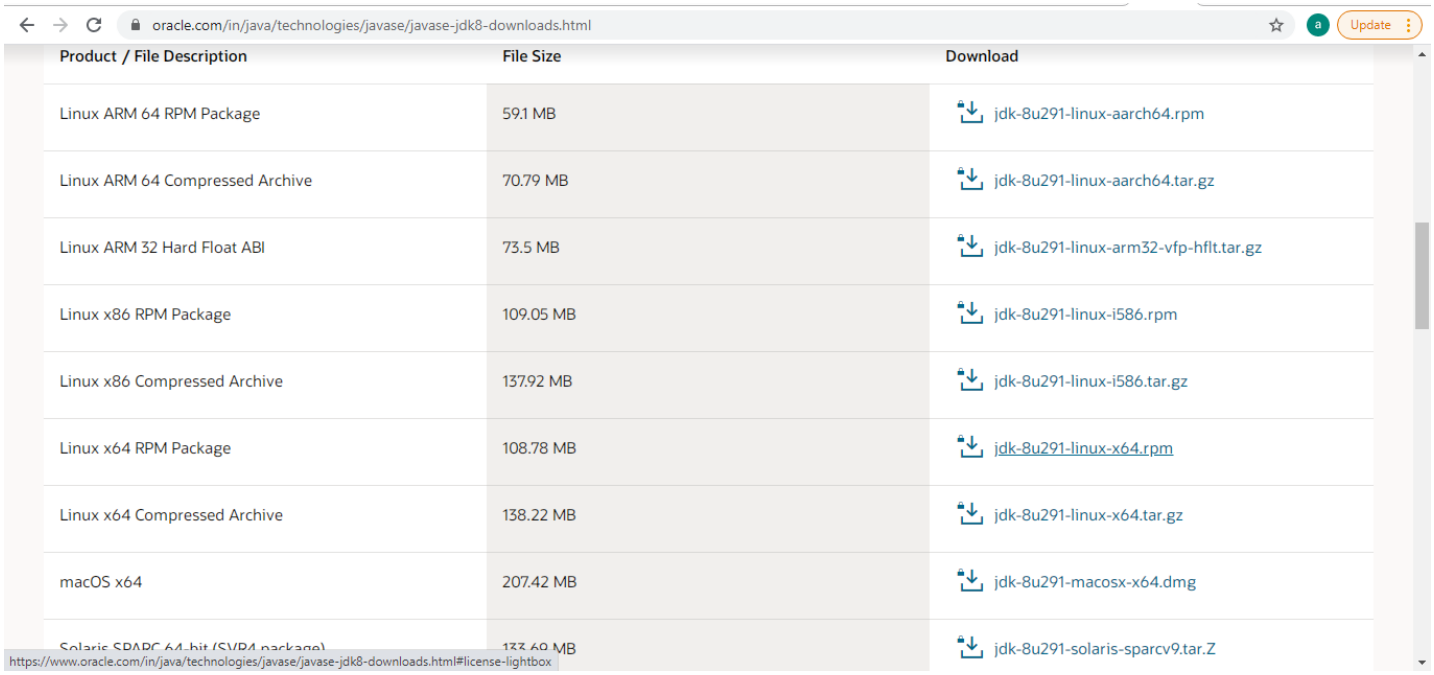
Features of Hadoop

- Hadoop is Open Source
- Hadoop cluster is Highly Scalable
- Hadoop provides Fault Tolerance
- Hadoop provides High Availability
- Hadoop is very Cost-Effective
- Hadoop is Faster in Data Processing
- Hadoop is based on the Data Locality concept
- Hadoop provides Feasibility

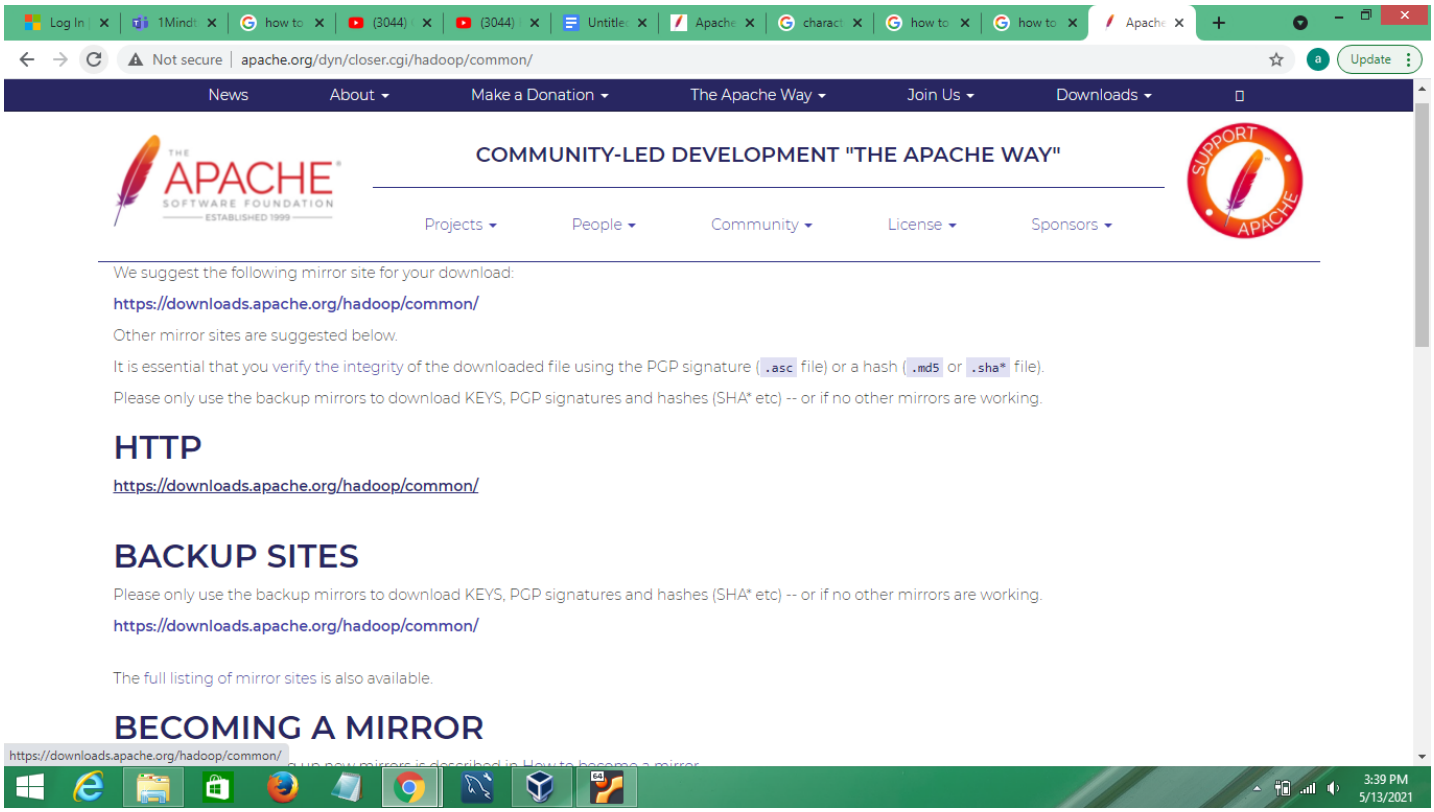
KO1: Able to install Hadoop

1.Hadoop is running using Java so first download java if it is not installed using the below link.

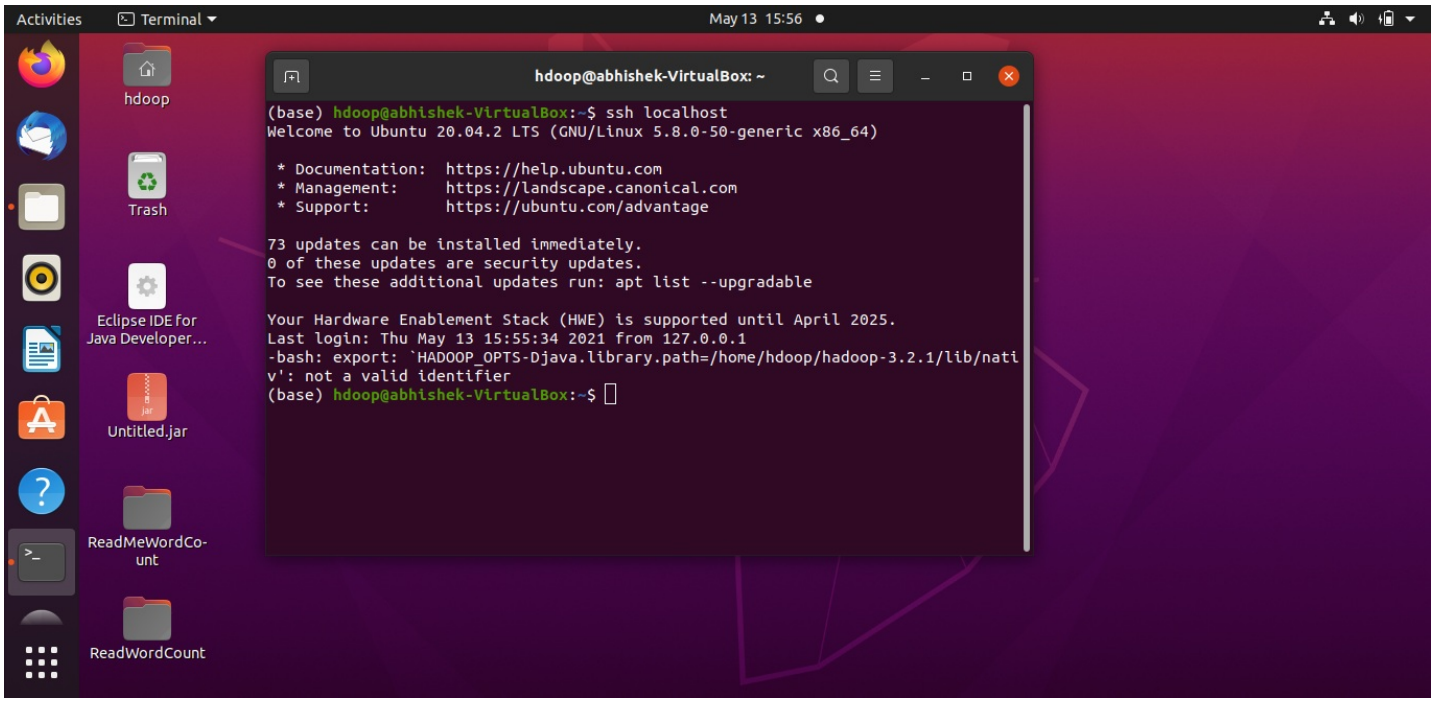
Download Java



2.Download Hadoop from apache.org

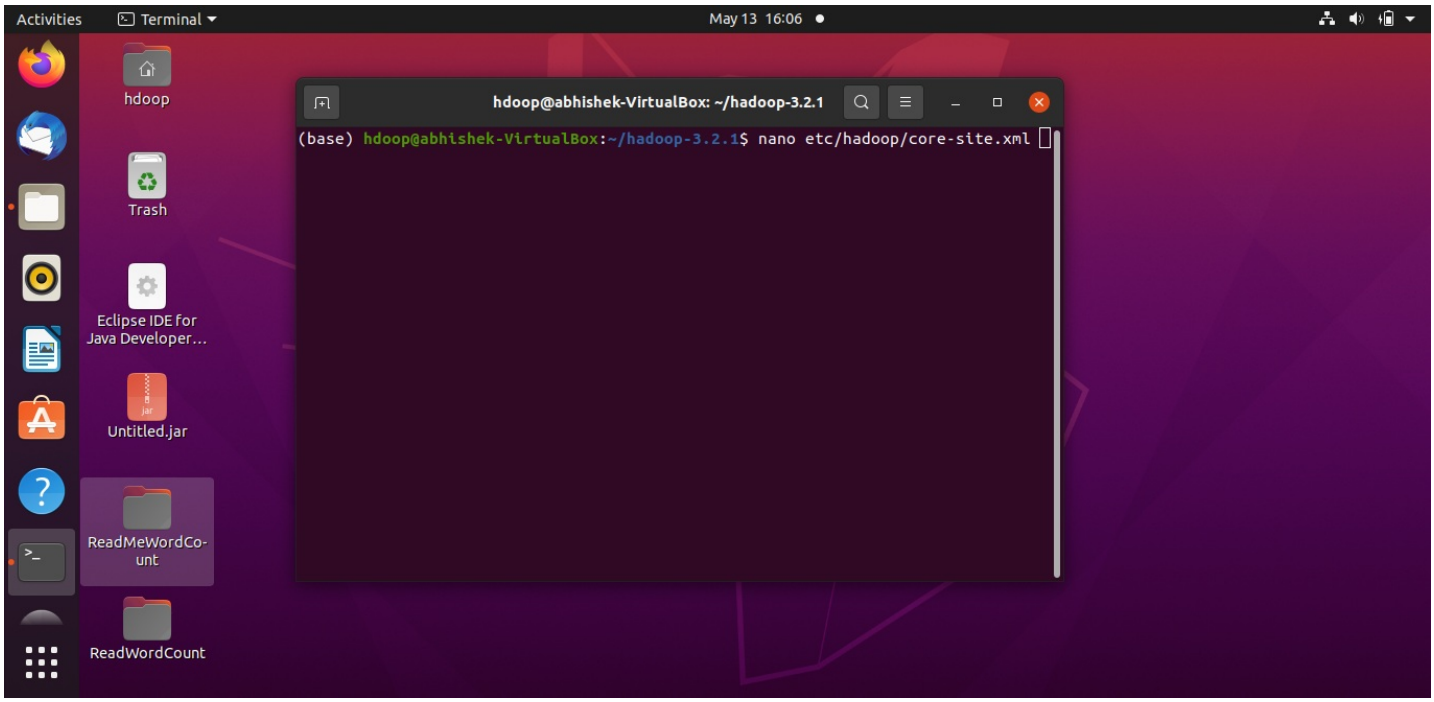


3.Check the localhost connection .



