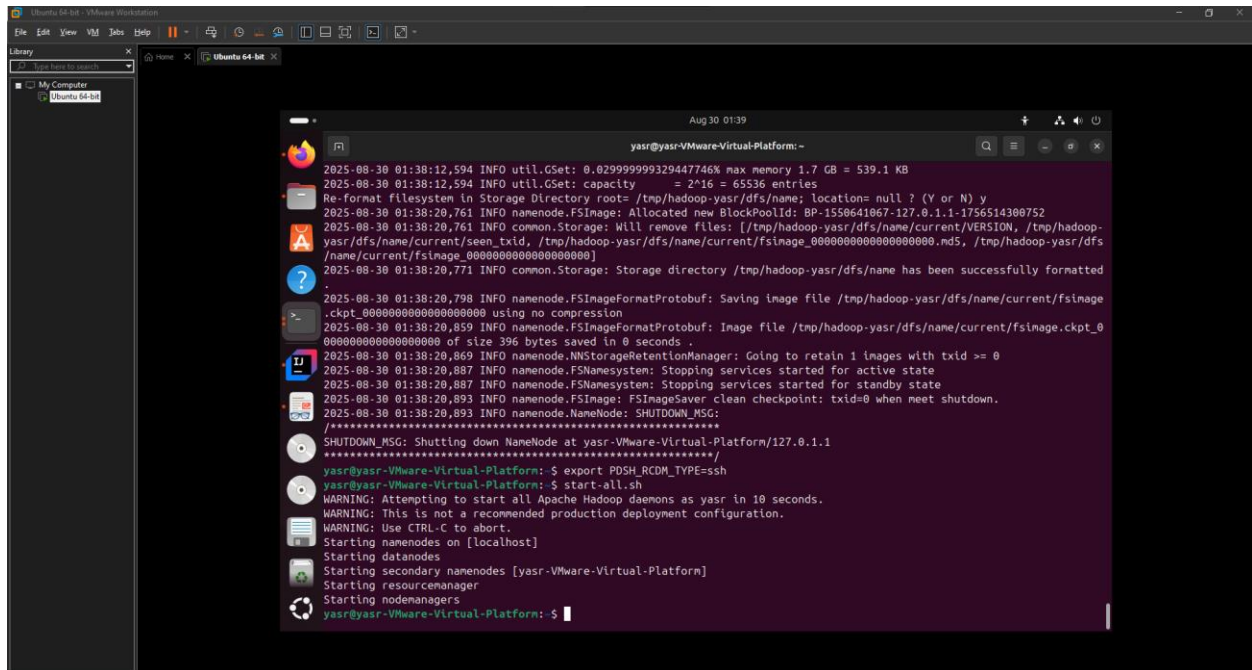


# Rapport du Projet WordCount avec Hadoop

## 1. Objectif du Projet

L'objectif du projet est de développer une application Java utilisant **Hadoop MapReduce** pour compter le nombre d'occurrences de chaque mot dans un fichier texte. Le projet est réalisé avec **Maven** pour la gestion des dépendances et s'exécute en **mode Hadoop mono-nœud**.



```
Aug 30 01:39
yasar@yasar-Virtual-Platform: ~
2025-08-30 01:38:12,594 INFO util.GSet: 0.029999999329447746% max memory 1.7 GB = 539.1 KB
2025-08-30 01:38:12,594 INFO util.GSet: capacity = 2^16 = 65536 entries
Re-format filesystem in Storage Directory root= /tmp/hadoop-yasar/dfs/name; location= null ? (Y or N) y
2025-08-30 01:38:20,761 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1550641067-127.0.1.1-1756514300752
2025-08-30 01:38:20,761 INFO common.Storage: Will remove files: [/tmp/hadoop-yasar/dfs/name/current/VERSION, /tmp/hadoop-yasar/dfs/name/current/seen_txid, /tmp/hadoop-yasar/dfs/name/current/fsimage_000000000000000000.mds, /tmp/hadoop-yasar/dfs/name/current/fsimage_000000000000000000]
2025-08-30 01:38:20,771 INFO common.Storage: Storage directory /tmp/hadoop-yasar/dfs/name has been successfully formatted
2025-08-30 01:38:20,798 INFO namenode.FSImageFormatProtobuf: Saving image file /tmp/hadoop-yasar/dfs/name/current/fsimage.ckpt_000000000000000000 using no compression
