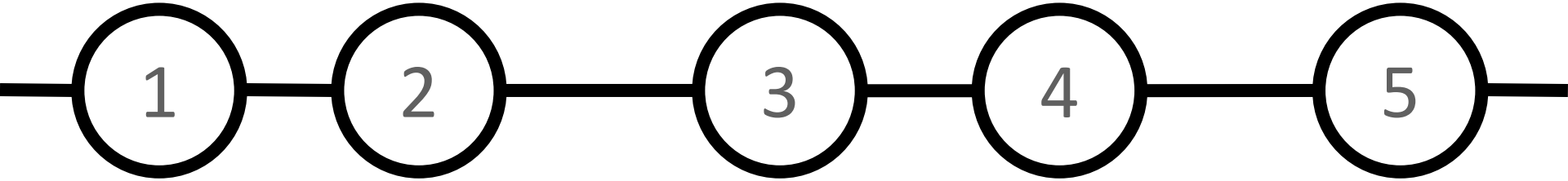
Part 3

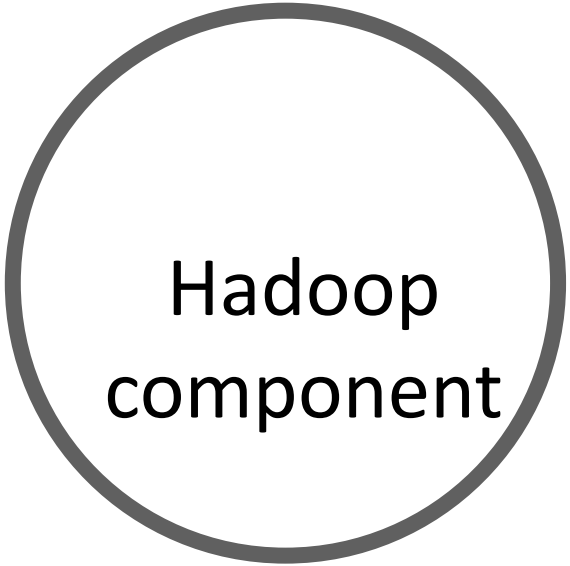
Ahmed Ramadan, ary00@fayoum.edu.eg

AGENDA



1. Hadoop Components
2. Map-Reduce
3. Word Count Example
4. Run Word Count
5. End
   Questions

Ahmed Ramadan, ary00@fayoum.edu.eg



Ahmed Ramadan, ary00@fayoum.edu.eg

Hadoop v1.0

MapReduce
Data Processing
& Resource Management

HDFS
Distributed File Storage

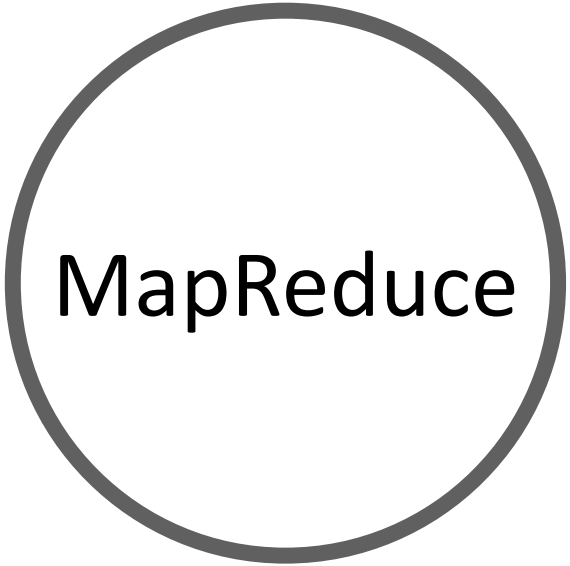
Hadoop v2.0

MapReduce

Other Data Processing Frameworks

YARN
Resource Management

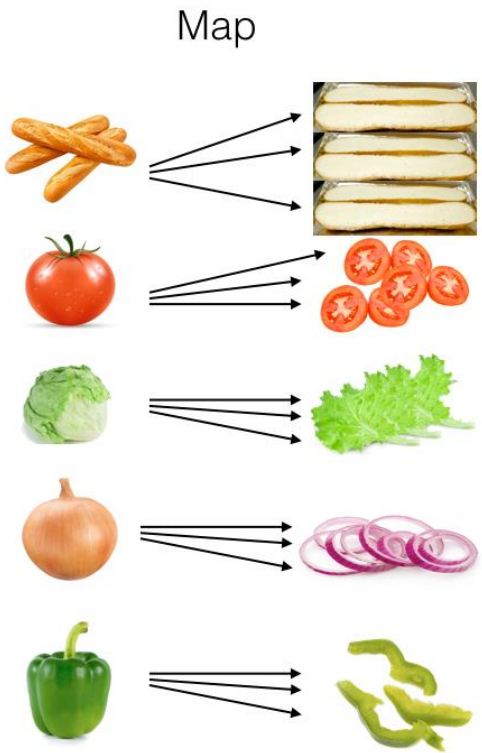
HDFS
Distributed File Storage



Map Reduce Model

Ahmed Ramadan, ary00@fayoum.edu.eg

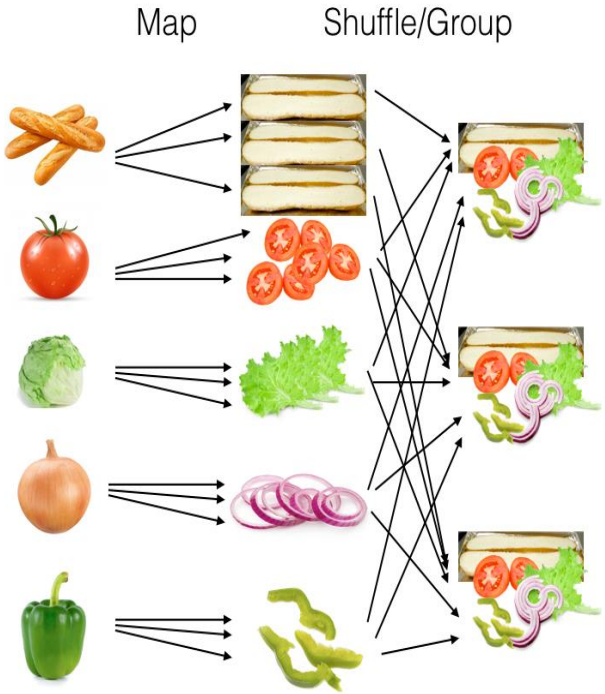
Map



We distribute our raw ingredients amongst the **workers** in our shop. One person takes the tomatoes, one person takes the lettuce, one person takes the onions, and so on. We'll call this the "map" stage.

Ahmed Ramadan, ary00@fayoum.edu.eg

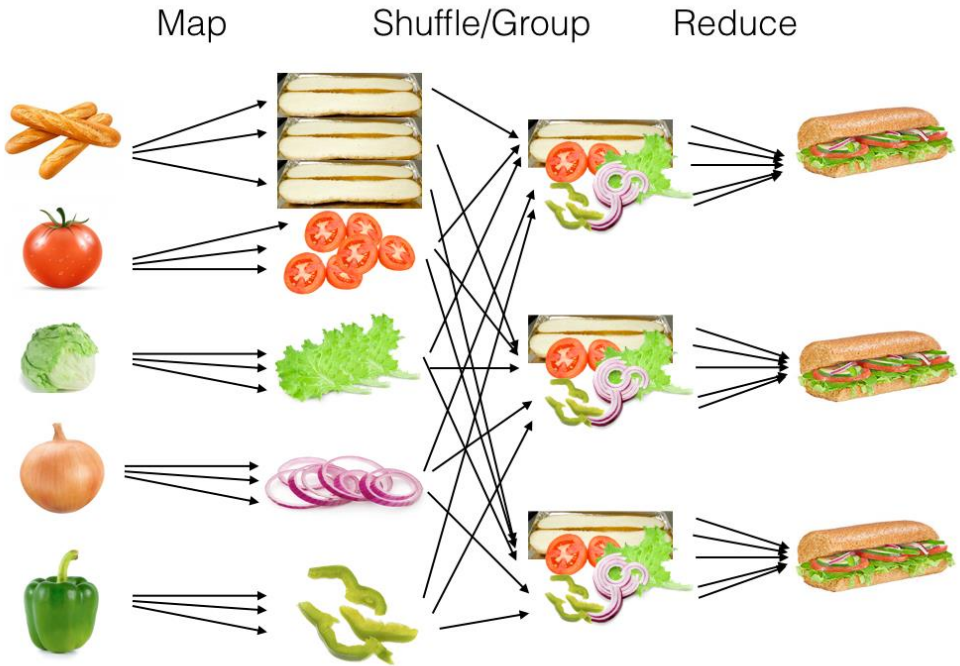
shuffle/group



Next, we take these processed ingredients (which we'll call "mapper intermediates") and group them together into piles, so that making a sandwich becomes easy. **We'll call this the "shuffle/group" stage.**