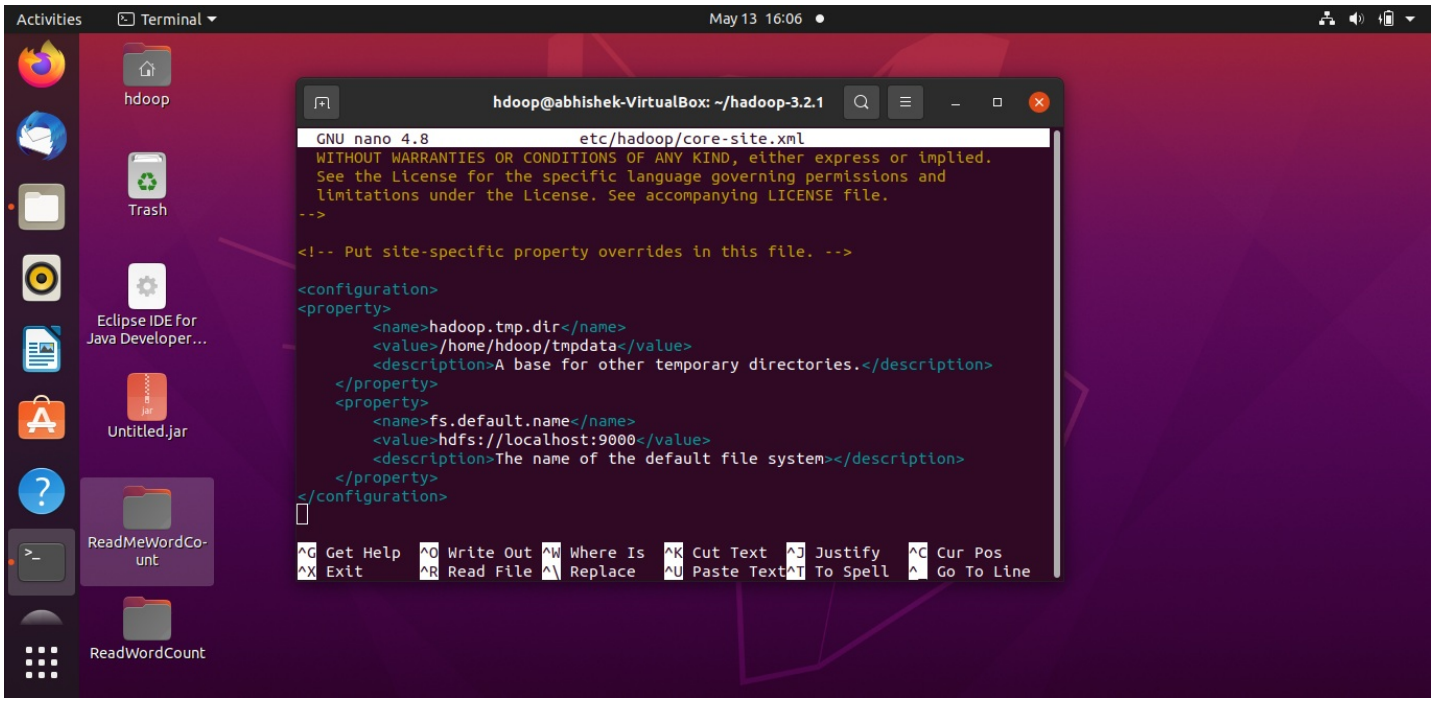
4.Unzip the downloaded Hadoop file. The Unzipped folder contains many subfolders.Make the changes in the **etc** folder.

5.Open the file **core-site.xml**

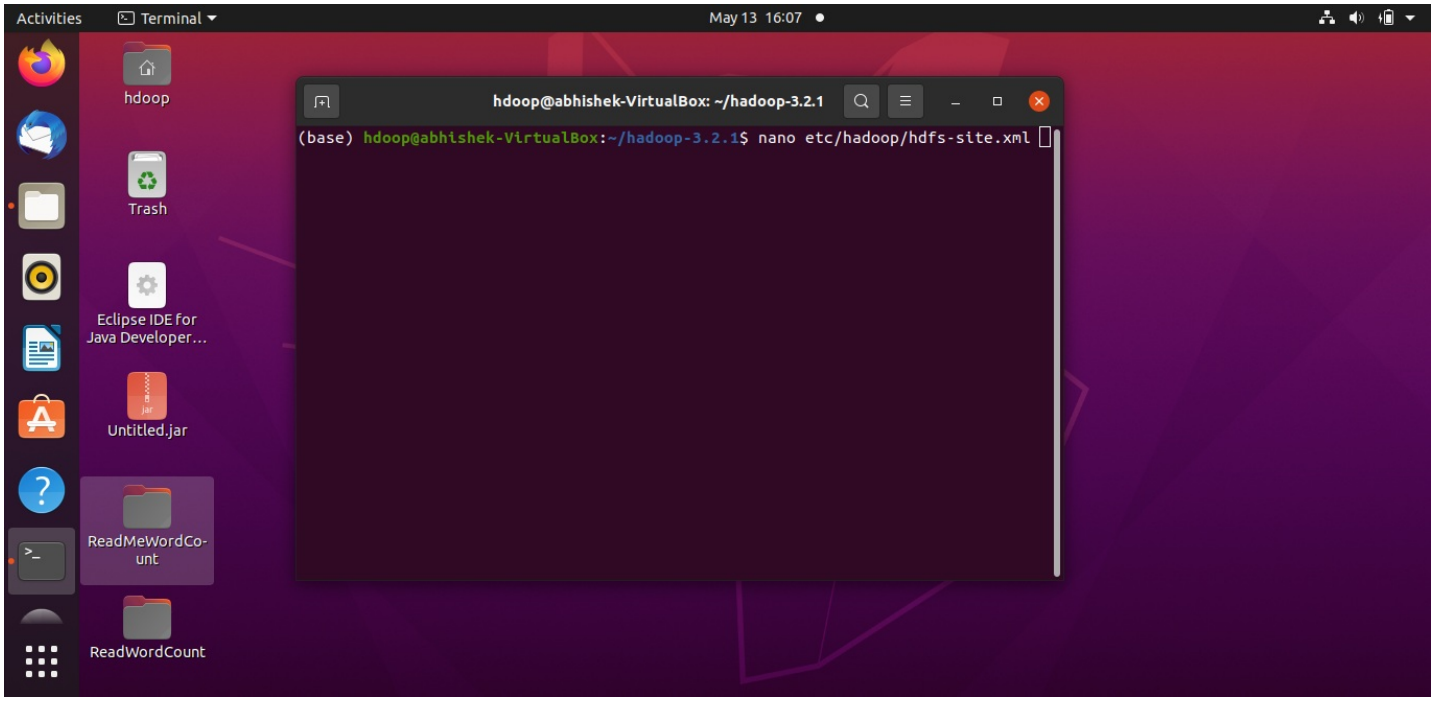


Add two properties in this file.

- temporary directory
- default file system

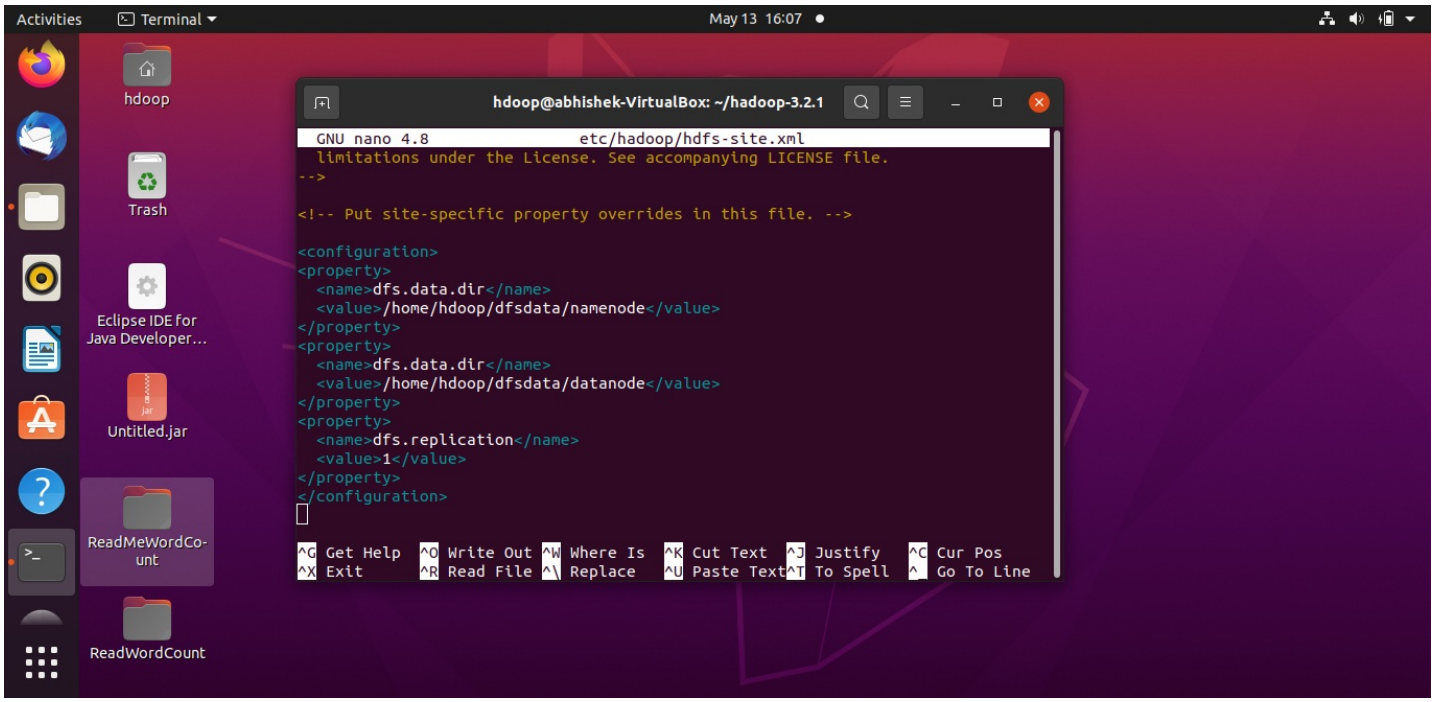


7.Now open the file **hdfs-site.xml**.

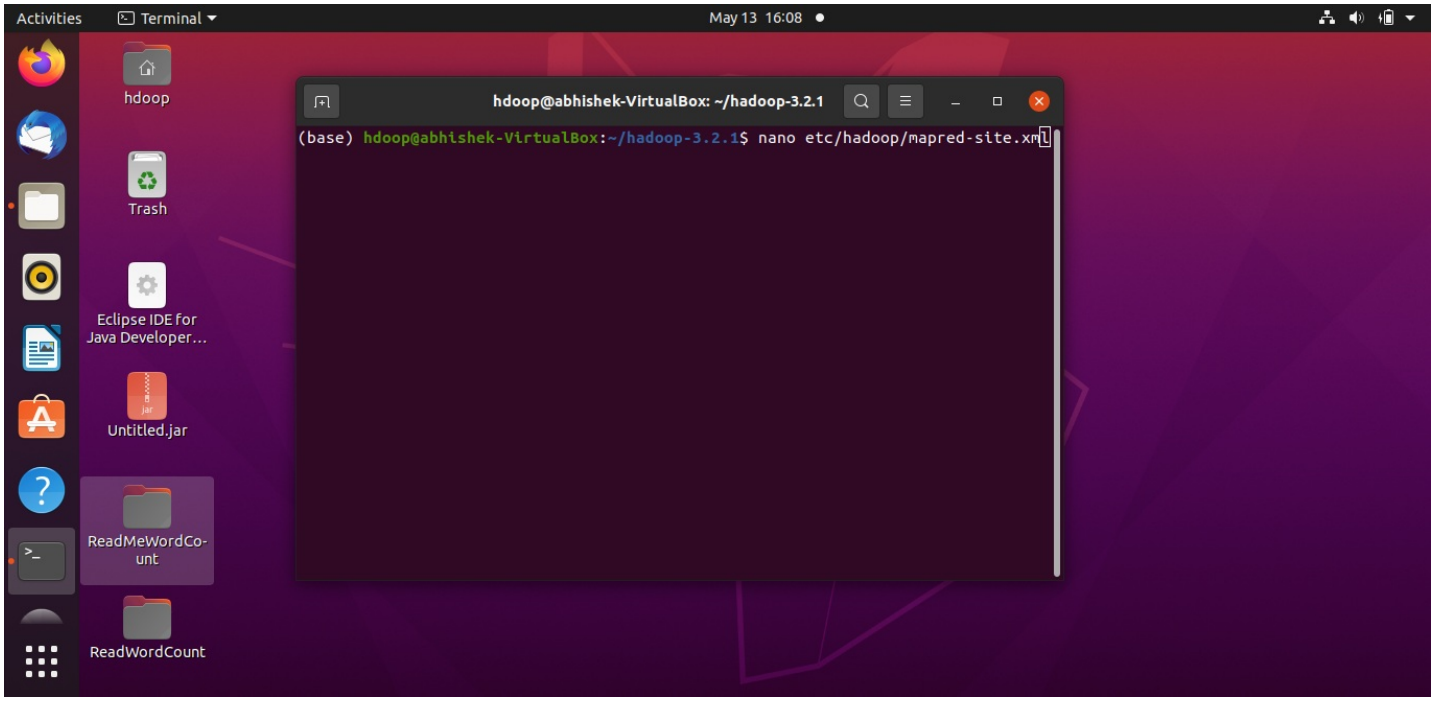


Add three properties in this file.