2025-08-30 01:38:20,859 INFO namenode.FSImageFormatProtobuf: Image file /tmp/hadoop-yasar/dfs/name/current/fsimage.ckpt_000000000000000000 of size 396 bytes saved in 0 seconds
2025-08-30 01:38:20,869 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2025-08-30 01:38:20,887 INFO namenode.FSNamesystem: Stopping services started for active state
2025-08-30 01:38:20,887 INFO namenode.FSNamesystem: Stopping services started for standby state
2025-08-30 01:38:20,893 INFO namenode.FSImageSaver: clean checkpoint: txid=0 when meet shutdown.
2025-08-30 01:38:20,893 INFO namenode.NameNode: SHUTDOWN_MSG:
//*****
SHUTDOWN_MSG: Shutting down NameNode at yasar-Virtual-Platform/127.0.1.1
*****
yasar@yasar-Virtual-Platform: ~$ export PDSH_RCMD_TYPE=ssh
yasar@yasar-Virtual-Platform: ~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as yasar in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [yasar-Virtual-Platform]
Starting resourcemanager
Starting nodemanagers
yasar@yasar-Virtual-Platform: ~$
```

## 2. Pré-requis

Avant de commencer, les éléments suivants doivent être installés et configurés :

- **Java JDK 8**  
Vérification :
- `java -version`

→ doit afficher 1.8

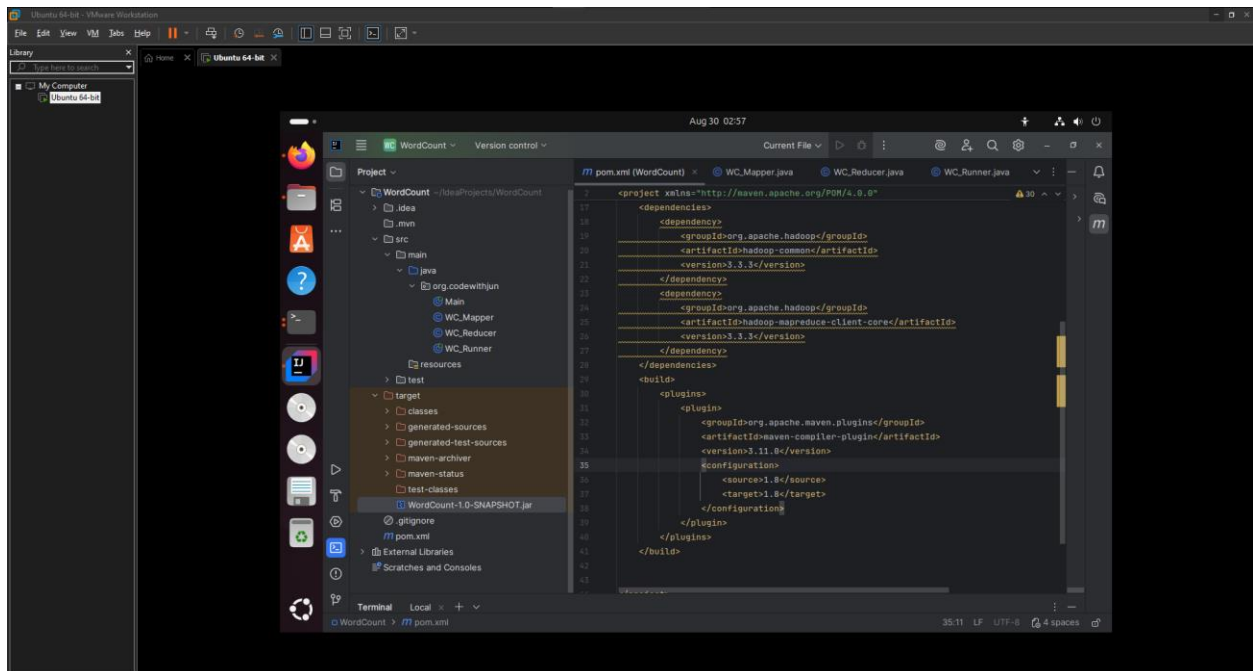
- **Maven**  
Vérification :
- `mvn -v`
- **Hadoop en mono-nœud**  
Installation nécessaire pour pouvoir exécuter la commande `hadoop jar`.
- **IDE recommandé : Eclipse**