Ahmed Ramadan, ary00@fayoum.edu.eg

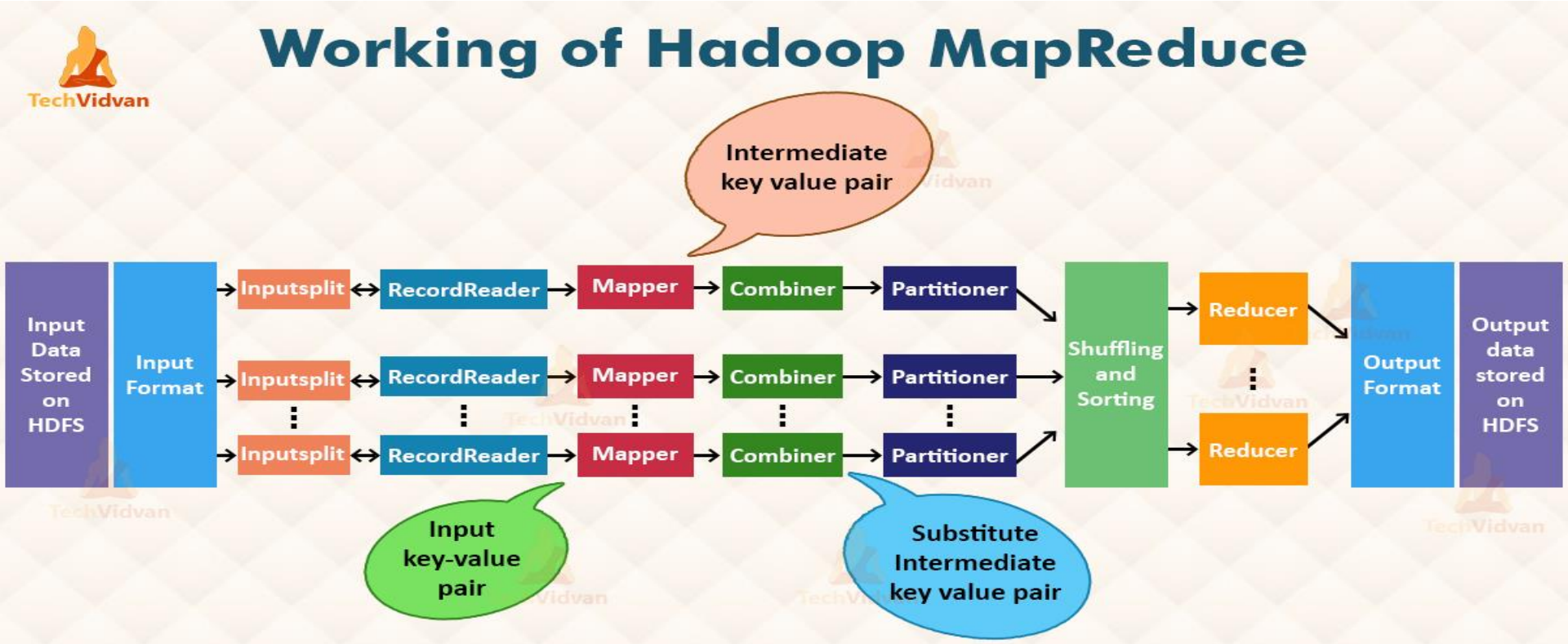
Reduce



Finally, we'll combine the ingredients into a sandwich. **We'll call this the "reduce" stage.**

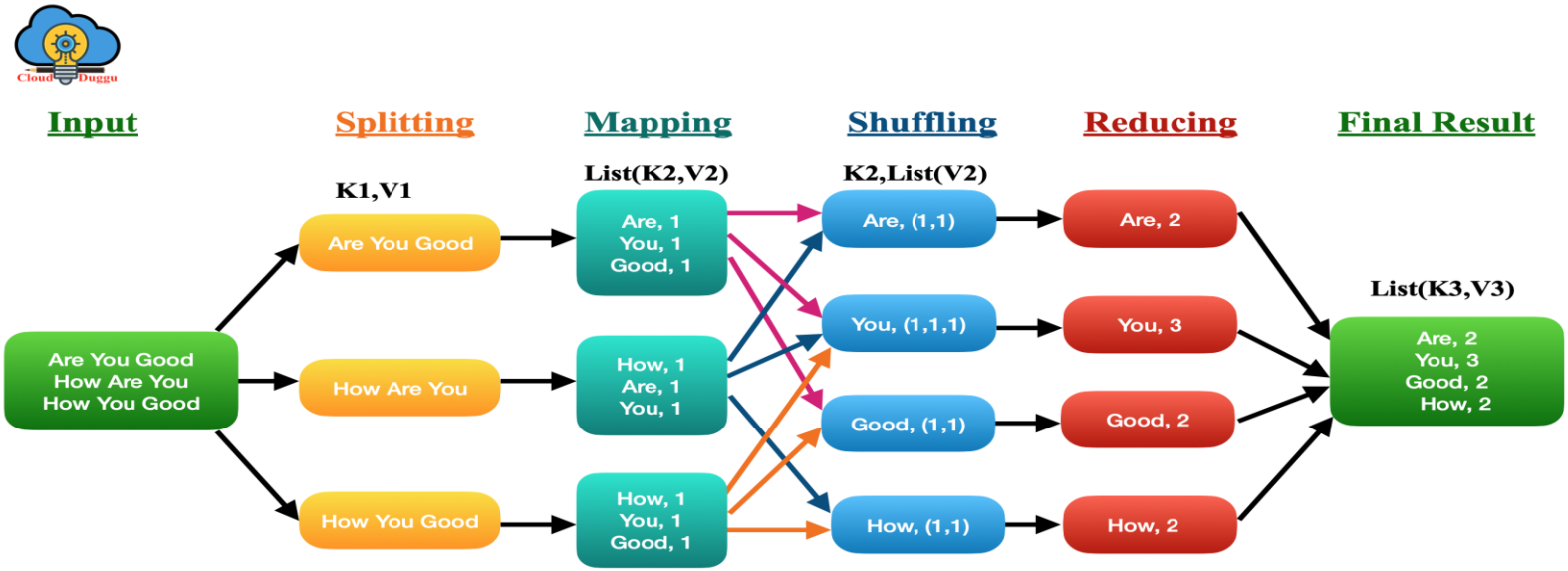
Ahmed Ramadan, ary00@fayoum.edu.eg

Overall stages



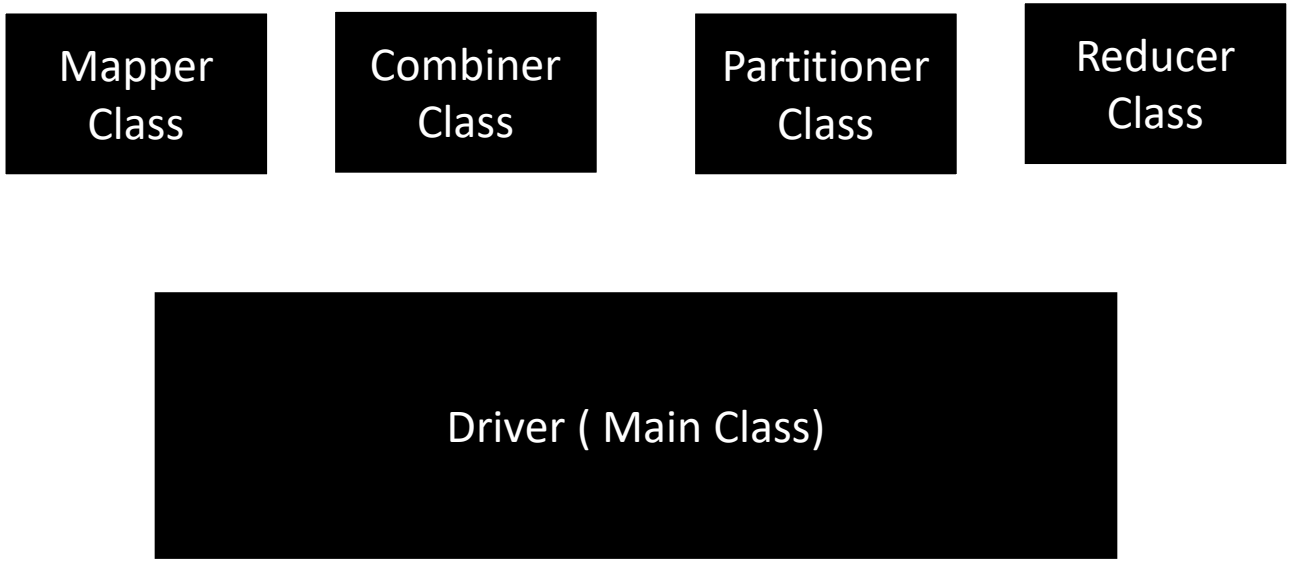
Ahmed Ramadan, ary00@fayoum.edu.eg

Wordcount Example



Ahmed Ramadan, ary00@fayoum.edu.eg

Hadoop Client Job



Ahmed Ramadan, ary00@fayoum.edu.eg

Driver

- The code that runs on the client machine configures the **job** details by creating an object from the Job **class**, which implements the **JobContext** interface.

```
Configuration conf = new Configuration();
Job job = Job.getInstance(conf, “Job Name");
```

- It submit the job to cluster.

- It parse job argument to identify job parameters for example : input & output directory.