- datanode
- namenode
- duplicated values

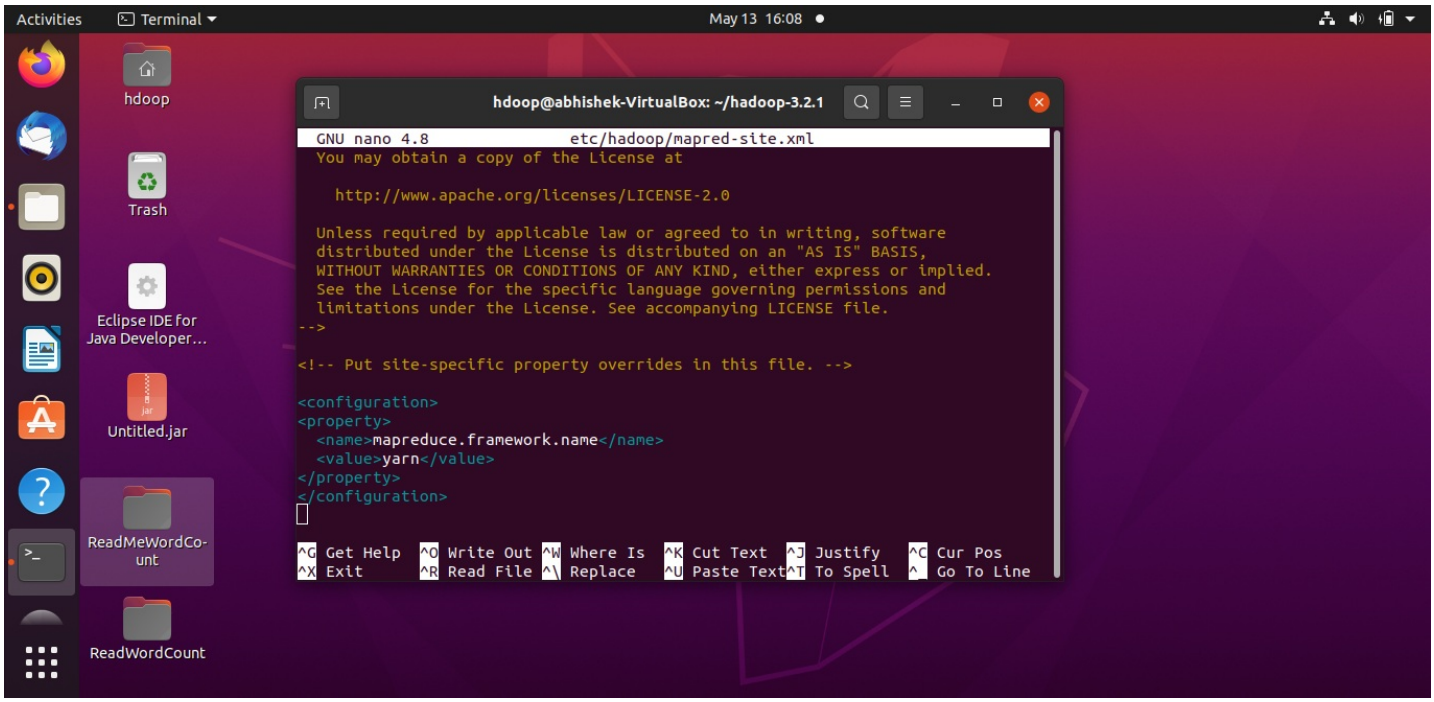


8.Now open the file **mapred-site.xml**.

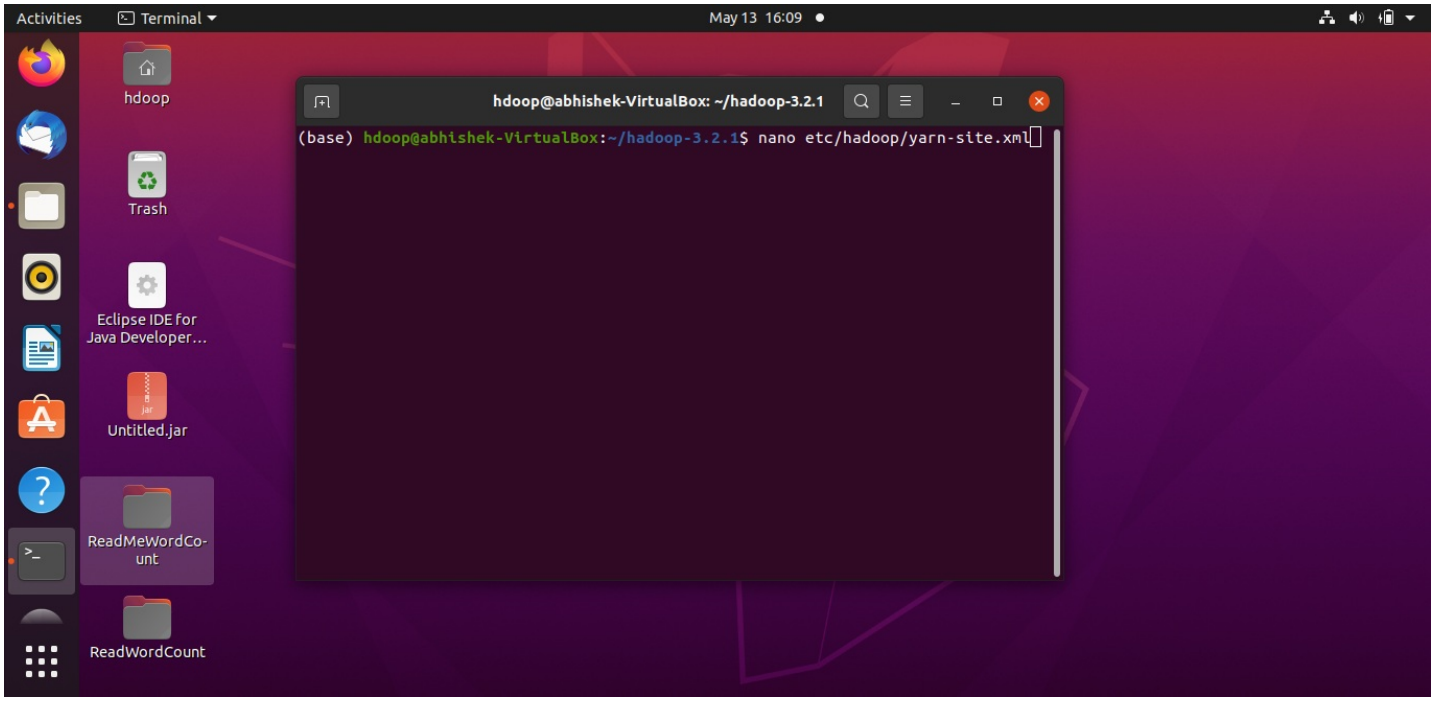


Add one properties in this file.

- yarnframework

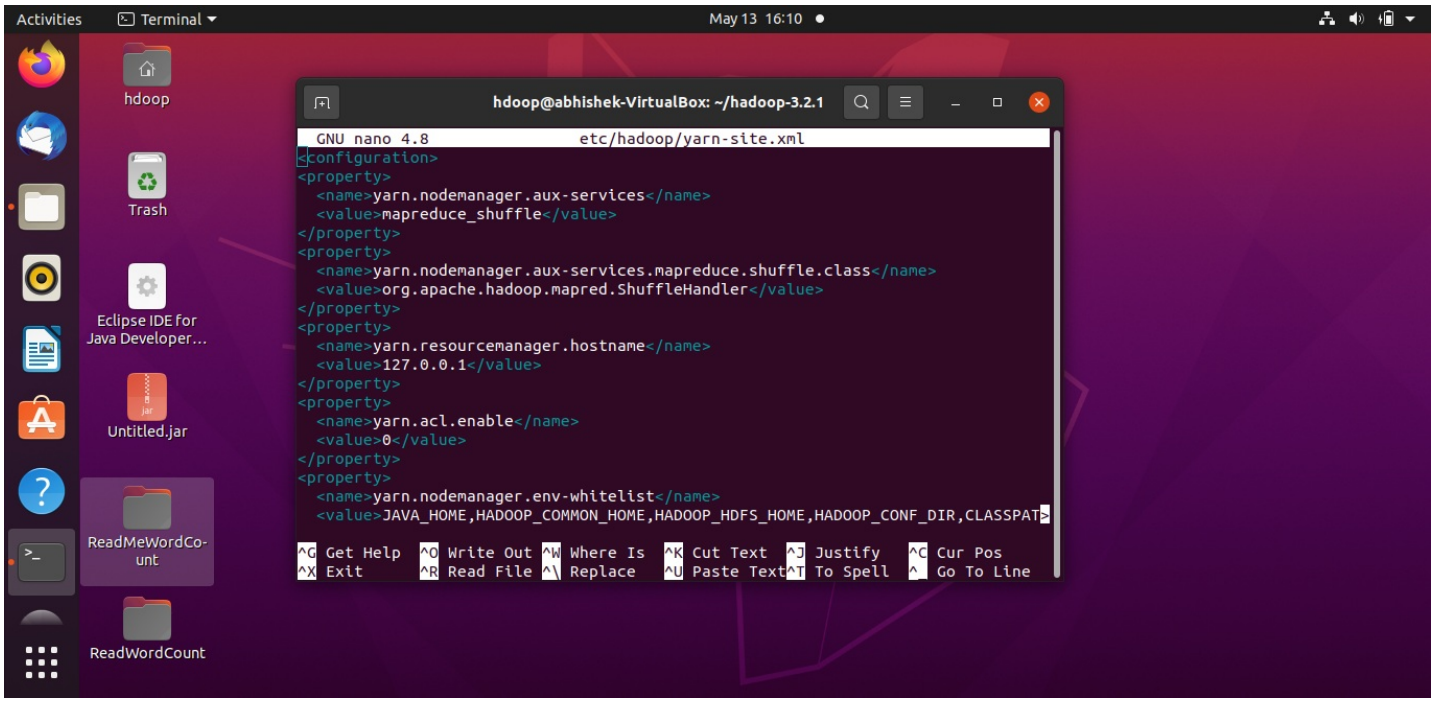


9.Now open the file **yarn-site.xml**.

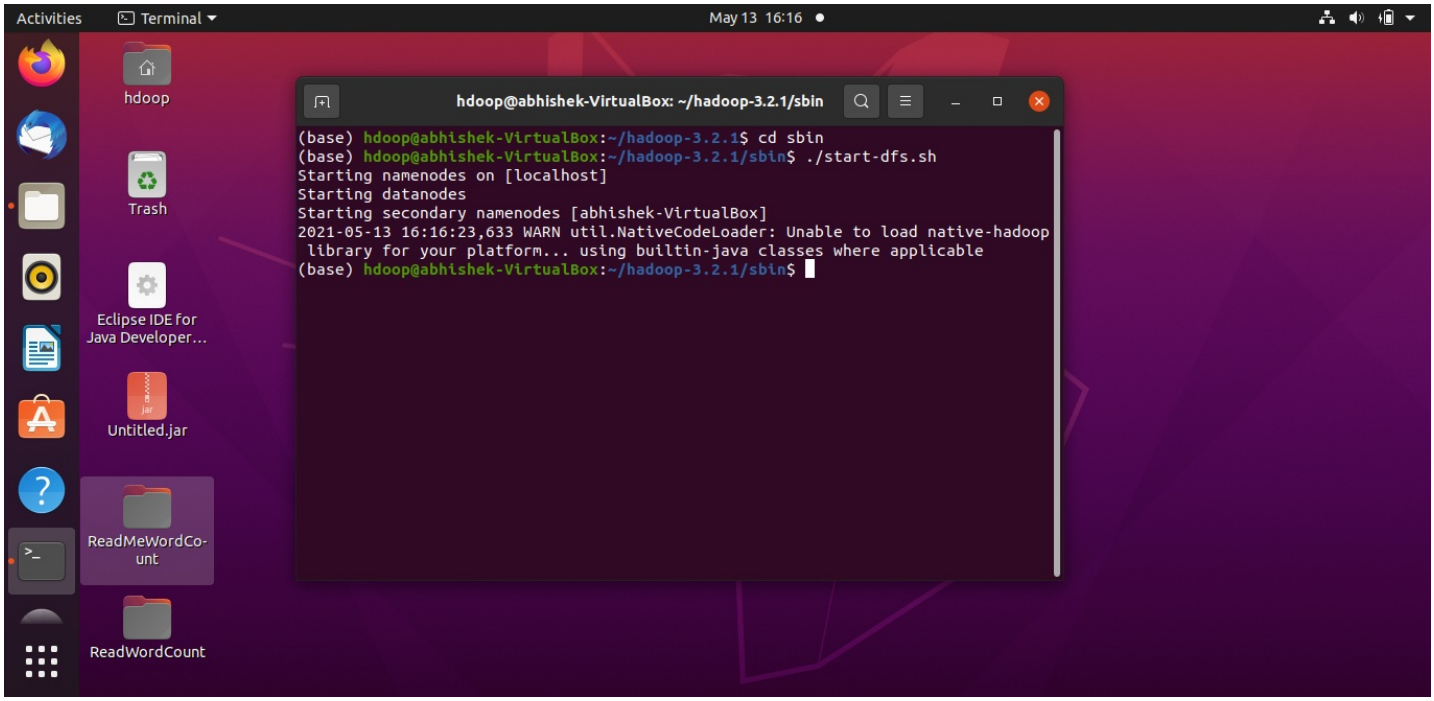


Add five properties in this file.