### 3. Création du Projet Maven

1. Dans Eclipse :
  - **File → New → Maven Project**
  - Cocher **Create a simple project** → Next
  - Paramètres du projet :
    - **Group Id** : org.codewithjun
    - **Artifact Id** : wordcount
    - **Packaging** : jar
  - Finish

### 4. Gestion des Dépendances Hadoop

Dans le fichier pom.xml, ajouter les dépendances suivantes :



<dependencies>

<dependency>

<groupId>org.apache.hadoop</groupId>

<artifactId>hadoop-common</artifactId>

```
<version>3.3.3</version>
</dependency>
<dependency>
  <groupId>org.apache.hadoop</groupId>
  <artifactId>hadoop-mapreduce-client-core</artifactId>
  <version>3.3.3</version>
</dependency>
</dependencies>
```

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.11.0</version>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
      </configuration>
    </plugin>
  </plugins>
</build>
```

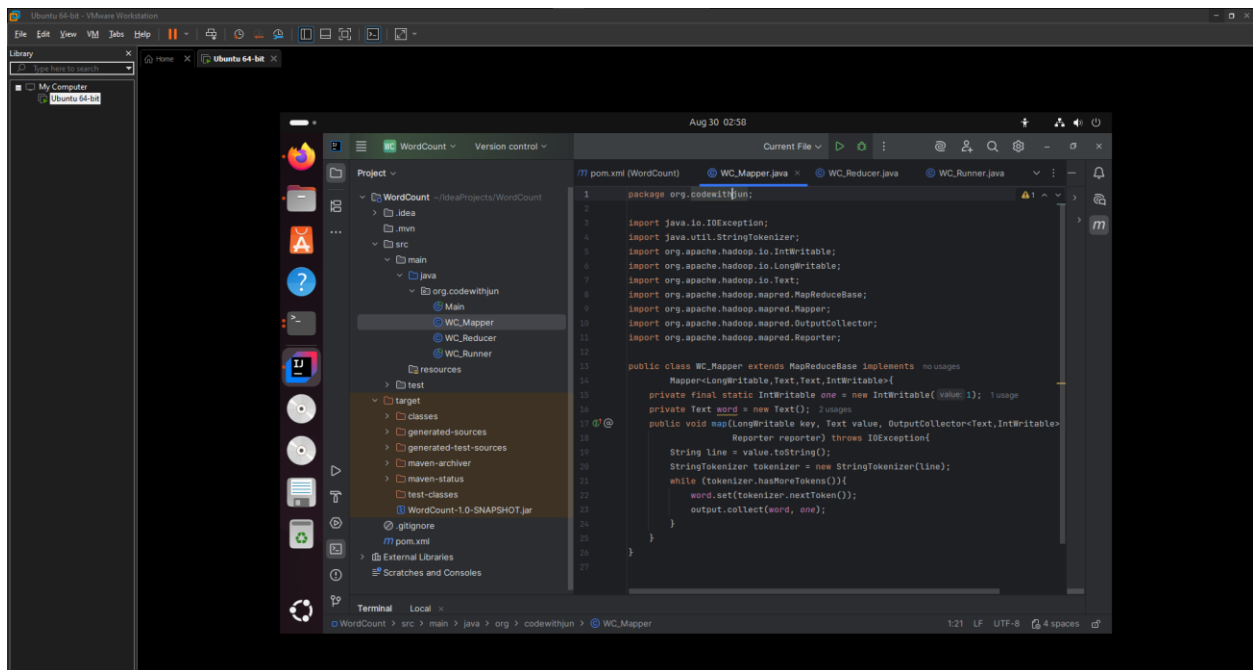
Sauvegarder le fichier pour que Maven télécharge les dépendances.

---

## 5. Création des Classes Java

**Package :** org.codewithjun

### 5.1 WC\_Mapper.java