Ahmed Ramadan, ary00@fayoum.edu.eg

Driver Job Configuration:

- It submit the job to cluster. The job object allow you to set configuration for your M/R Job.
- Configure Map per , Combiner ,Partitioner ,Reducer classes.

```
job.setMapperClass(MapperSide.class);
job.setReducerClass(ReducerSide.class);
```

- Set Input /Output [Key- Value] data types for each Mapper & Reducer.

```
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
```

- Configure input & output directory.

```
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
```


Ahmed Ramadan, ary00@fayoum.edu.eg

Keys and Values

- Keys and Values in Hadoop are objects not primitive data types.
- Values are Objects which implement **Writable**.
- Keys are Objects which implement **WritableComparable**. **[Sorting]**
- **int** in Java Match IntWritable , **string** java is Text in Hadoop.

Ahmed Ramadan, ary00@fayoum.edu.eg

Mapper

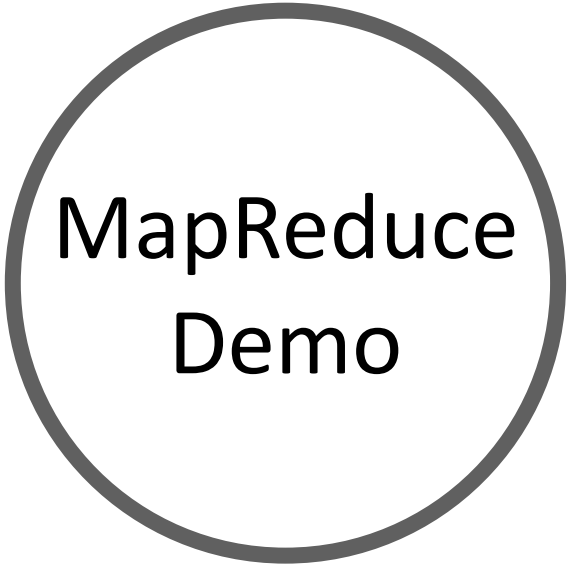
- The mapper class deals with a single input **split**(block).
- All mapper classes must extend the **Mapper base class**.
- All mapper must specify the **key and values** for input and output.
- All mappers must **override** the **map** method and pass the key, value, and Context.
- The **context** is used to **write** the intermediate data and all information about the job conf

Ahmed Ramadan, ary00@fayoum.edu.eg

Reducer

- The Reducer receives a **Key and an iterable** collection of Writable objects.
- It also receives a Context object.
- All reducers classes must extend **the Reducer** base class.
- All **Reducer** must specify the key and values for intermediate input and final (or intermediate) output.
- All Reducer must override **reduce** method and pass the key , iterable and context.

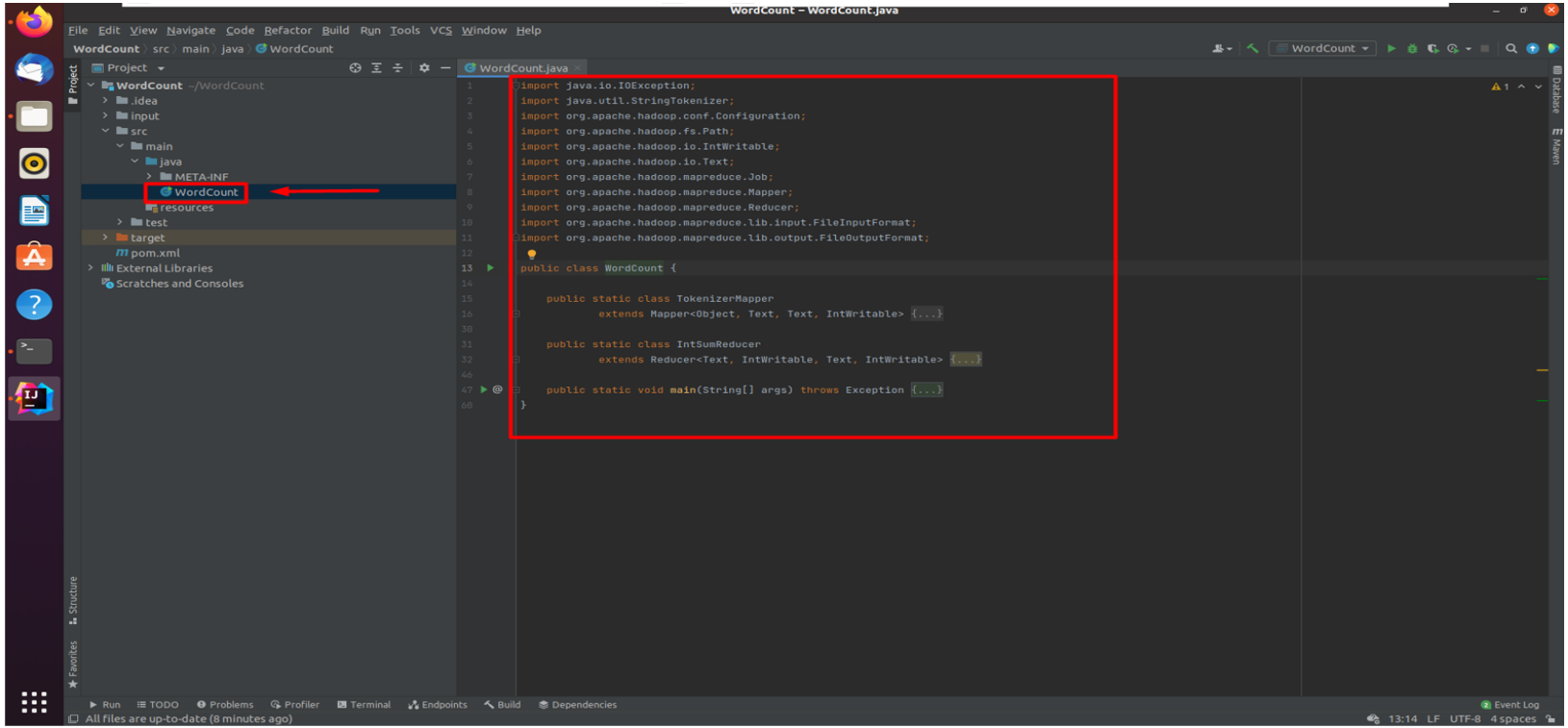
Ahmed Ramadan, ary00@fayoum.edu.eg



Run MapReduce Job Locally

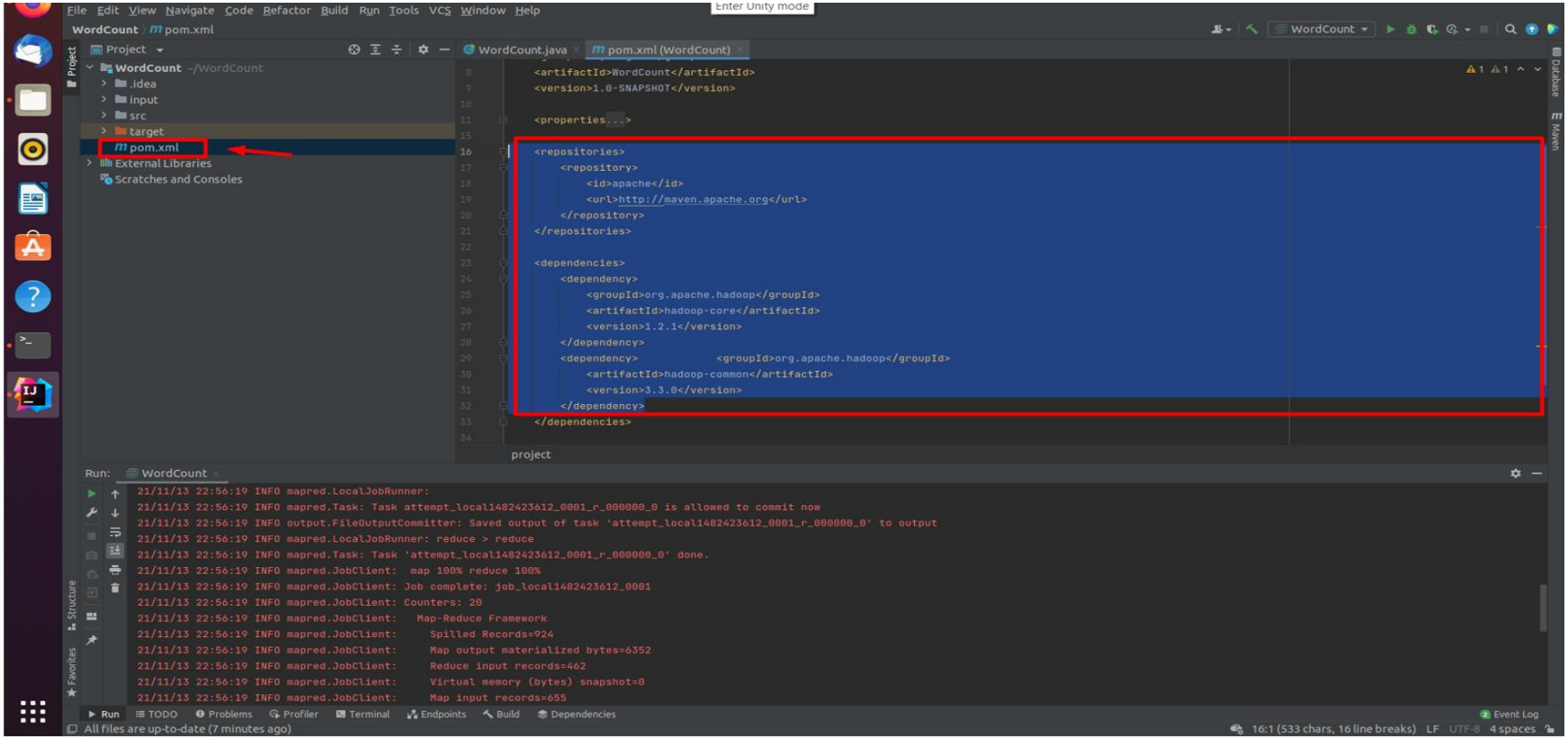
Ahmed Ramadan, ary00@fayoum.edu.eg

Create Class WordCount.java



Ahmed Ramadan, ary00@fayoum.edu.eg

Edit pom.xml File (add repo , dependencies)