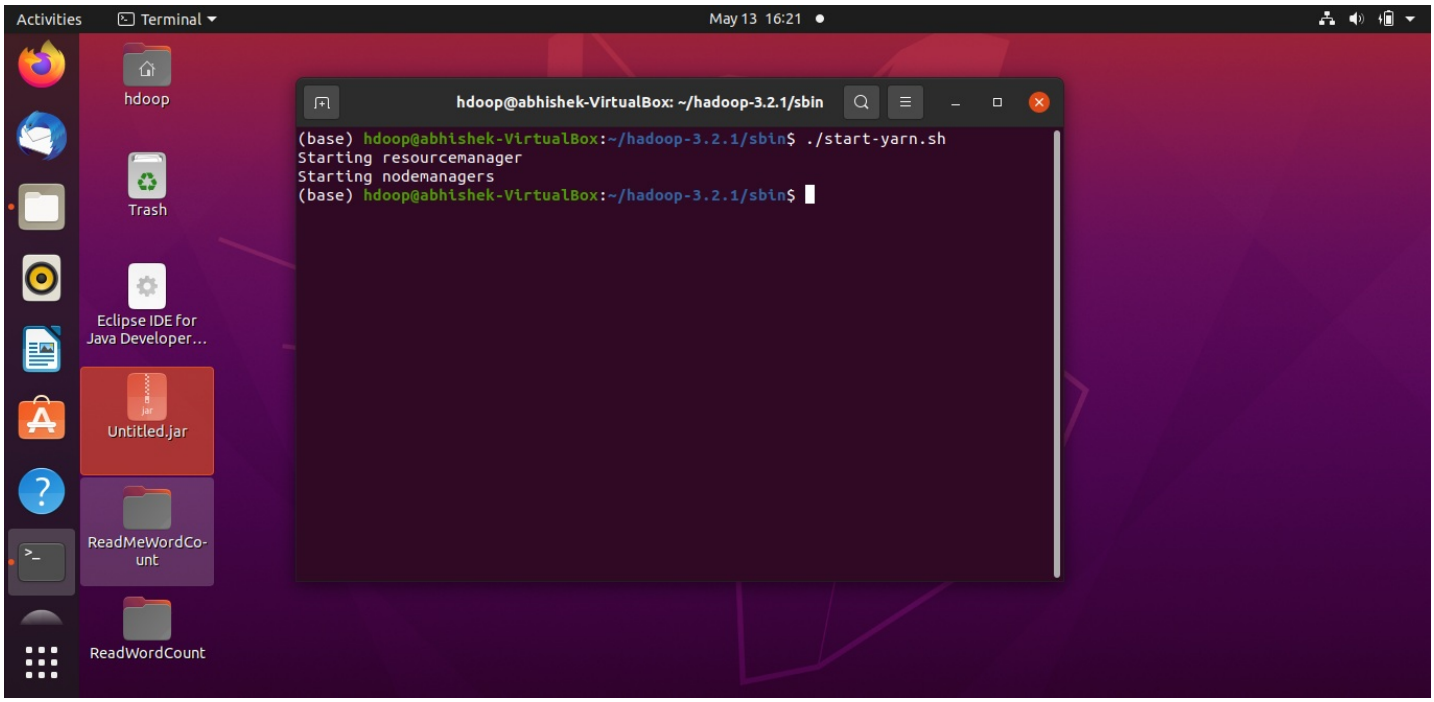
- services
- class
- hostname
- enable acl
- add whitelist



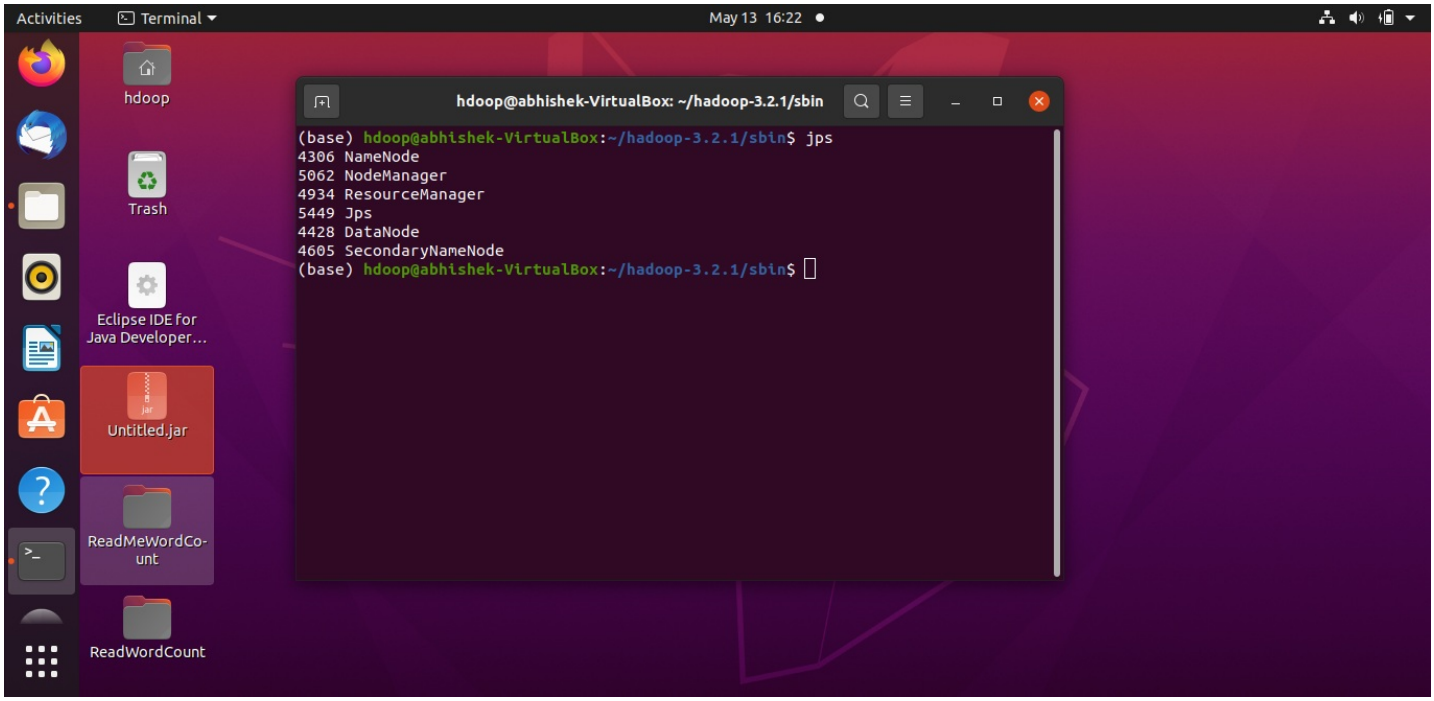
10.Start the **dfs** services



11.Start the **yarn** services



12.Check that all the services of dfs and yarn is started by running **jps** command.



- All the services has started.Hadoop is successfully installed we are good to go further.

KO2: Able to store data in HDFS

What is HDFS

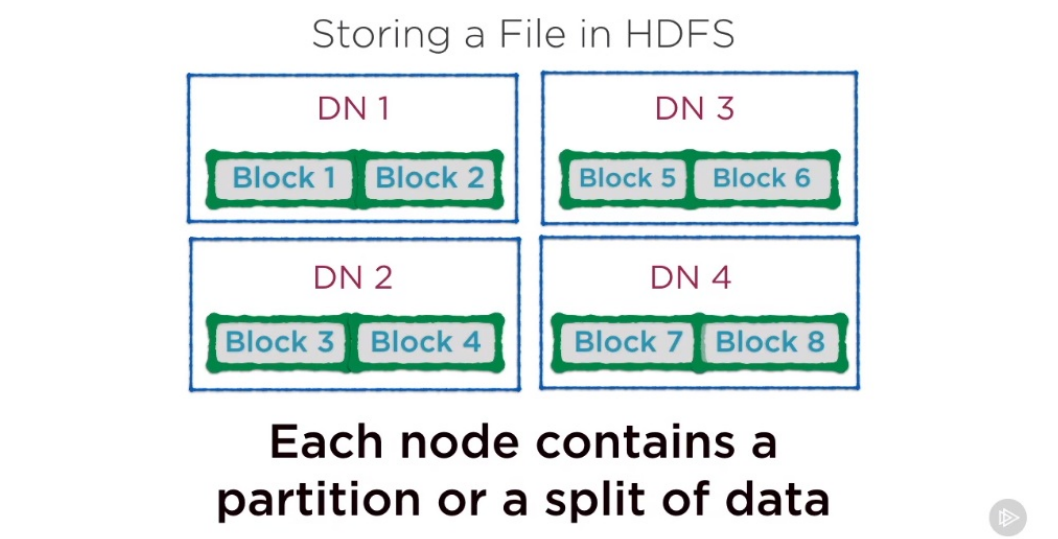
Hadoop Distributed file system is used to store and retrieve files. * HDFS is built on commodity hardware * Highly fault tolerant,hardware failure is the norm. * Suited to Batch processing data access has high throughput rather than low latency. * Supports very large data sets.

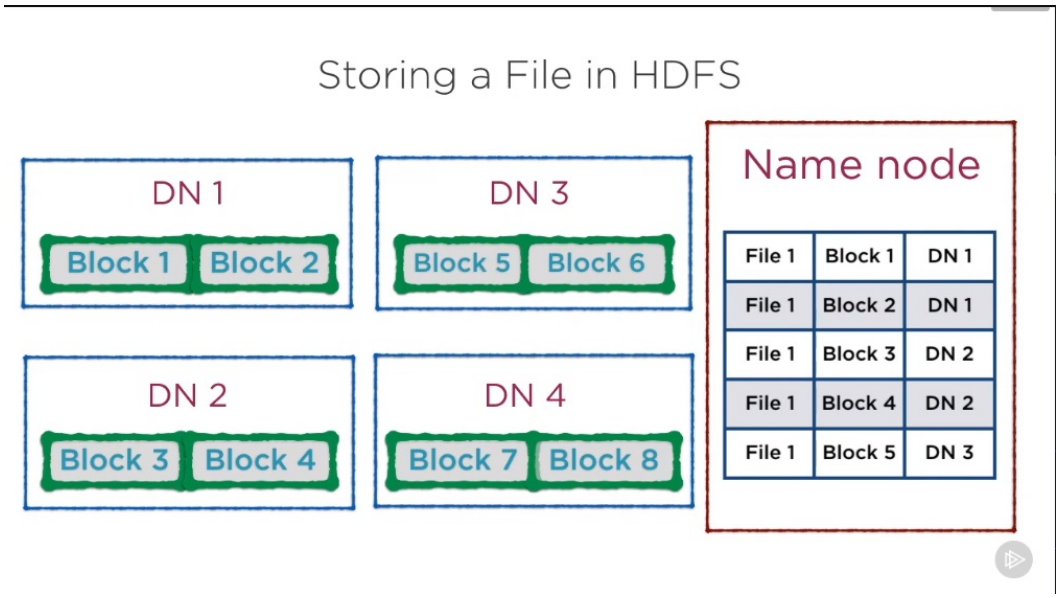
Master And Slave Nodes

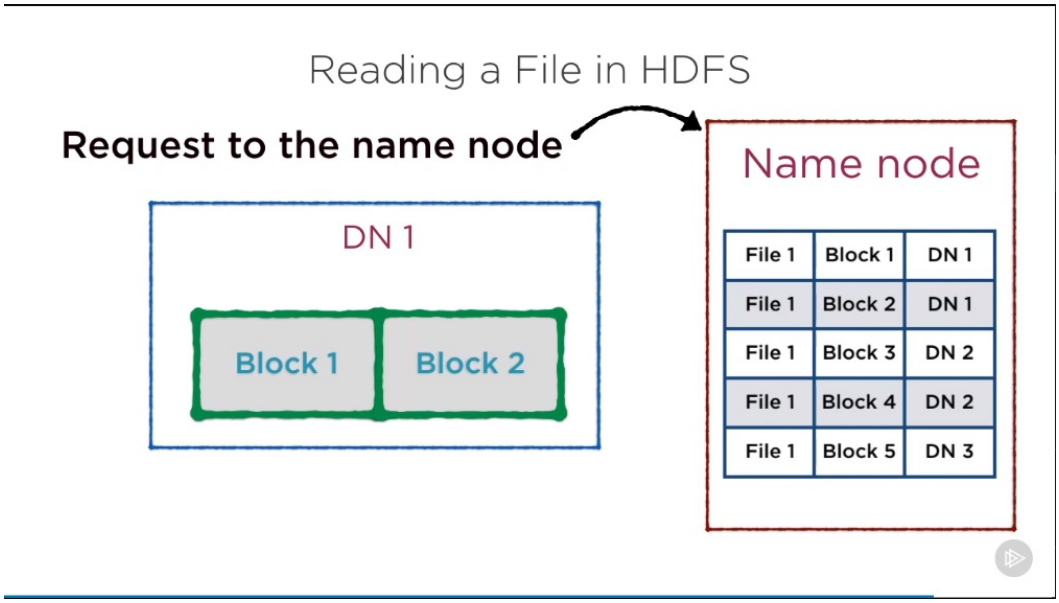
There are many nodes and hdfs choose among all a **Master** and all other **Slave**. Namenode is the master node which contains all metadata of all other slave nodes. Datanode is the node which physically stores the data.

Storing File in HDFS

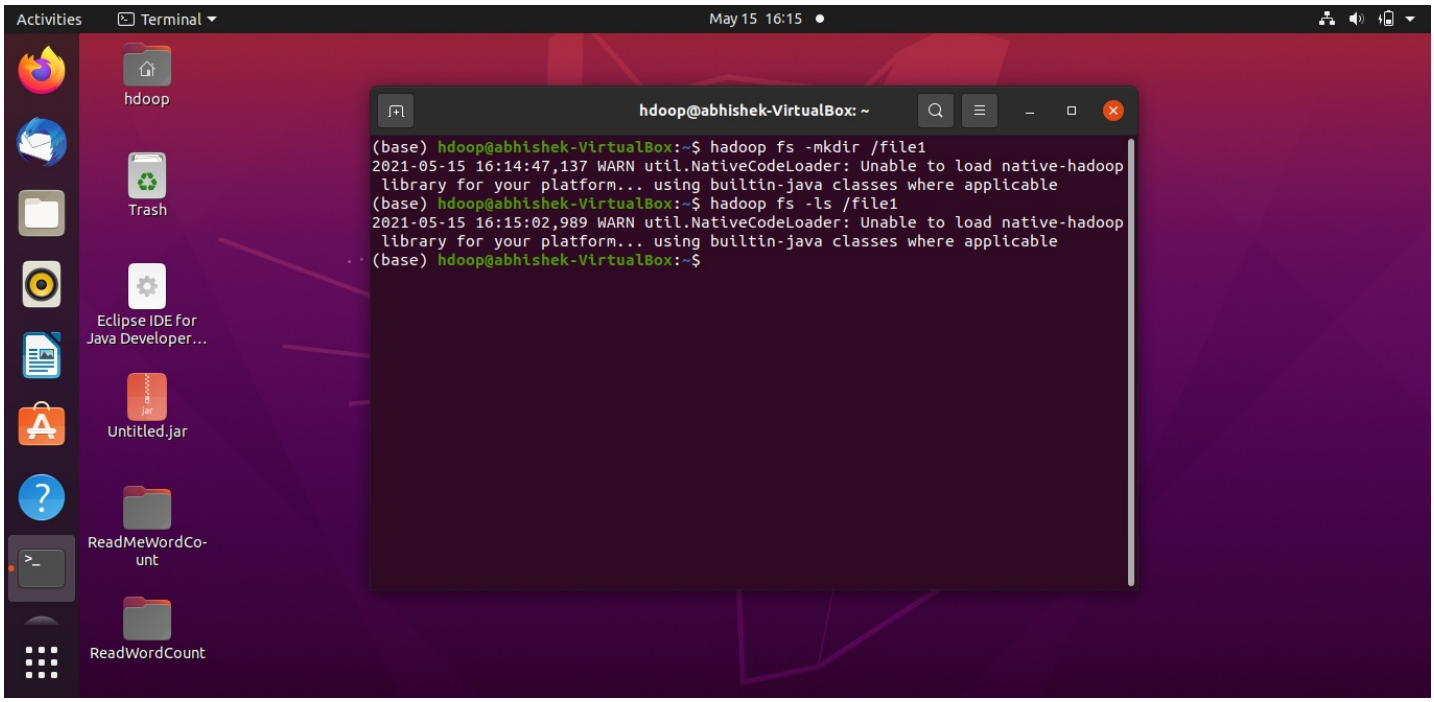
If there is a very large file say in Petabytes then file will split into blocks and these blocks are stored at different nodes. The size of each block is same 128 MB.