```
package org.codewithjun;
```

```
import java.io.IOException;
```

```
import java.util.StringTokenizer;
```

```
import org.apache.hadoop.io.IntWritable;
```

```
import org.apache.hadoop.io.LongWritable;
```

```
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapred.MapReduceBase;
```

```
import org.apache.hadoop.mapred.Mapper;
```

```
import org.apache.hadoop.mapred.OutputCollector;
```

```
import org.apache.hadoop.mapred.Reporter;
```

```
public class WC_Mapper extends MapReduceBase implements
```

```
Mapper<LongWritable,Text,Text,IntWritable>{
```

```
private final static IntWritable one = new IntWritable(1);
```

```
private Text word = new Text();
```

```
public void map(LongWritable key, Text value, OutputCollector<Text,IntWritable> output,
```

```

        Reporter reporter) throws IOException{

String line = value.toString();

StringTokenizer tokenizer = new StringTokenizer(line);

while (tokenizer.hasMoreTokens()){

    word.set(tokenizer.nextToken());

    output.collect(word, one);

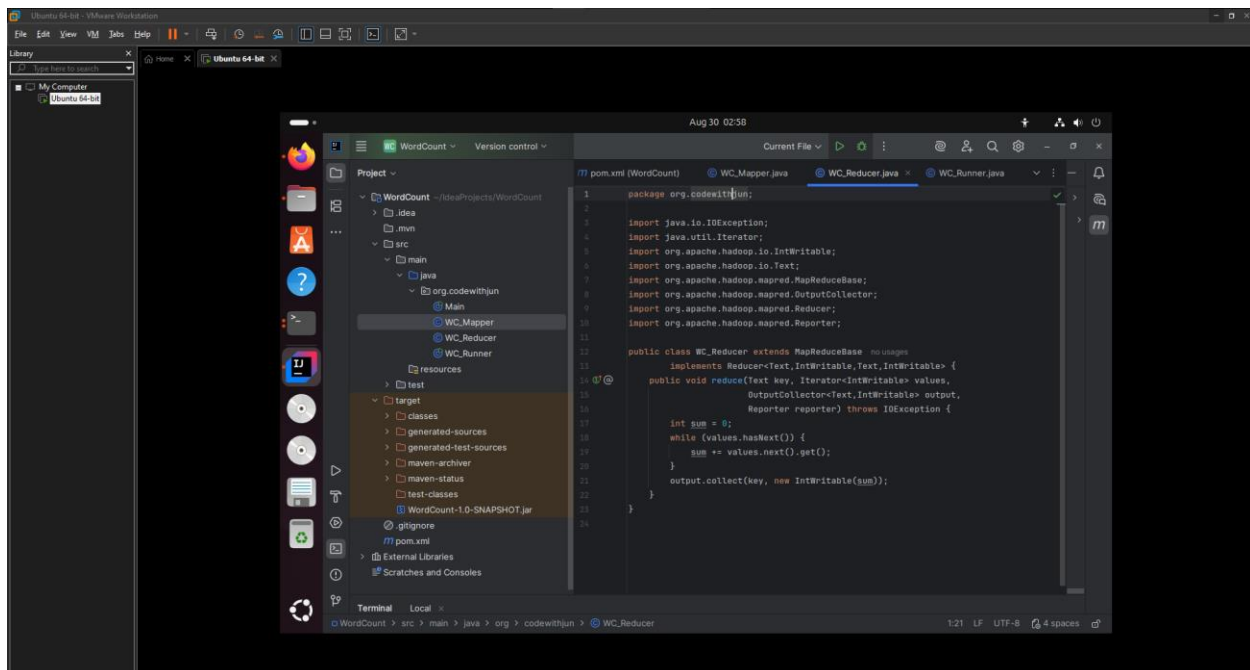
}

}

}

```

## 5.2 WC\_Reducer.java



```