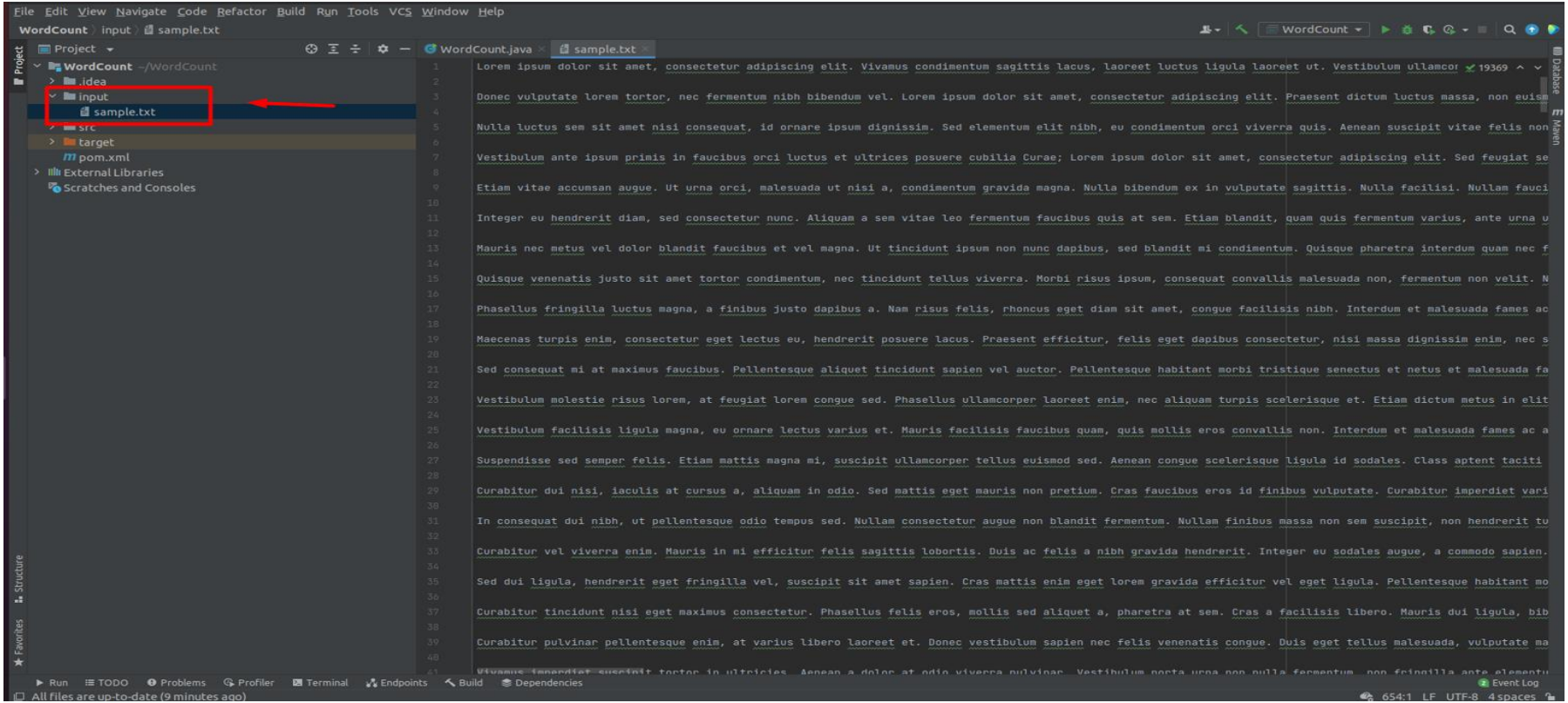


Edit pom.xml File (add repo , dependencies)

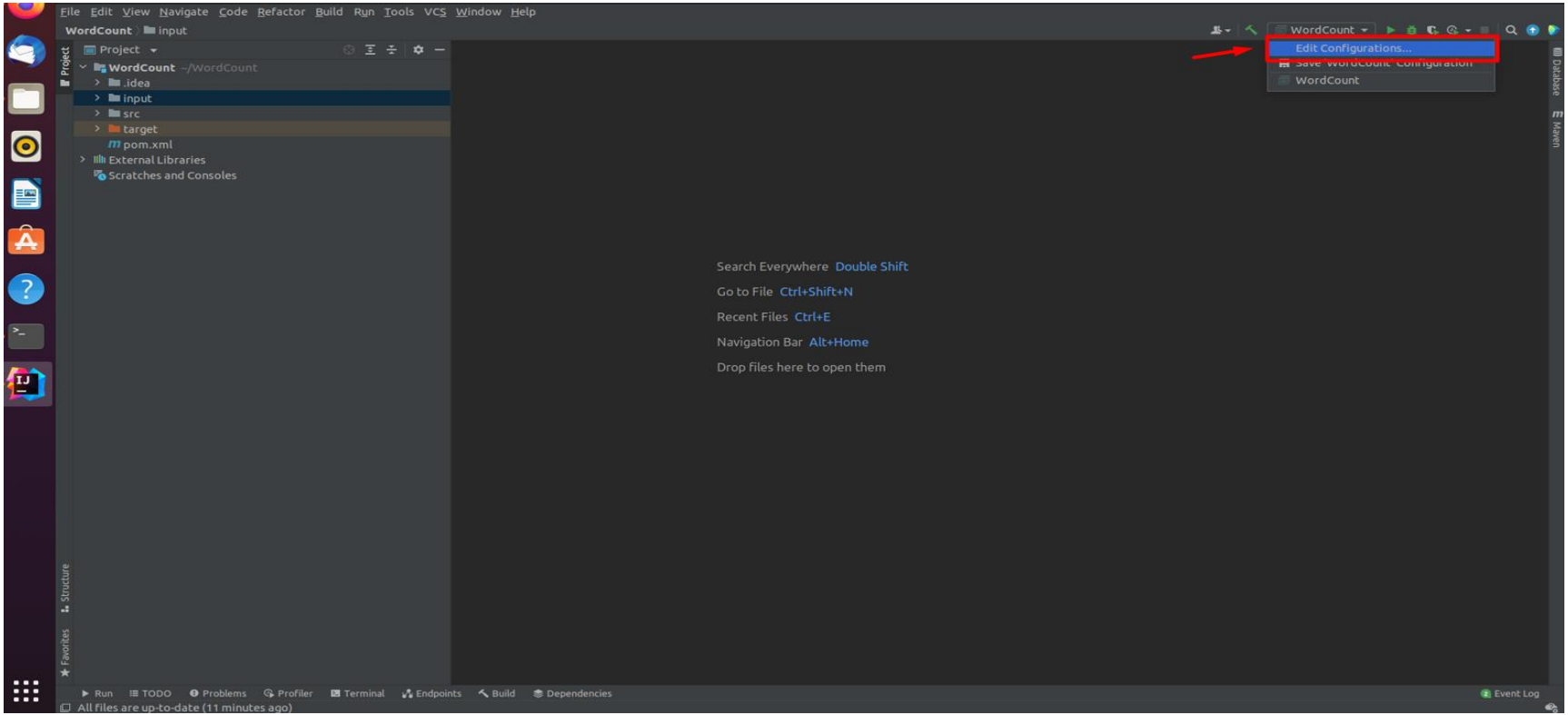
```
<repositories>
    <repository>
      <id>apache</id>
      <url>http://maven.apache.org</url>
    </repository>
  </repositories>

  <dependencies>
    <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-core</artifactId>
      <version>1.2.1</version>
    </dependency>
    <dependency>          <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-common</artifactId>
      <version>3.3.0</version>
    </dependency>
  </dependencies>
```