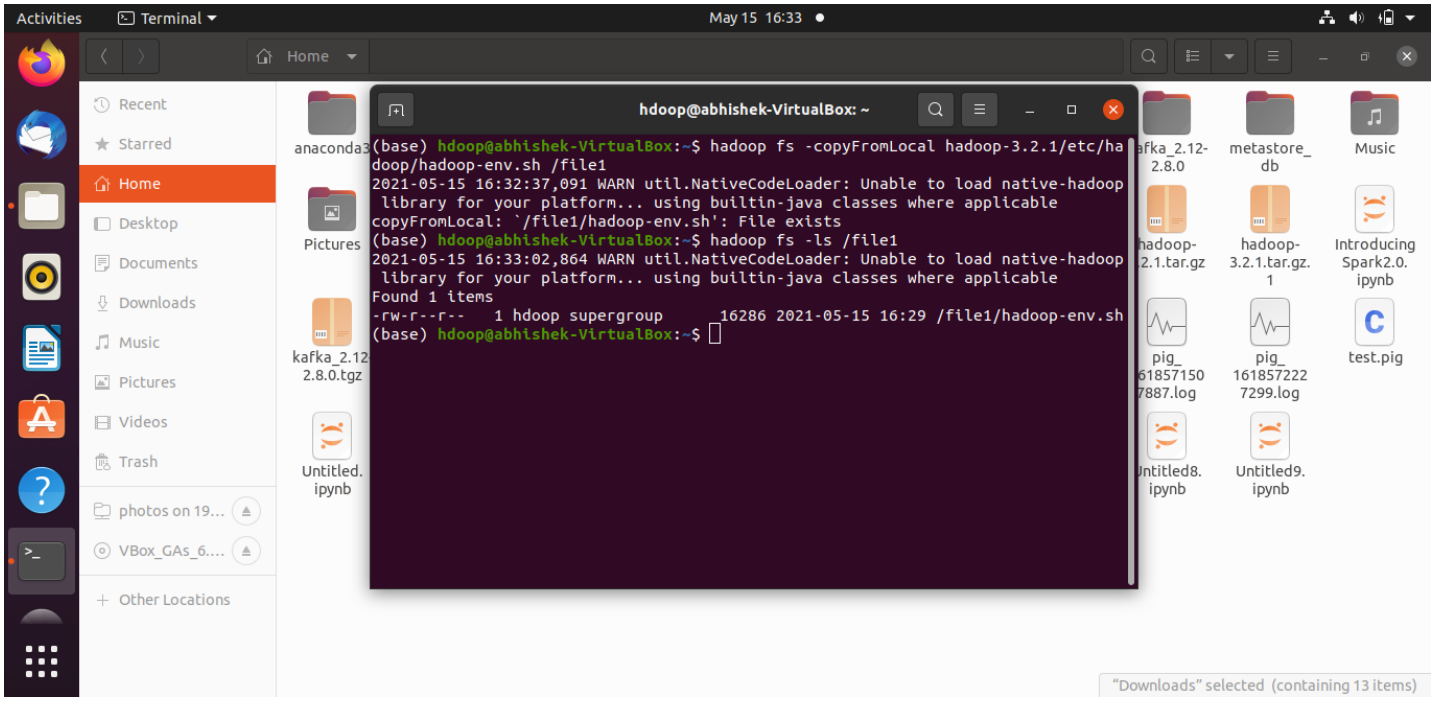




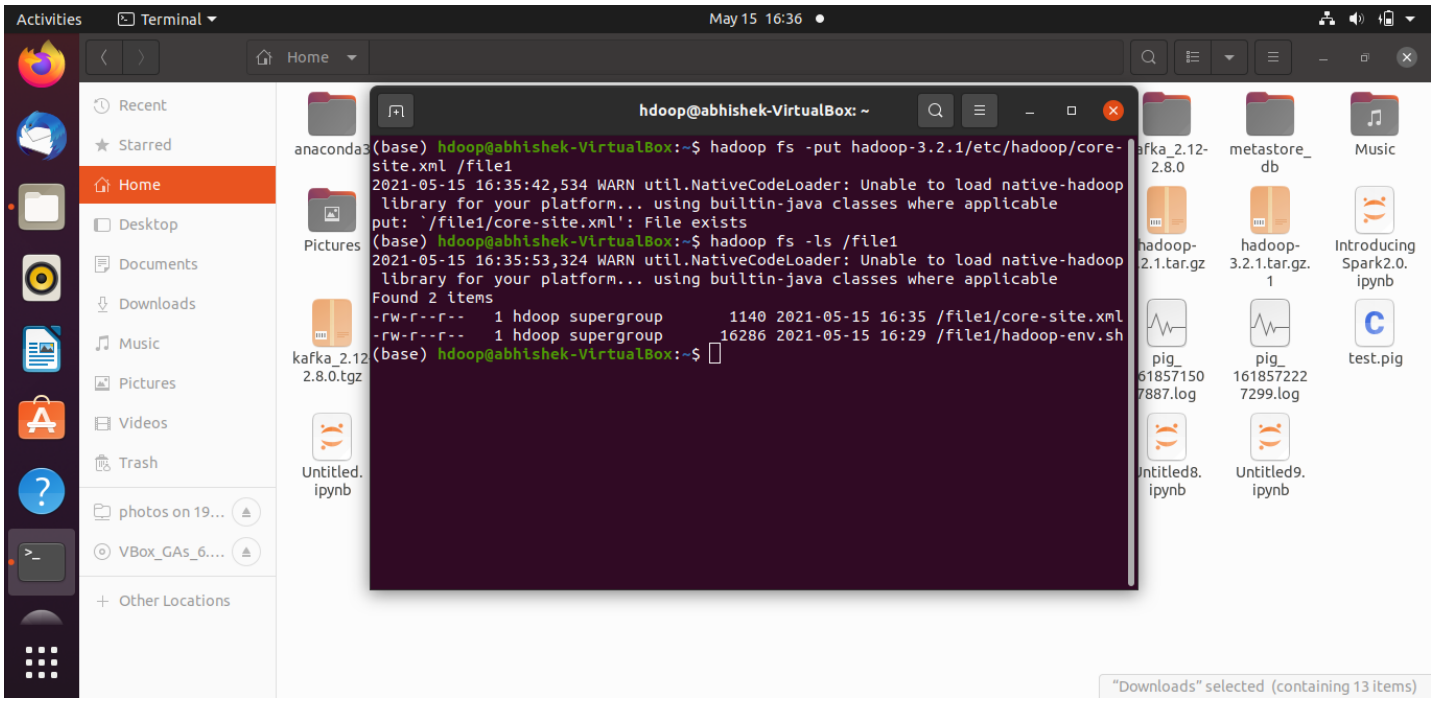
Step-1) Make the directory



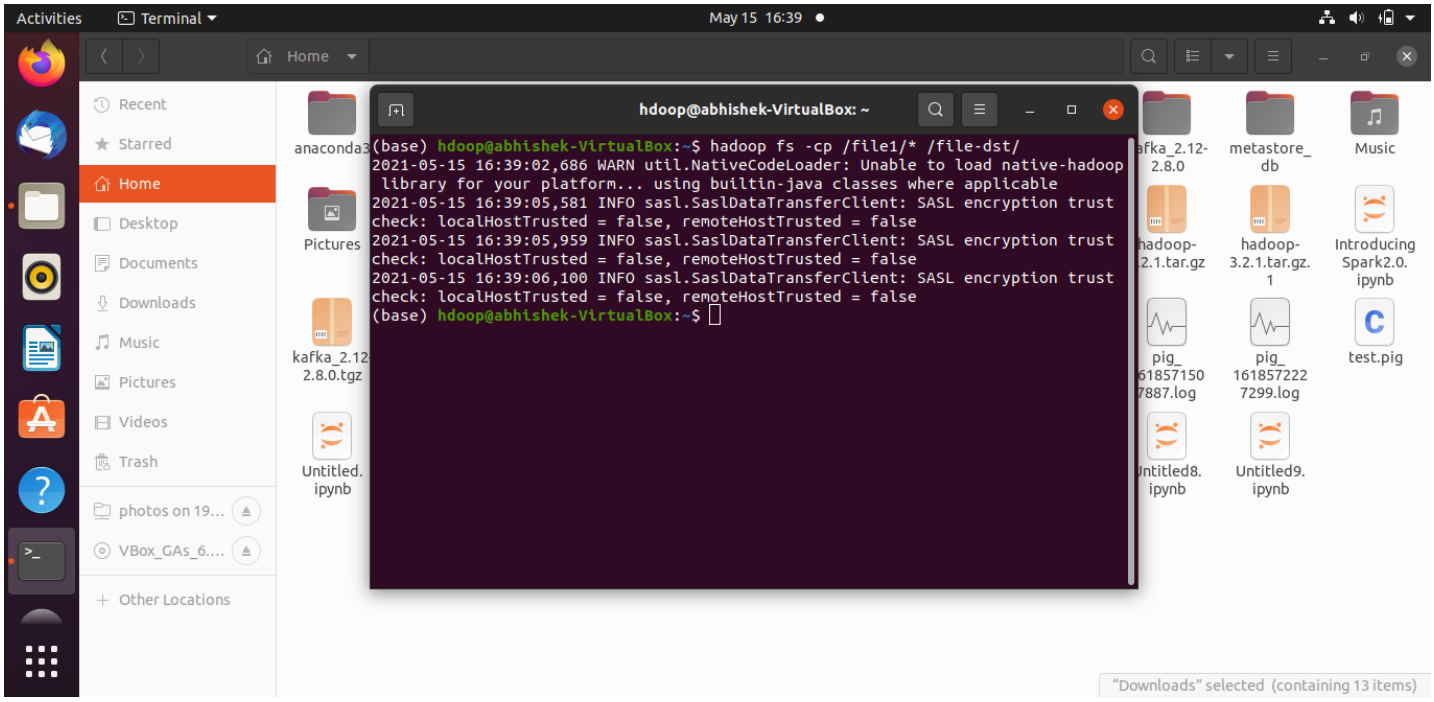
step-2) copy file from local to new directory