package org.codewithjun;
```

```
import java.io.IOException;
```

```
import java.util.Iterator;
```

```
import org.apache.hadoop.io.IntWritable;
```

```
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapred.MapReduceBase;
```

```

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reducer;

import org.apache.hadoop.mapred.Reporter;

public class WC_Reducer extends MapReduceBase
implements Reducer<Text,IntWritable,Text,IntWritable> {

    public void reduce(Text key, Iterator<IntWritable> values,
        OutputCollector<Text,IntWritable> output,
        Reporter reporter) throws IOException {

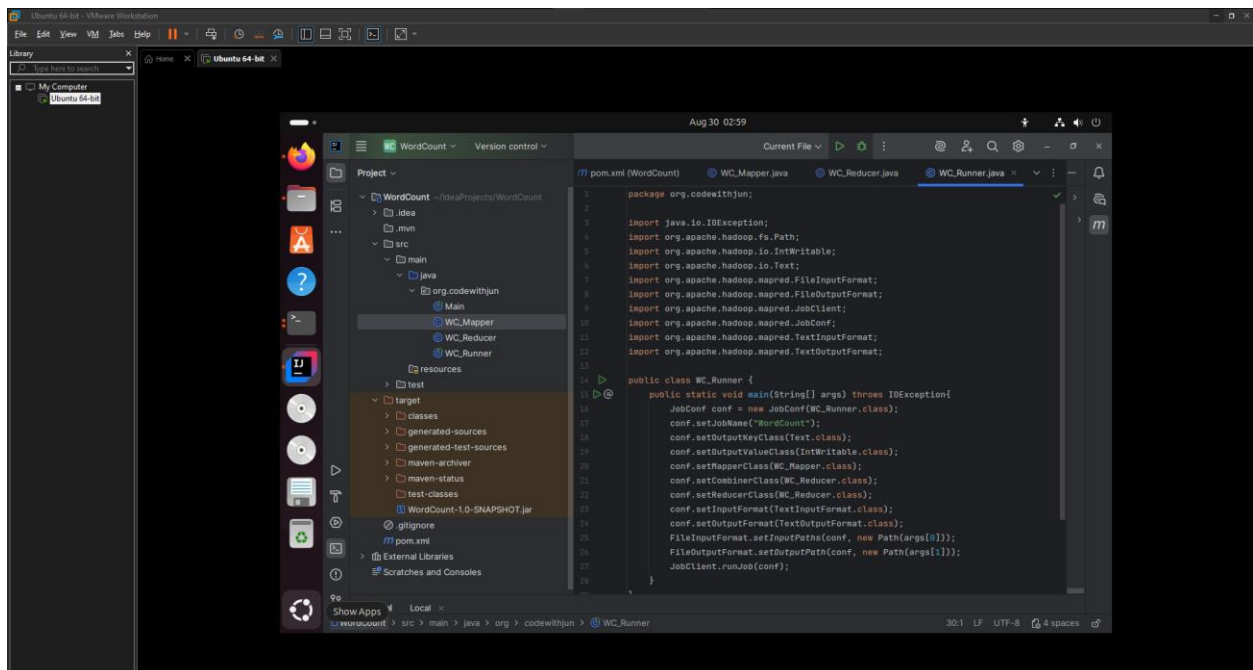
        int sum = 0;

        while (values.hasNext()) {
            sum += values.next().get();
        }

        output.collect(key, new IntWritable(sum));
    }
}

```

### 5.3 WC\_Runner.java



```
package org.codewithjun;
```

```
import java.io.IOException;
```

```
import org.apache.hadoop.fs.Path;
```

```
import org.apache.hadoop.io.IntWritable;
```

```
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapred.FileInputFormat;
```