Ahmed Ramadan, ary00@fayoum.edu.eg

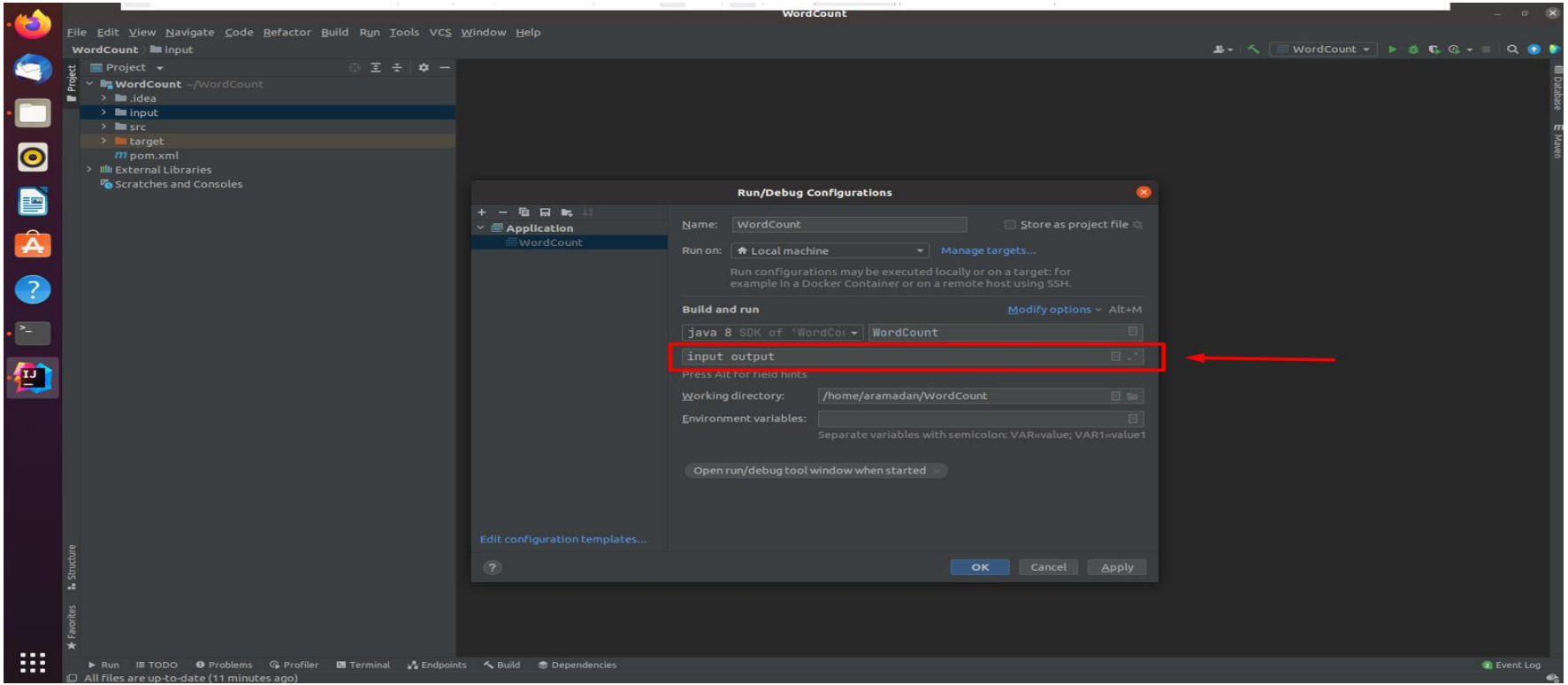
Create Input directory & Input file



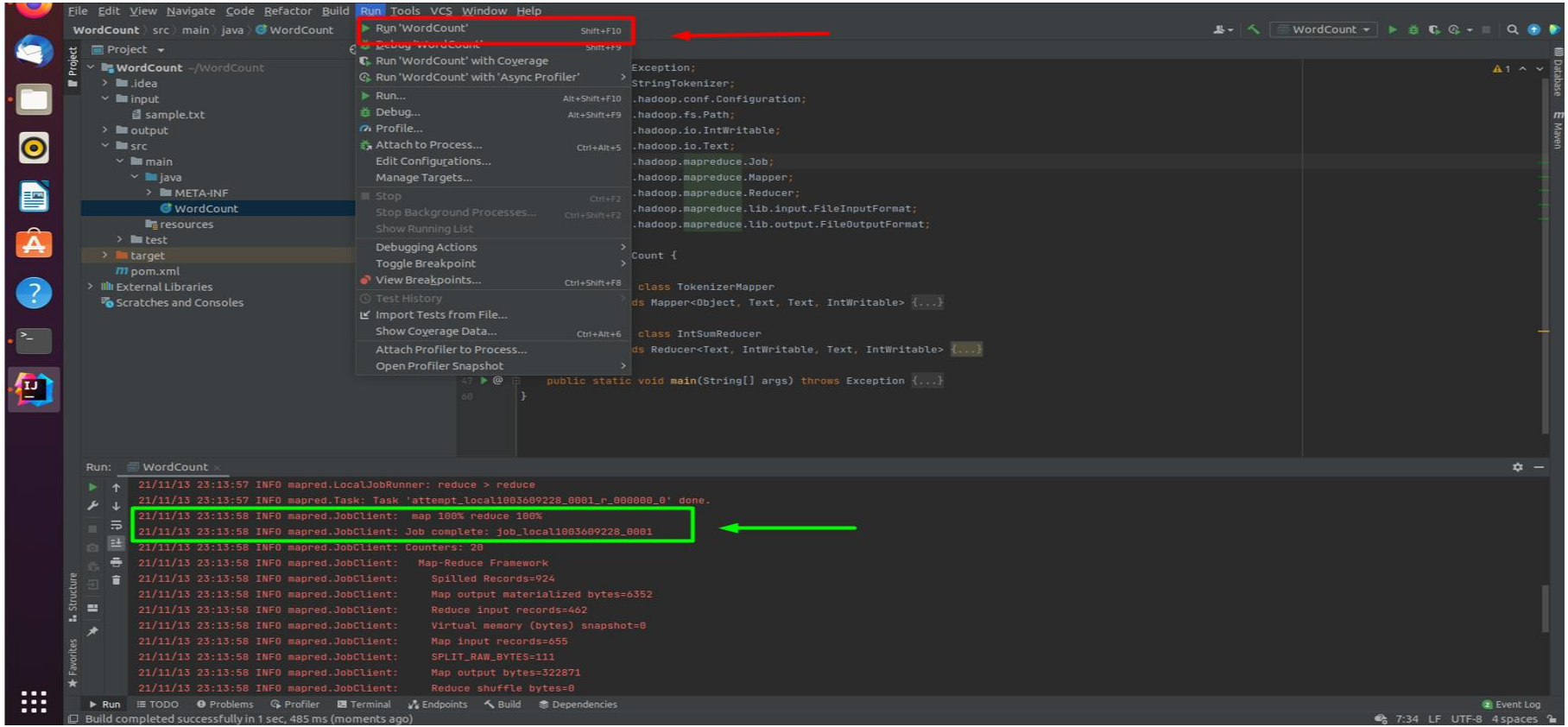
Edit Configuration(To add arguments)