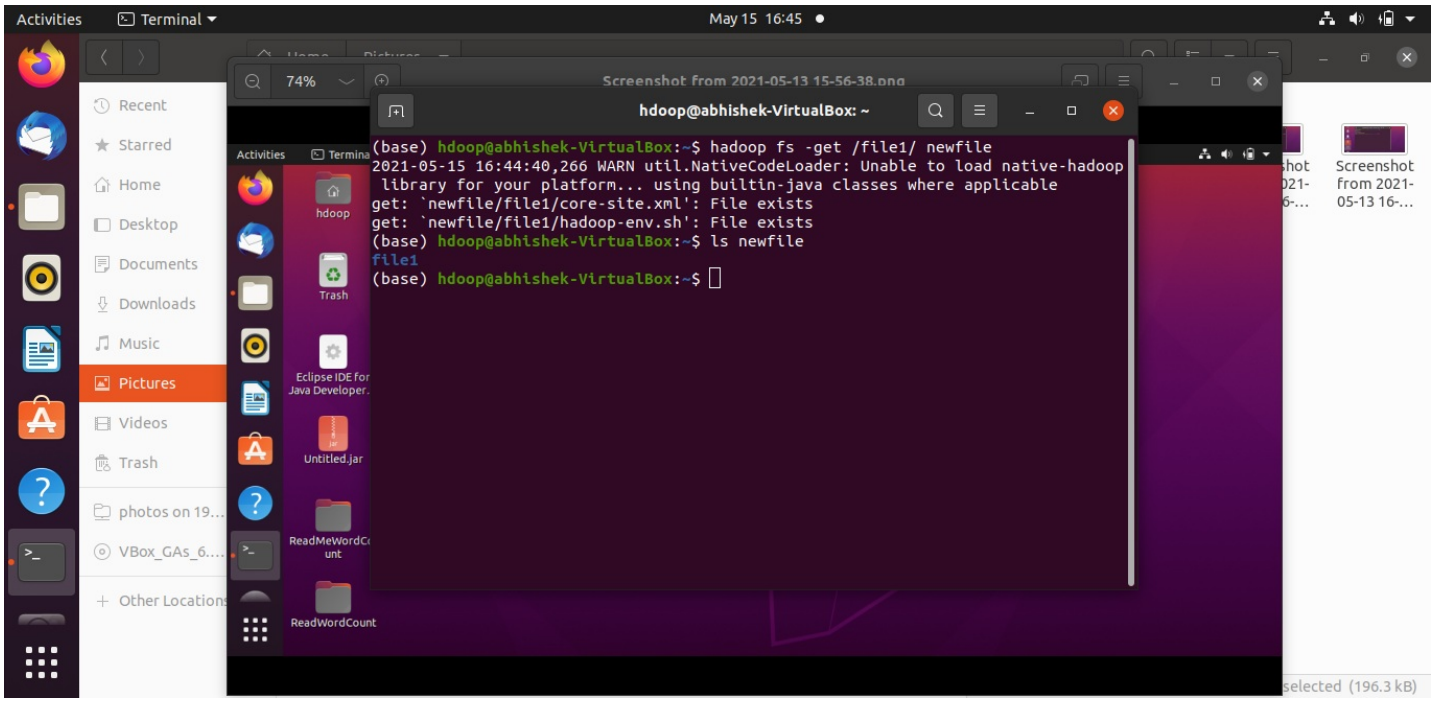
Another way to copy file



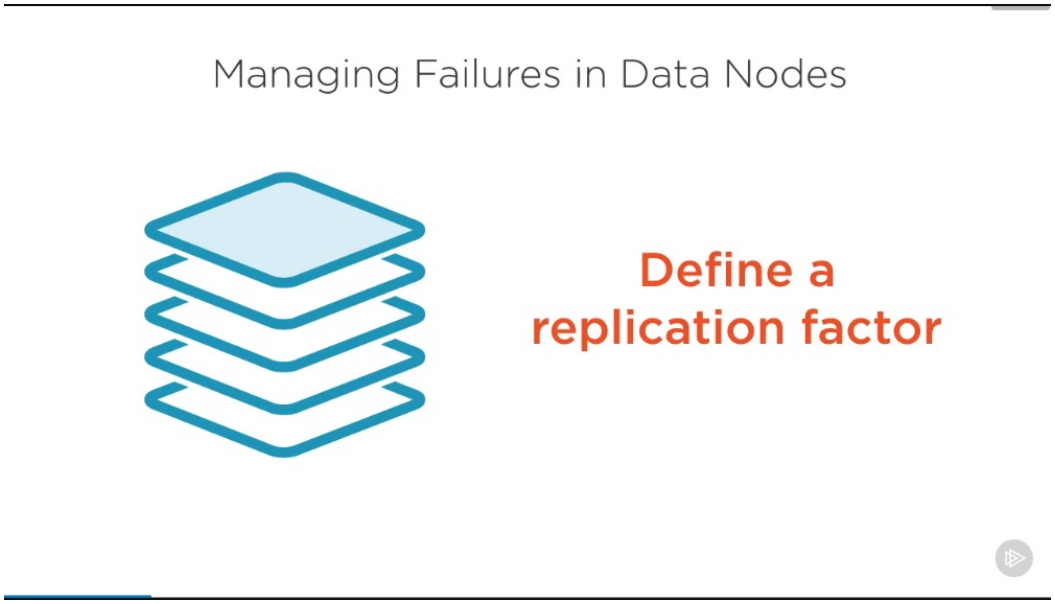
step-3) copy one file into new file

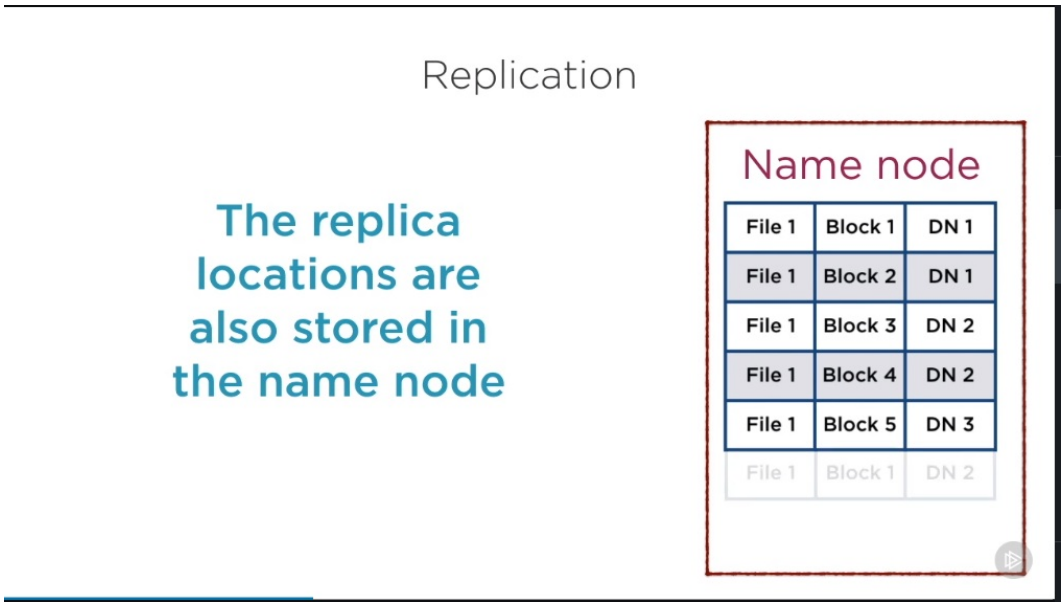


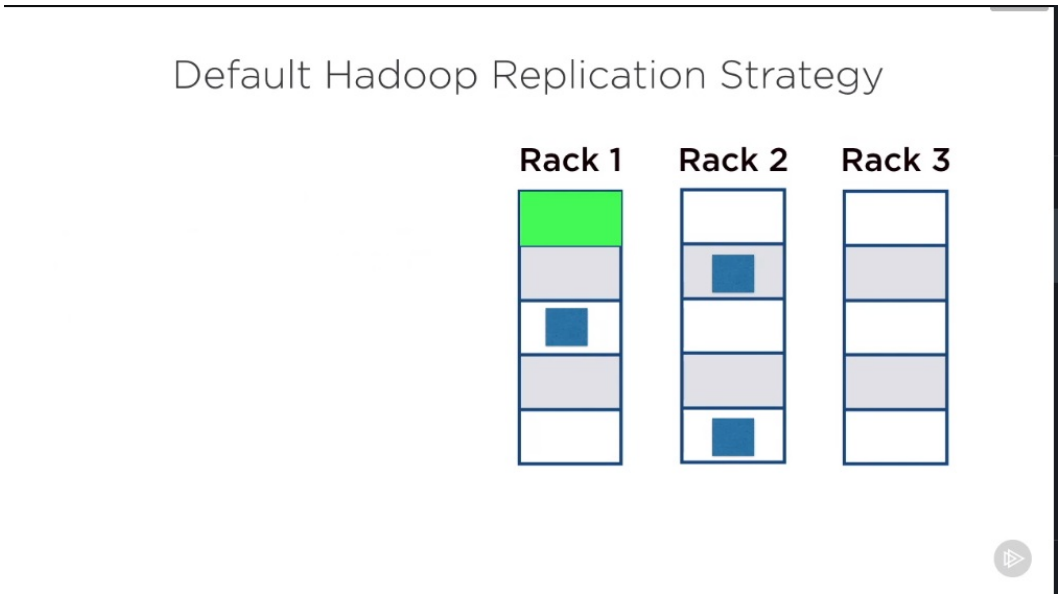
step-4) Get file from HDFS

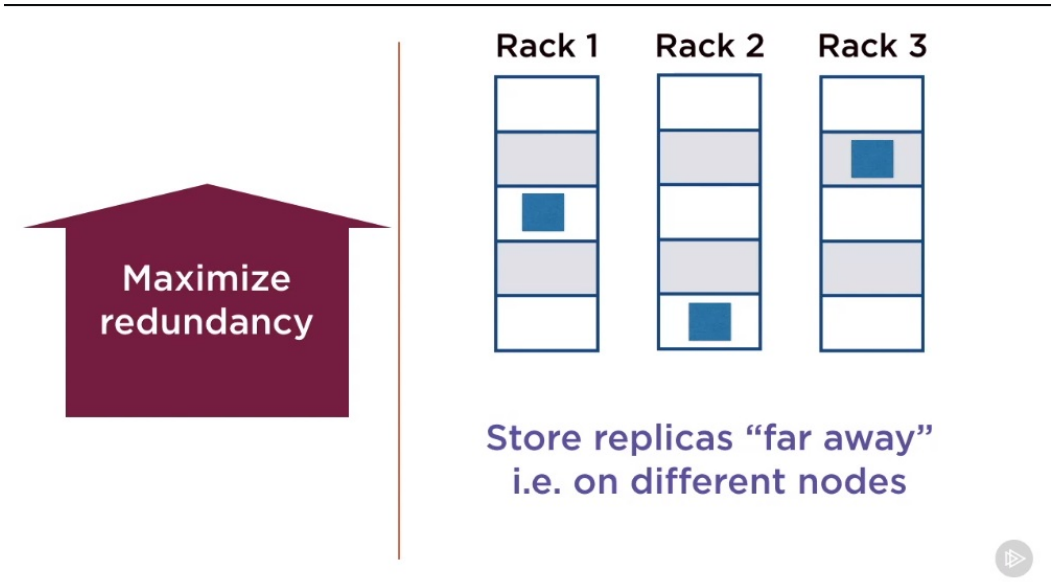


Fault Tolerance of Datanode



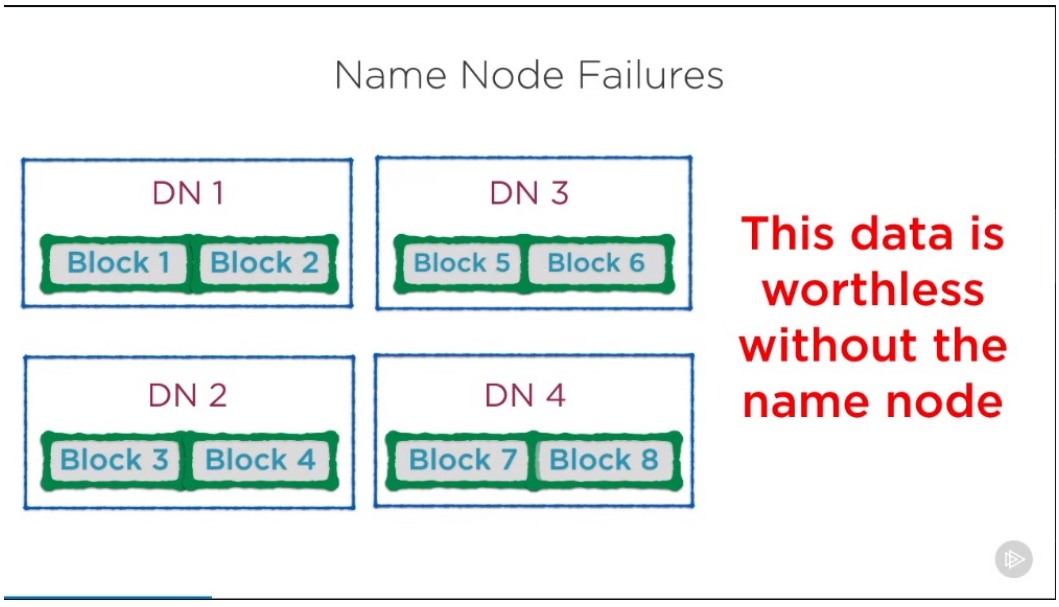






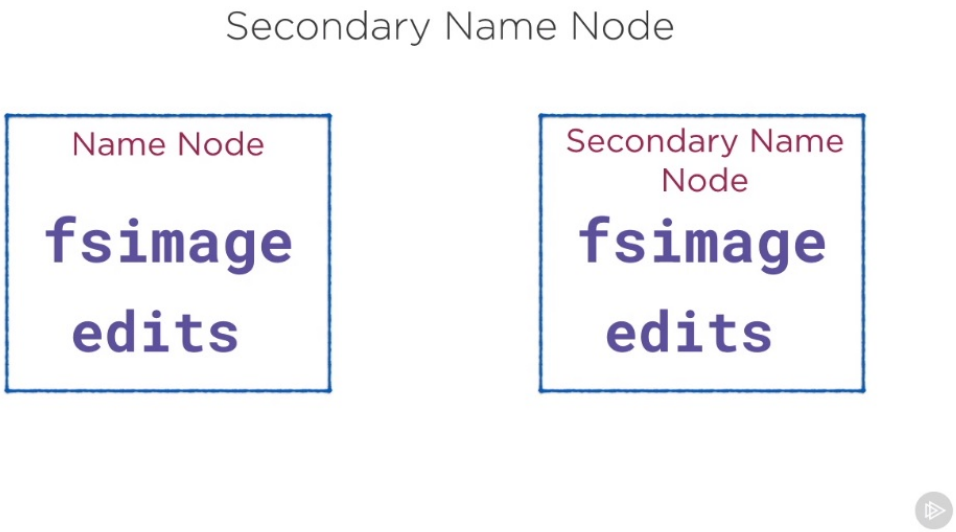
Fault Tolerance of Namenode

Namenode is the heart of the HDFS.