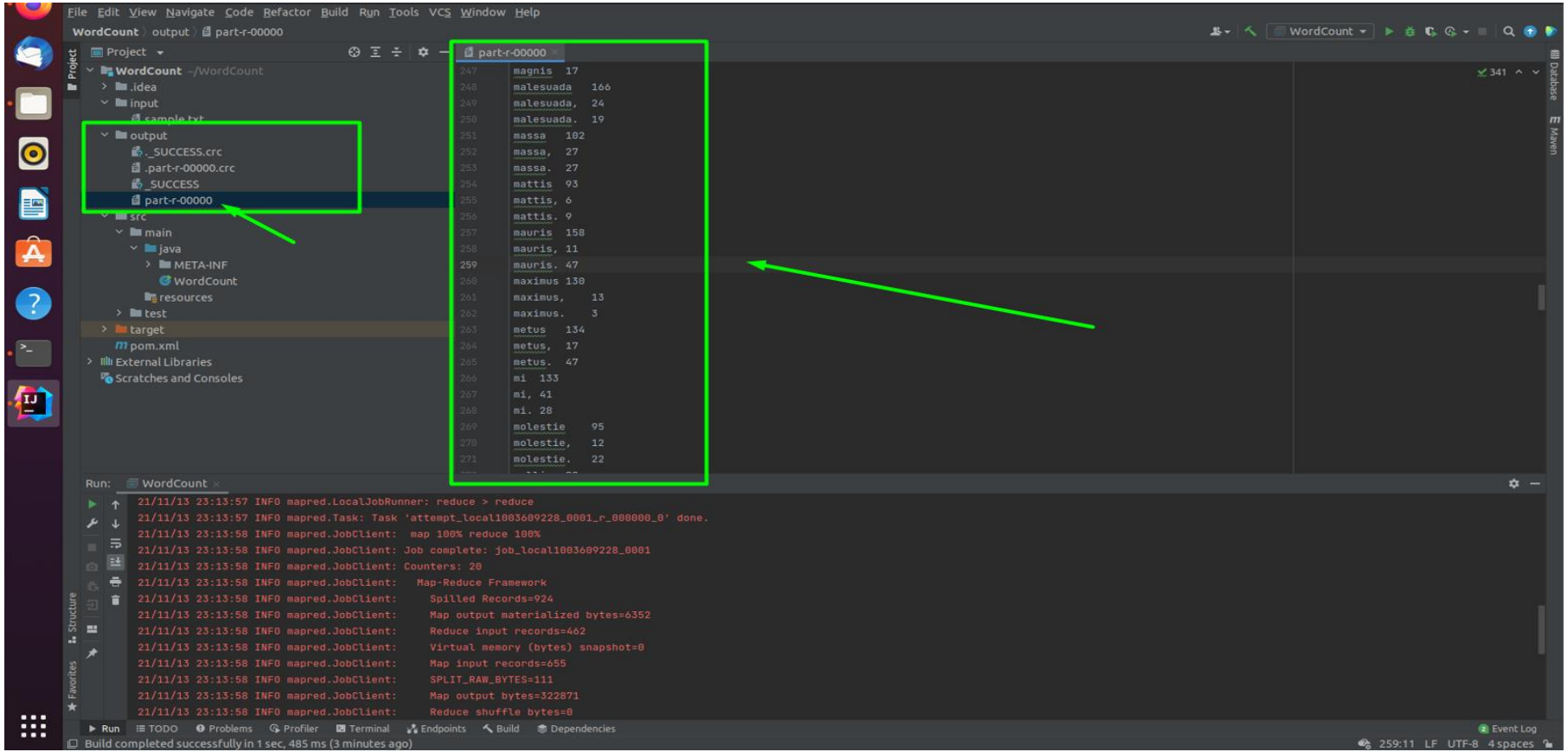
Add input & output dirs in arguments



Run Project => Log tell you the Job Completed with 100%Map – 100% Reduce



Output Directory

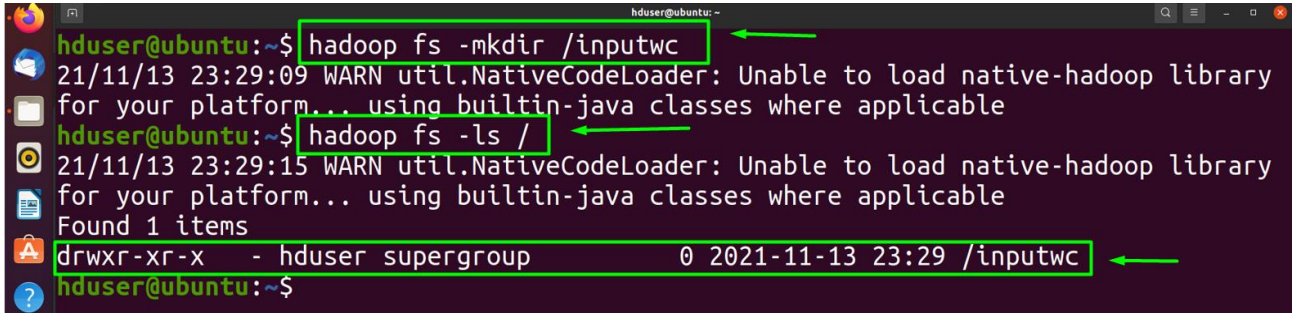


MapReduce on HDFS

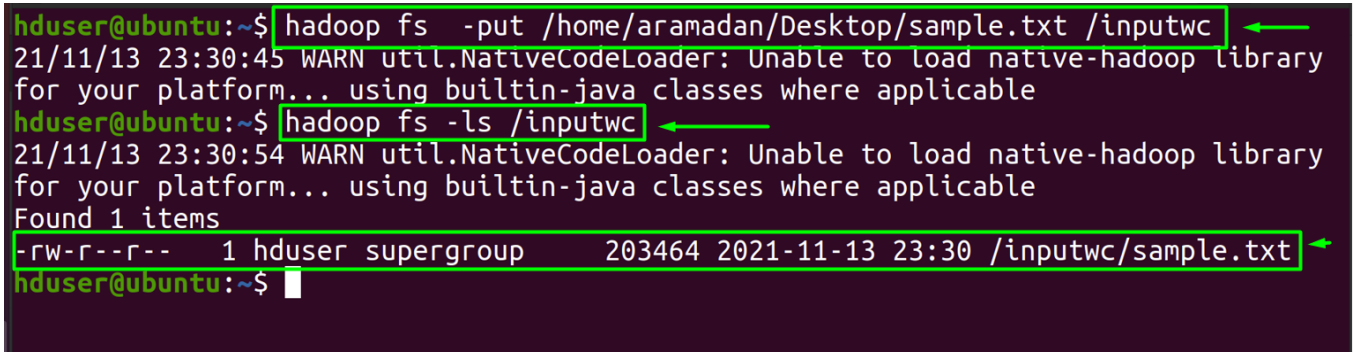
Run MapReduce Job on HDFS

Ahmed Ramadan, ary00@fayoum.edu.eg

Create Directory on hdfs

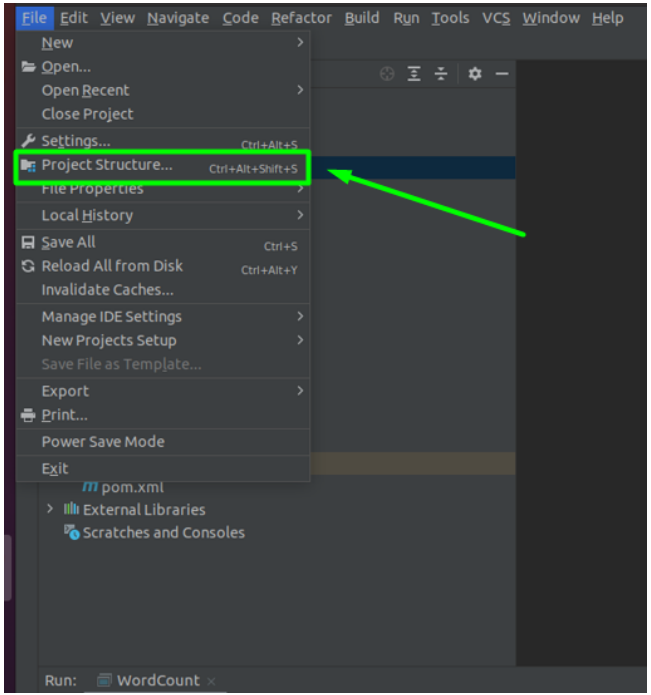


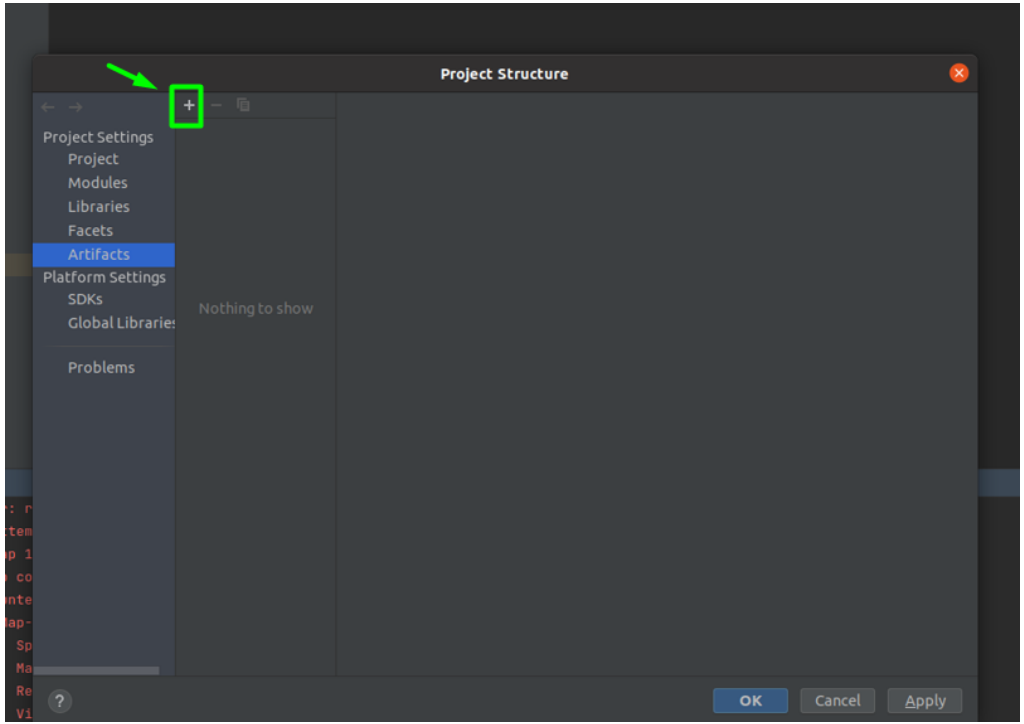
Move File from Linux to hdfs



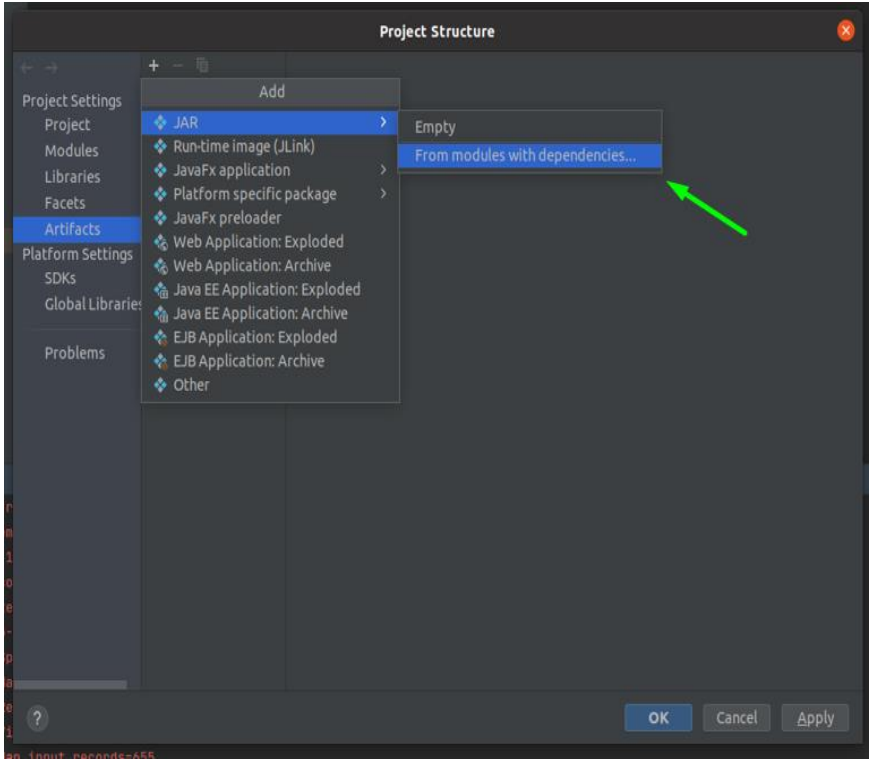
Ahmed Ramadan, ary00@fayoum.edu.eg

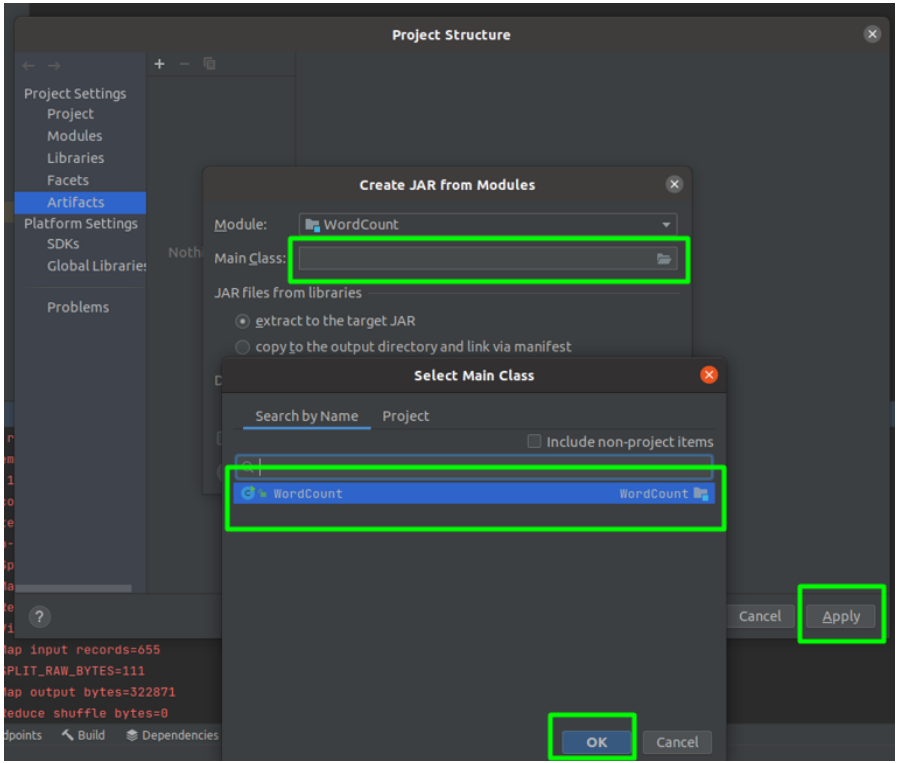
Set IntelliJ to build jar for artifacts





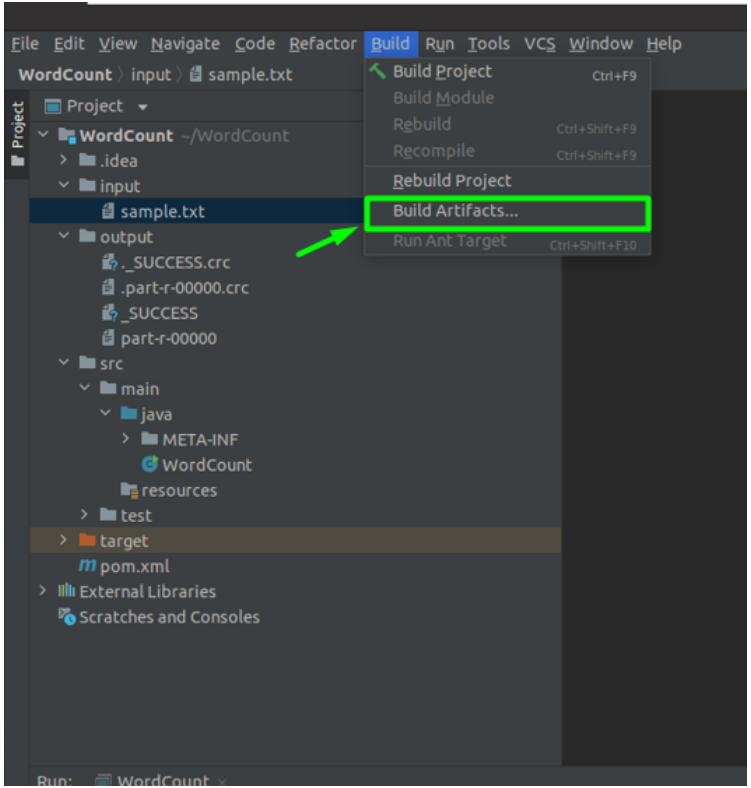
Set IntelliJ to build jar for artifacts

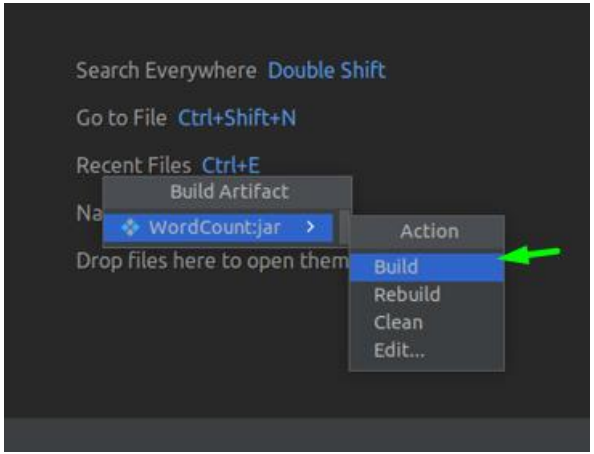


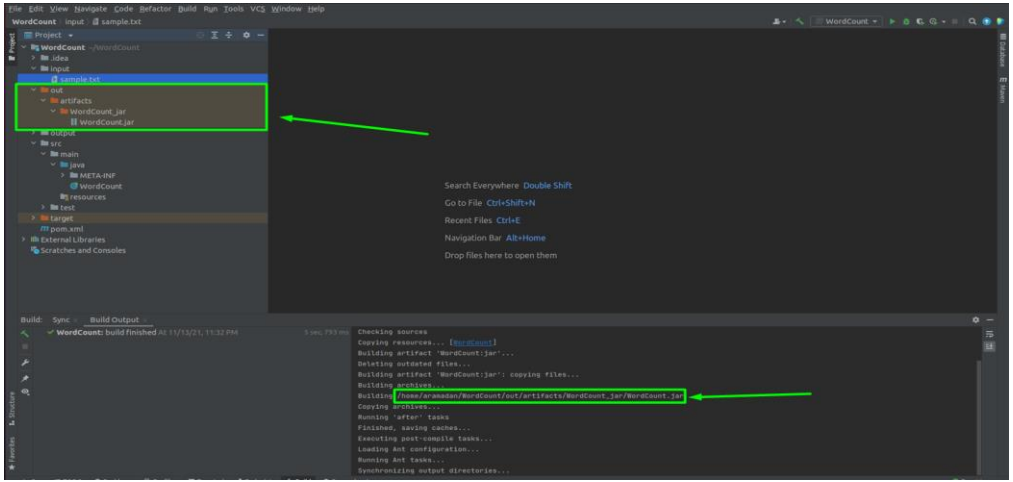


Ahmed Ramadan, ary00@fayoum.edu.eg

Build Arificate & Generate Jar File





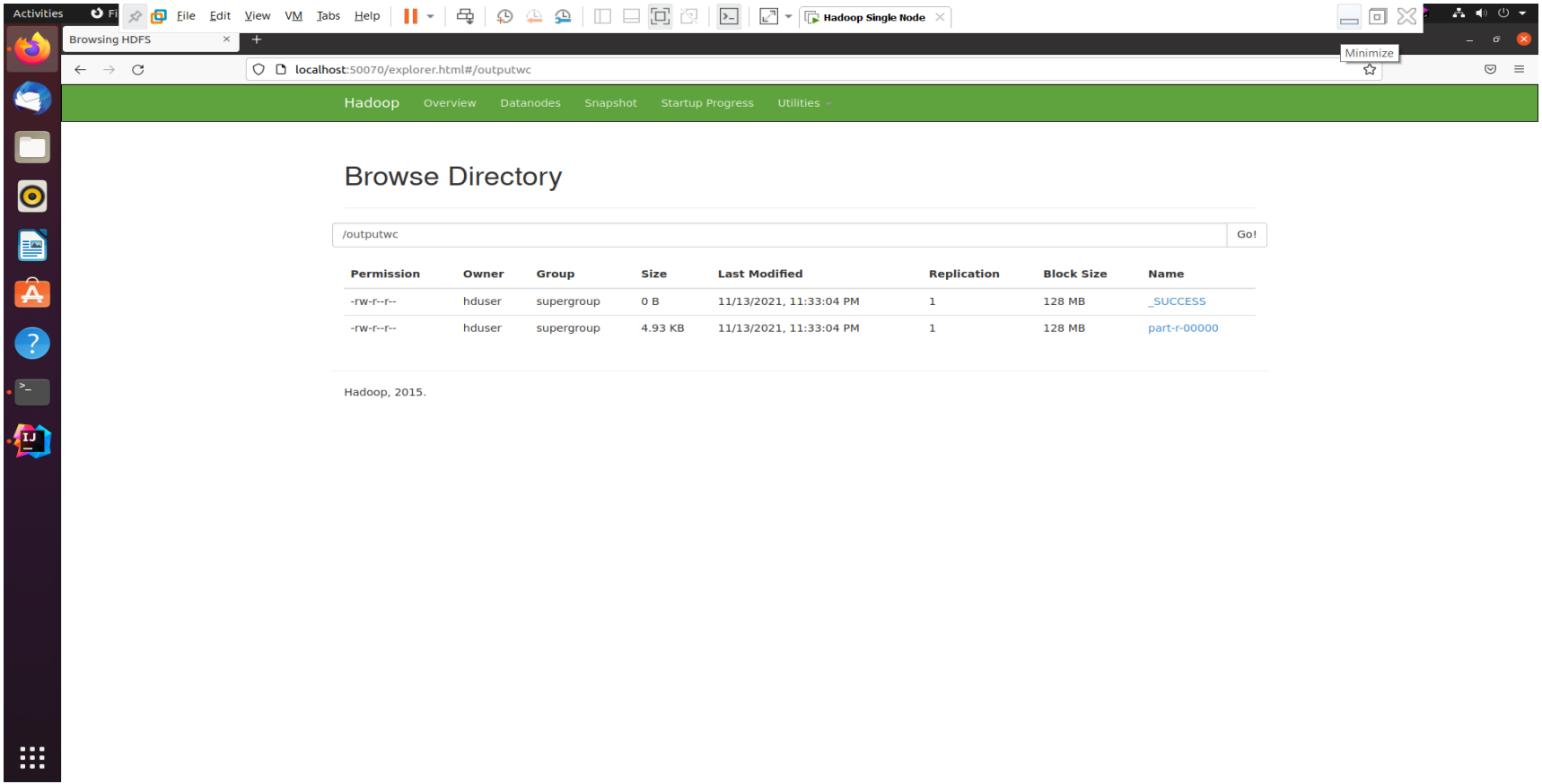


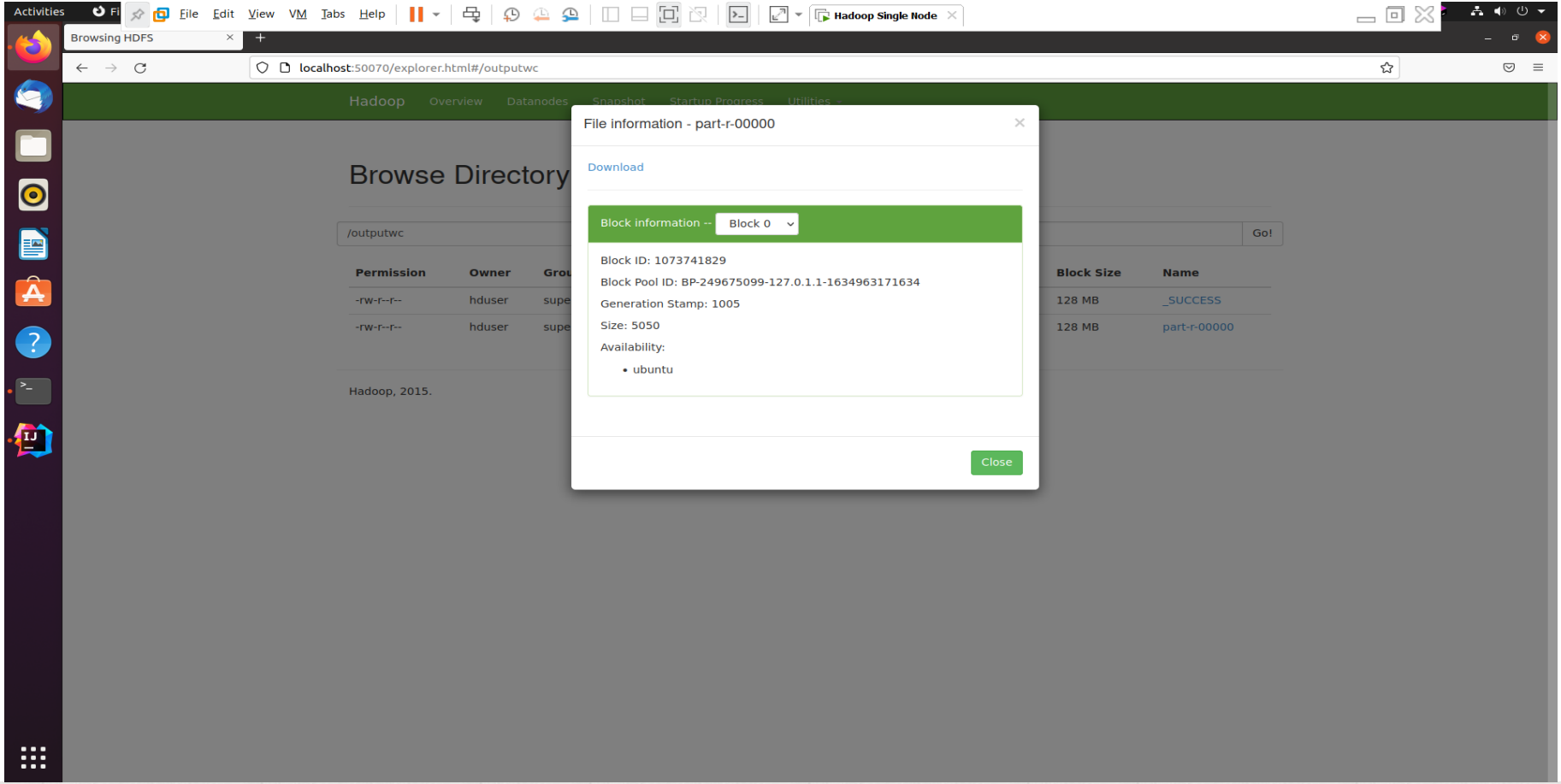
Run Jar on HDFS

```
hduser@ubuntu:~$ hadoop  jar /home/aramadan/WordCount/out/artifacts/WordCount_jar/WordCount.jar /inputwc/sample.txt /outputwc
```

```
cal767999734_0001_r_000000_0' to hdfs://localhost:54310/outputwc/_temporary/0/task_local767999734_0001_r_000000
21/11/13 23:33:04 INFO mapred.LocalJobRunner: reduce > reduce
21/11/13 23:33:04 INFO mapred.Task: Task 'attempt_local767999734_0001_r_000000_0' done.
21/11/13 23:33:04 INFO mapred.LocalJobRunner: Finishing task: attempt_local767999734_0001_r_000000_0
21/11/13 23:33:04 INFO mapred.LocalJobRunner: reduce task executor complete.
21/11/13 23:33:04 INFO mapreduce.Job: Job job_local767999734_0001 running in uber mode : false
21/11/13 23:33:04 INFO mapreduce.Job:  map 100% reduce 100%
21/11/13 23:33:04 INFO mapreduce.Job: Job job_local767999734_0001 completed successfully
21/11/13 23:33:04 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=98449674
                FILE: Number of bytes written=99779896
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=406928
                HDFS: Number of bytes written=5050
```

Results on web portal




Activities
File Edit View VM Tabs Help
Hadoop Single Node
Browsing HDFS
localhost:50070/explorer.html#/outputwc
Hadoop Overview Datanodes Snapshot Startup Progress Utilities
Browse Directory
/outputwc
Go!
Permission Owner Block Size Name
-rw-r--r-- hduser 128 MB _SUCCESS
-rw-r--r-- hduser 128 MB part-r-00000
Hadoop, 2015.
File information - part-r-00000
Download
Block information -- Block 0
Block ID: 1073741829
Block Pool ID: BP-249675099-127.0.1.1-1634963171634
Generation Stamp: 1005
Size: 5050
Availability:
ubuntu
Close

References

- https://reberhardt.com/cs110/summer-2018/lecture-notes/lecture-14/
- https://techvidvan.com/tutorials/how-mapreduce-works/
- https://www.cloudduggu.com/hadoop/architecture/

Ahmed Ramadan, ary00@fayoum.edu.eg

QUESTIONS

Ahmed Ramadan, ary00@fayoum.edu.eg

THANK YOU!

Ahmed Ramadan, ary00@fayoum.edu.eg