1. Metadata

2. Secondary Namenode



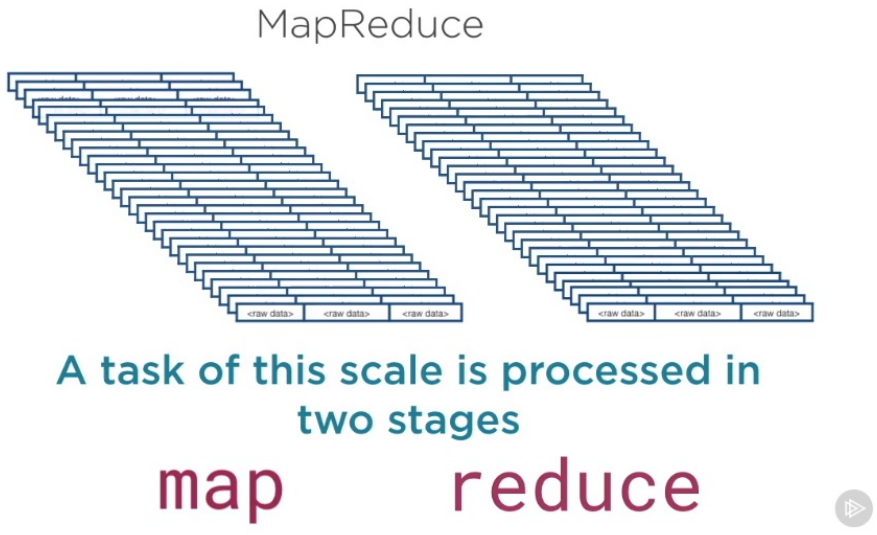
KO3: Able to Process data with MapReduce

1.What is MapReduce

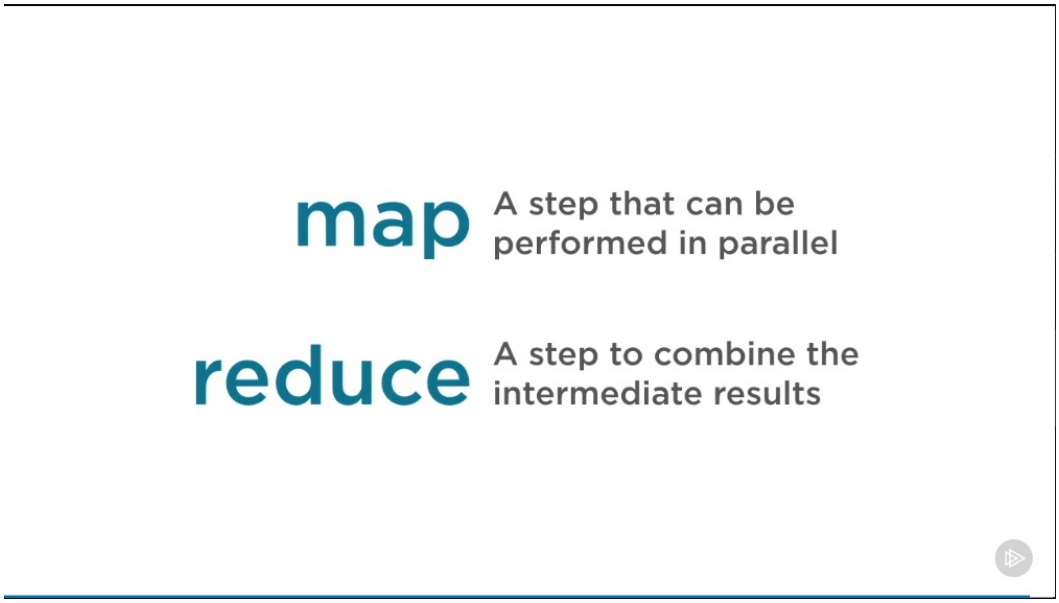
- It is a programming paradigm that follows Distributed Programming Approach.
- In today's world a lot of data is generated every minute. so, mapreduce process data in two phases.

1. **Map** Runs on multiple nodes of the clusters. So there are multiple Map.
2. **Reduce** Takes the output of the map and further process it to give final result.

So, for developer there are only two operations to be performed. Rest of the work is done by the Hadooop behind the scenes.

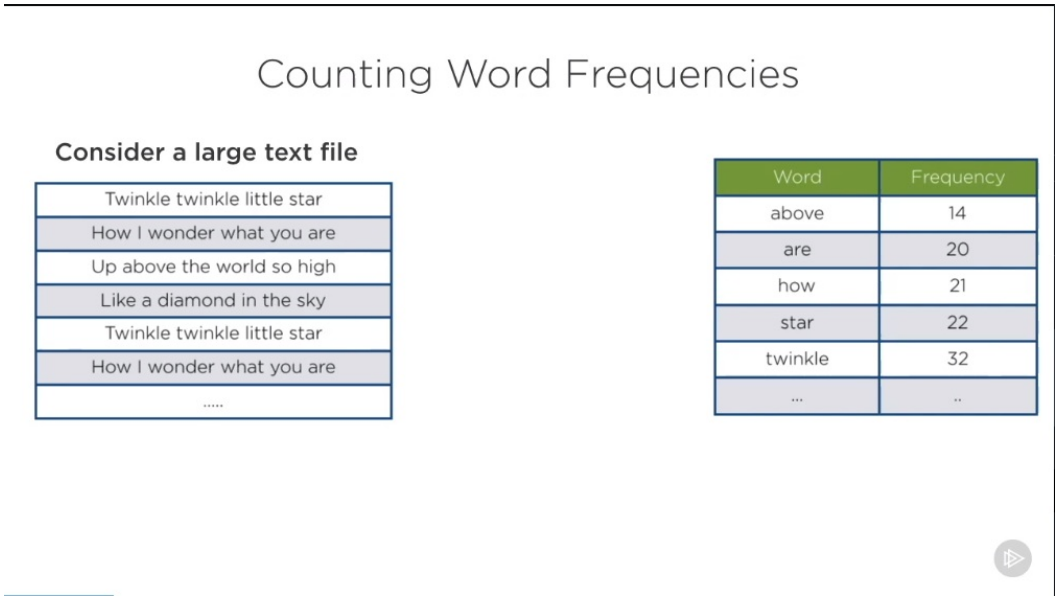


Map produces the output in the Key-Value and **Reduce** collects all Key-Value pairs from Map and produces result in a Meaningful way.

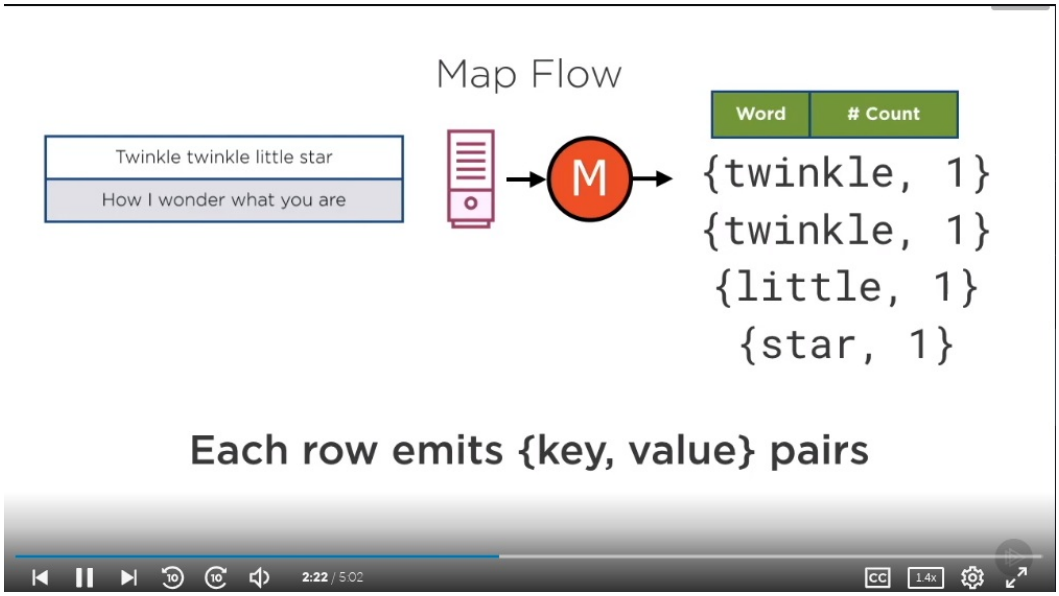


2.Data Flow in MapReduce

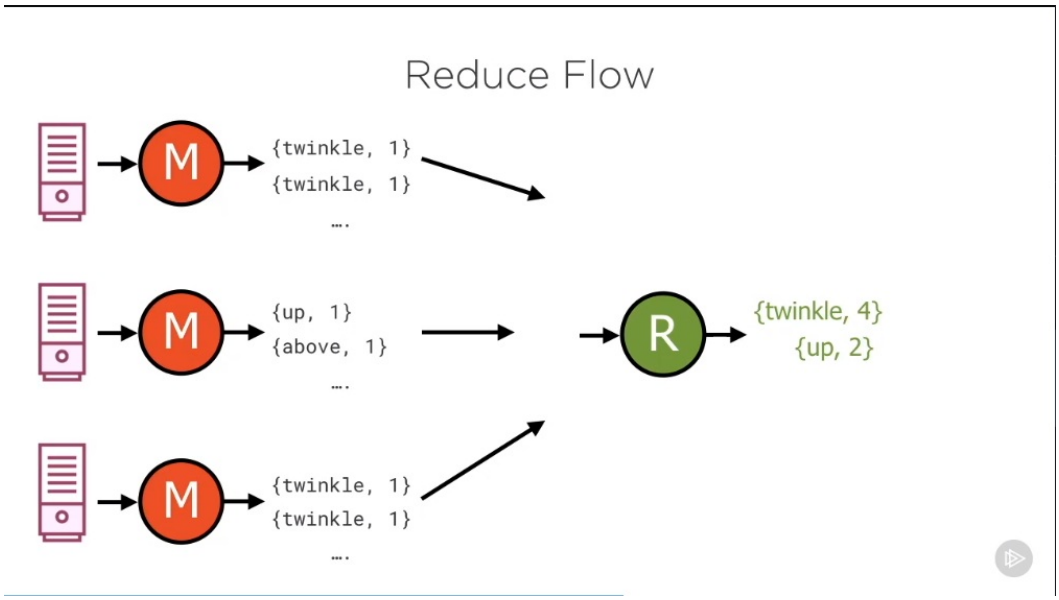
- MapReduce count the number of words from the file in petabytes.

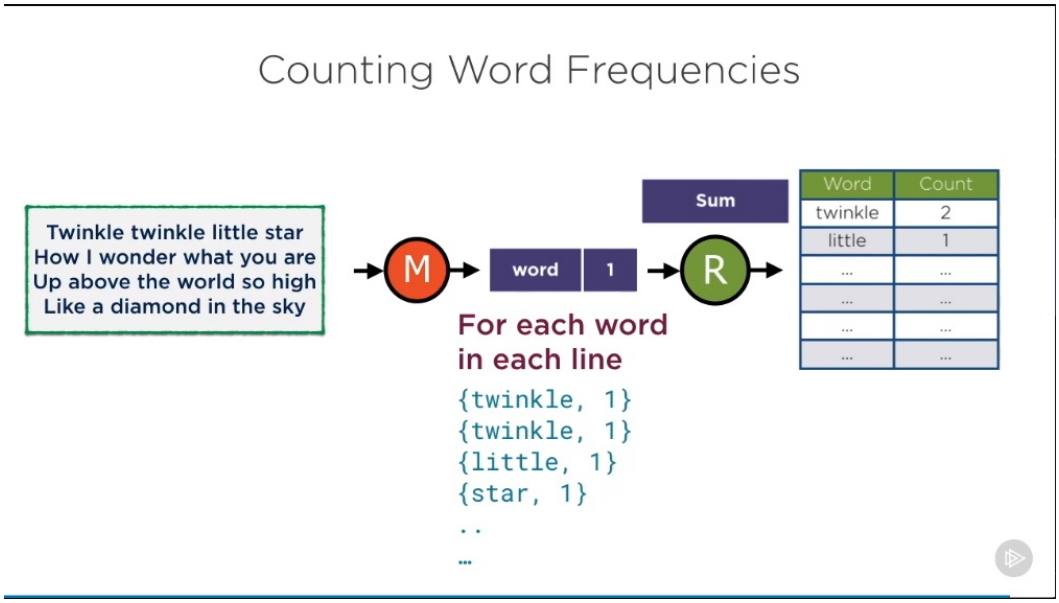


- In Map flow each row emits {Key,Value} pairs.



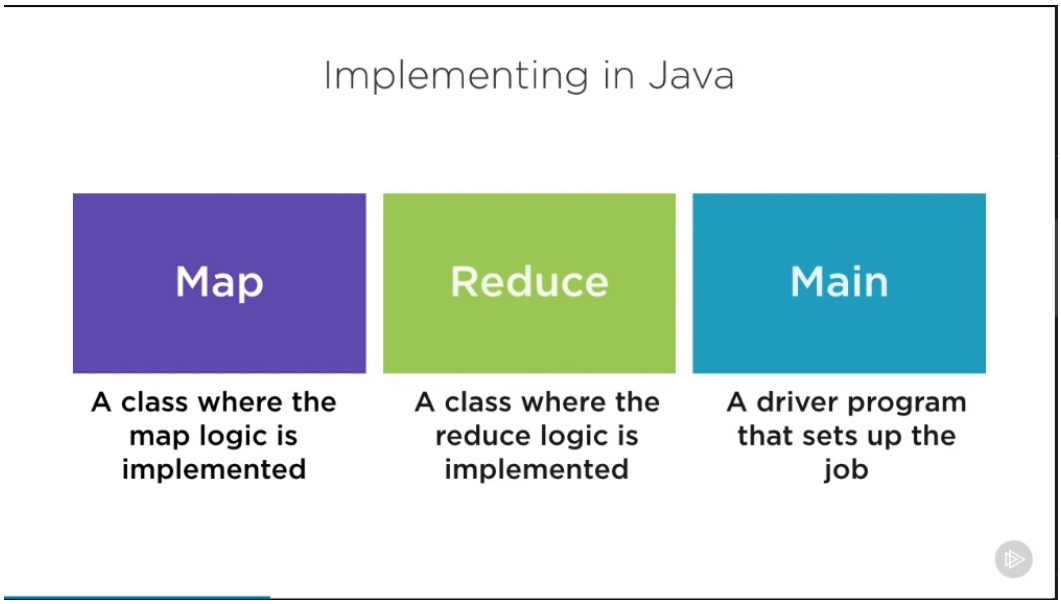
- In Reduce flow it produces the meaningful result.



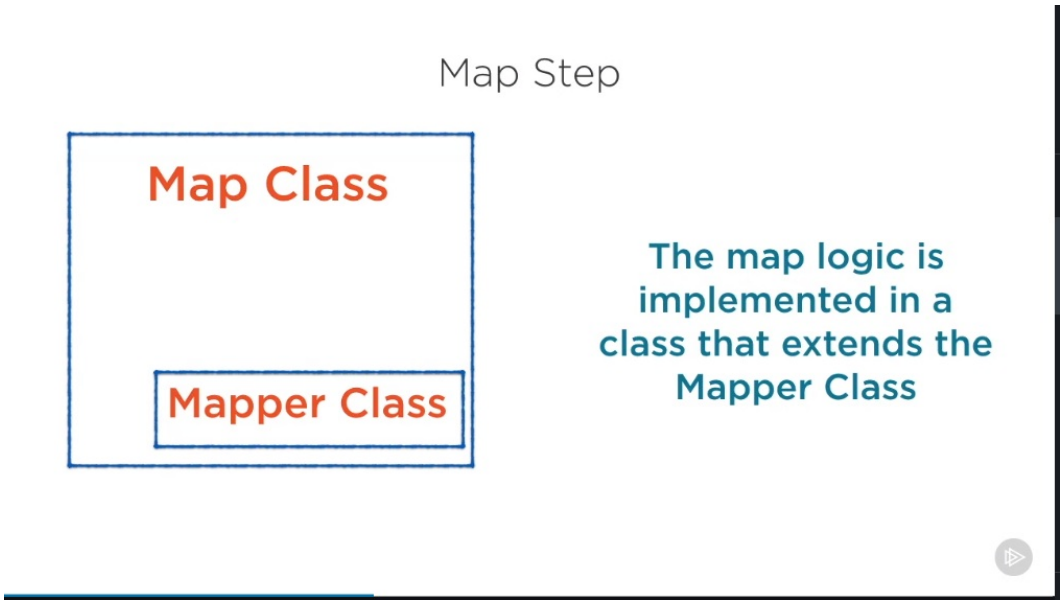


3.Implement MapReduce in java

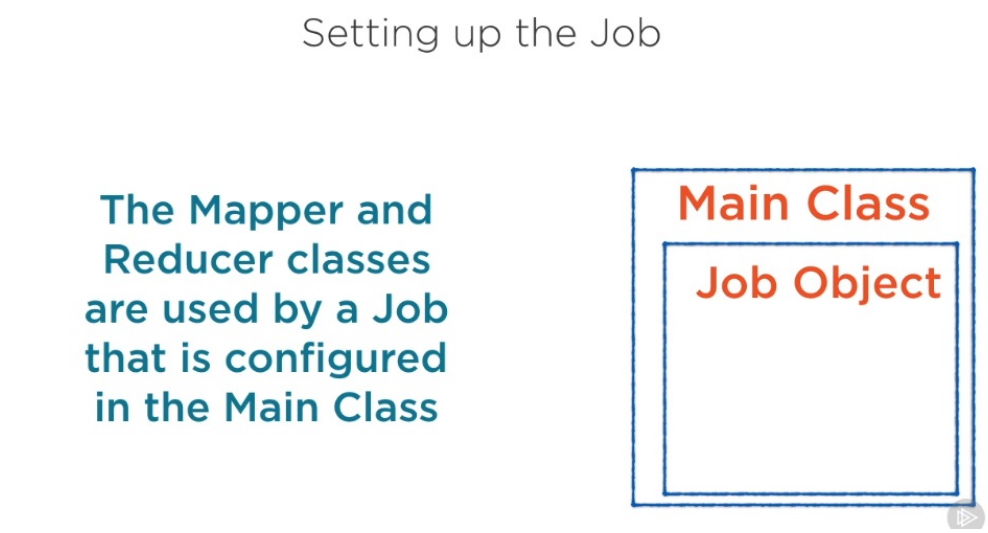
- It uses the Map and Reduce class where logic is implemented.



- Map Step



- Setting up the Job



4.Set up the Map Reduce

*Create the Map class

*Create the Reduce class

```java
package wordcount;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

import java.io.IOException;

public class Reduce extends Reducer<Text, IntWritable, Text, LongWritable> {

    @Override
    public void reduce(Text key,
                       Iterable<IntWritable> values,
                       Context context)
            throws IOException, InterruptedException {
        long count = 0;
        for (IntWritable value : values) {
            count++;
        }

        context.write(key, new LongWritable(count));
    }
}
```

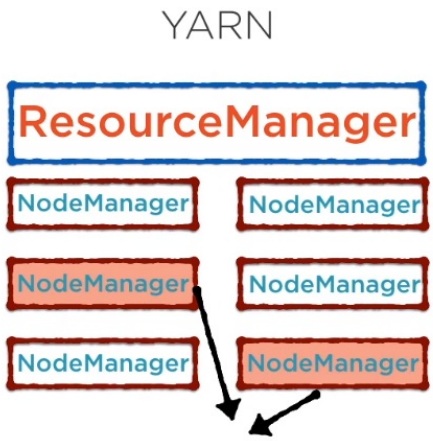
*Create the Main class

```java
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class Main extends Configured implements Tool {
    @Override
    public int run(String[] args) throws Exception{
        Job job = Job.getInstance(getConf());
        job.setJobName("wordcount");
        job.setJarByClass(Main.class);

        job.setMapOutputValueClass(IntWritable.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(LongWritable.class);

        job.setMapperClass(Map.class);
        job.setReducerClass(Reduce.class);

        Path inputFilePath = new Path(args[0]);
        Path outputFilePath = new Path(args[1]);

        FileInputFormat.addInputPath(job, inputFilePath);
        FileOutputFormat.setOutputPath(job, outputFilePath);

        return job.waitForCompletion(true) ? 0 : 1;
    }

    public static void main(String[] args) throws Exception {
        int exitCode = ToolRunner.run(new Main(), args);
        System.exit(exitCode);
    }
}
```

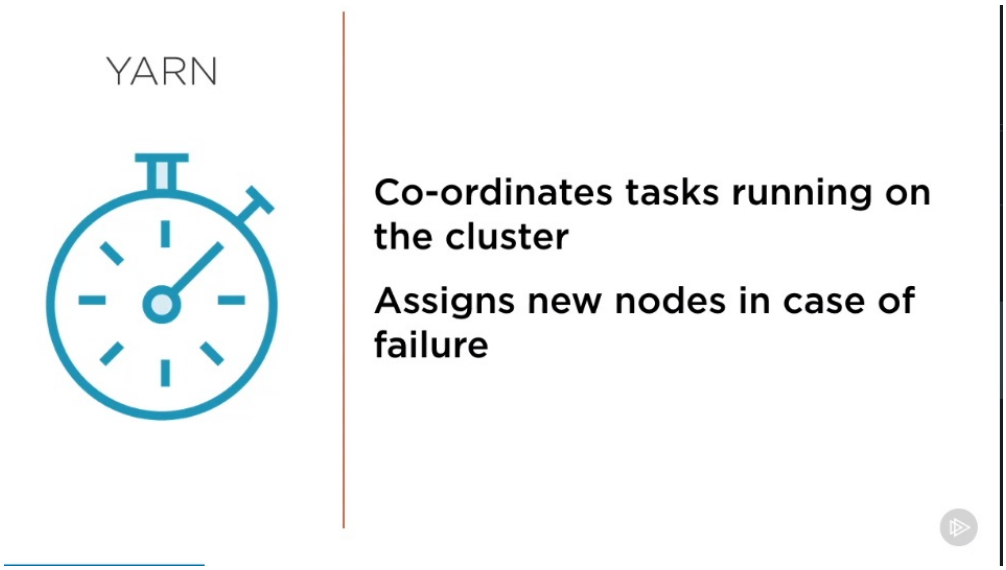
KO4: Able to Schedule and Manage data with MapReduce

YARN

In 2013, hadoop architecture was changed and MapReduce was divided into two MapReduce AND Yarn. * MapReduce used for what operatons we want to define. * Yarn was responsible how these operations run across cluster.



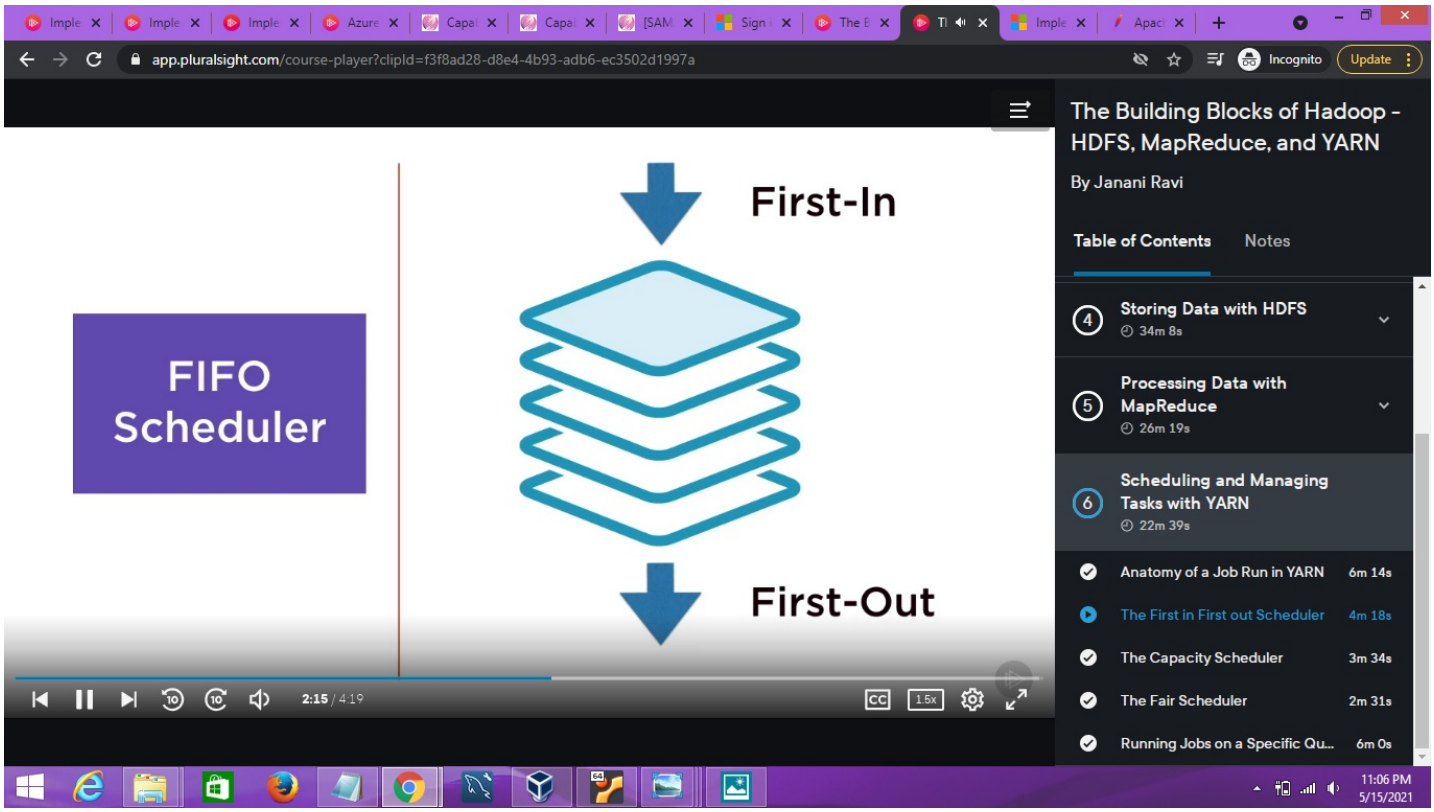
It has two components : 1. Resource Manager 2. Node Manager

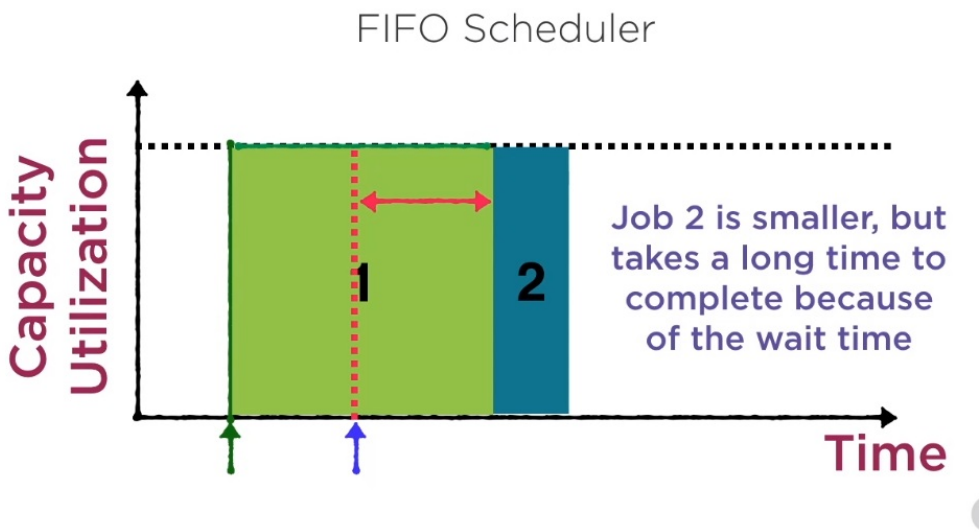


There are Three Scheduling Policies

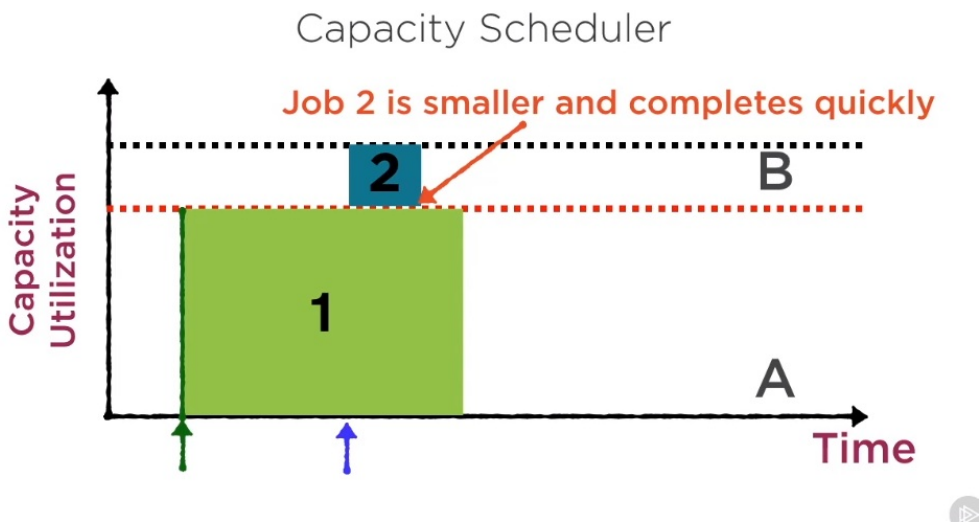
1. FIFO
2. Capacity
3. Fair

** FIFO Scheduler**





** Capacity Scheduler**



** Fair Scheduler