```
import org.apache.hadoop.mapred.FileOutputFormat;
```

```
import org.apache.hadoop.mapred.JobClient;
```

```
import org.apache.hadoop.mapred.JobConf;
```

```
import org.apache.hadoop.mapred.TextInputFormat;
```

```
import org.apache.hadoop.mapred.TextOutputFormat;
```

```
public class WC_Runner {
```

```
    public static void main(String[] args) throws IOException {
```

```
        JobConf conf = new JobConf(WC_Runner.class);
```

```
        conf.setJobName("WordCount");
```

```
        conf.setOutputKeyClass(Text.class);
```

```
        conf.setOutputValueClass(IntWritable.class);
```

```
        conf.setMapperClass(WC_Mapper.class);
```

```
        conf.setCombinerClass(WC_Reducer.class);
```

```
        conf.setReducerClass(WC_Reducer.class);
```

```
        conf.setInputFormat(TextInputFormat.class);
```

```
        conf.setOutputFormat(TextOutputFormat.class);
```

```
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
```

```
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
```

```
        JobClient.runJob(conf);
```

```
    }
```

```
}
```

---

## 6. Préparation du Fichier d'Entrée

```
yasr@yasr-VMware-Virtual-Platform:~/Desktop$ hadoop fs -mkdir /input
yasr@yasr-VMware-Virtual-Platform:~/Desktop$ hadoop fs -put input.txt /input
```

- Créer un dossier input dans le projet
- Ajouter un fichier input.txt contenant quelques lignes de texte

---

## 7. Compilation et Packaging

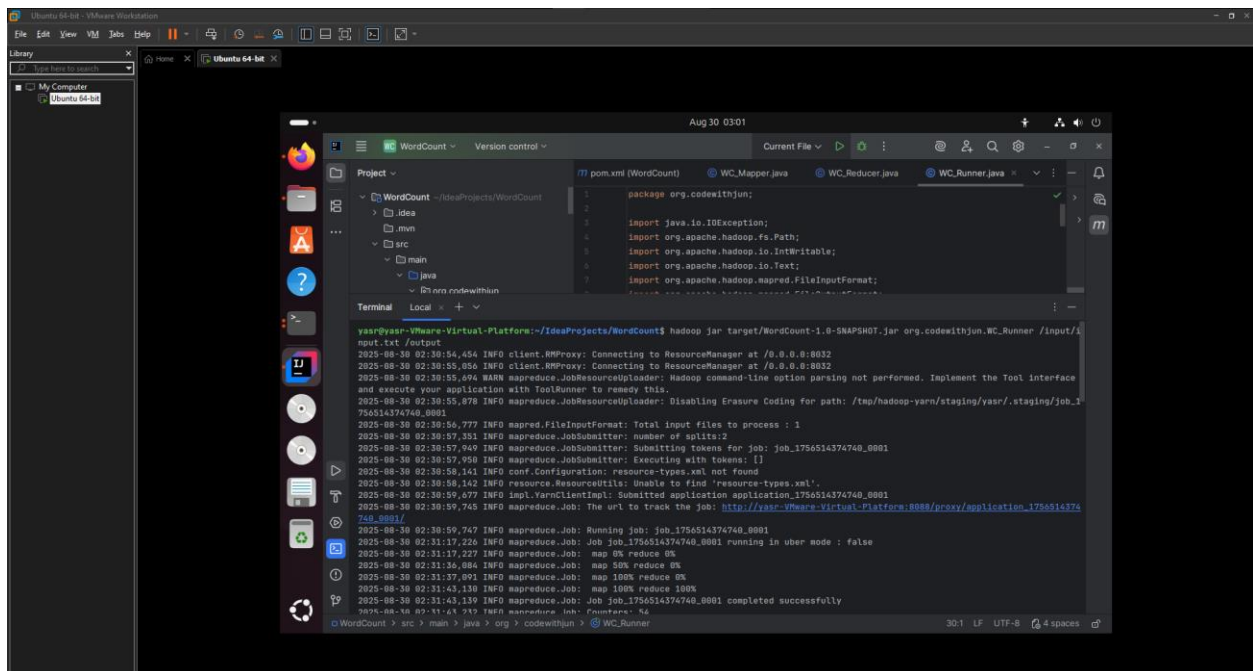
- Dans Eclipse : **Run As** → **Maven build...** → **Goals : clean package** → **Run**
- Ou en terminal : `mvn clean package`

Un fichier JAR sera généré dans le dossier target.

---

## 8. Exécution du Job WordCount avec Hadoop

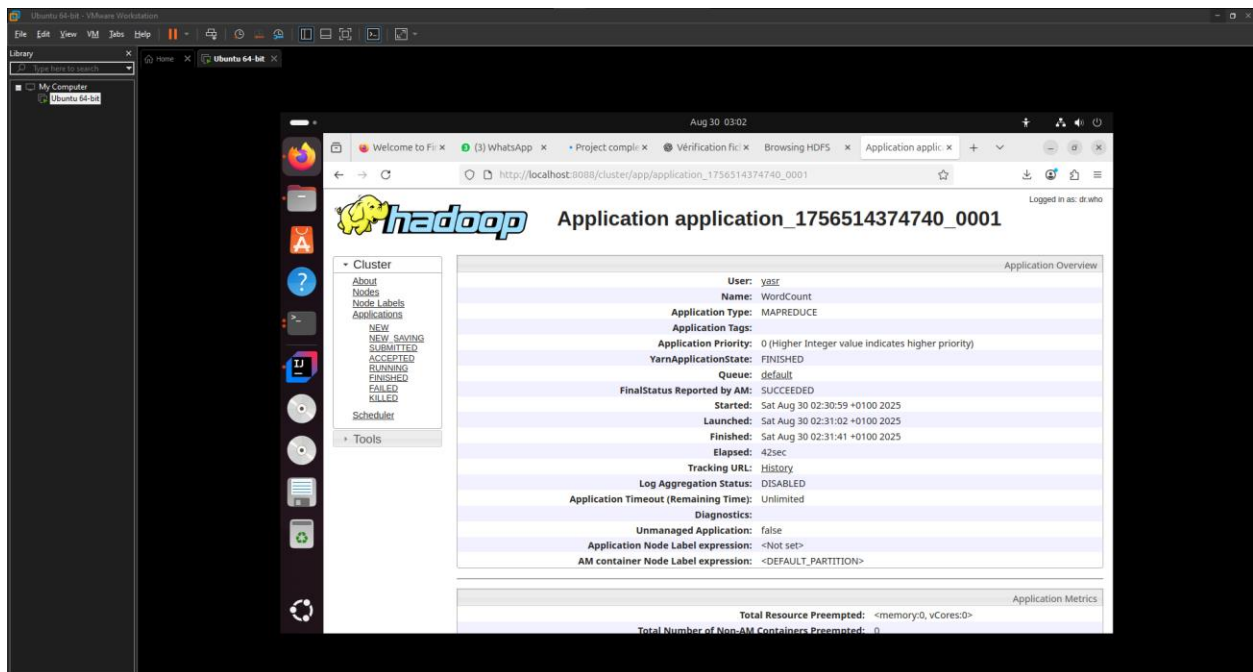
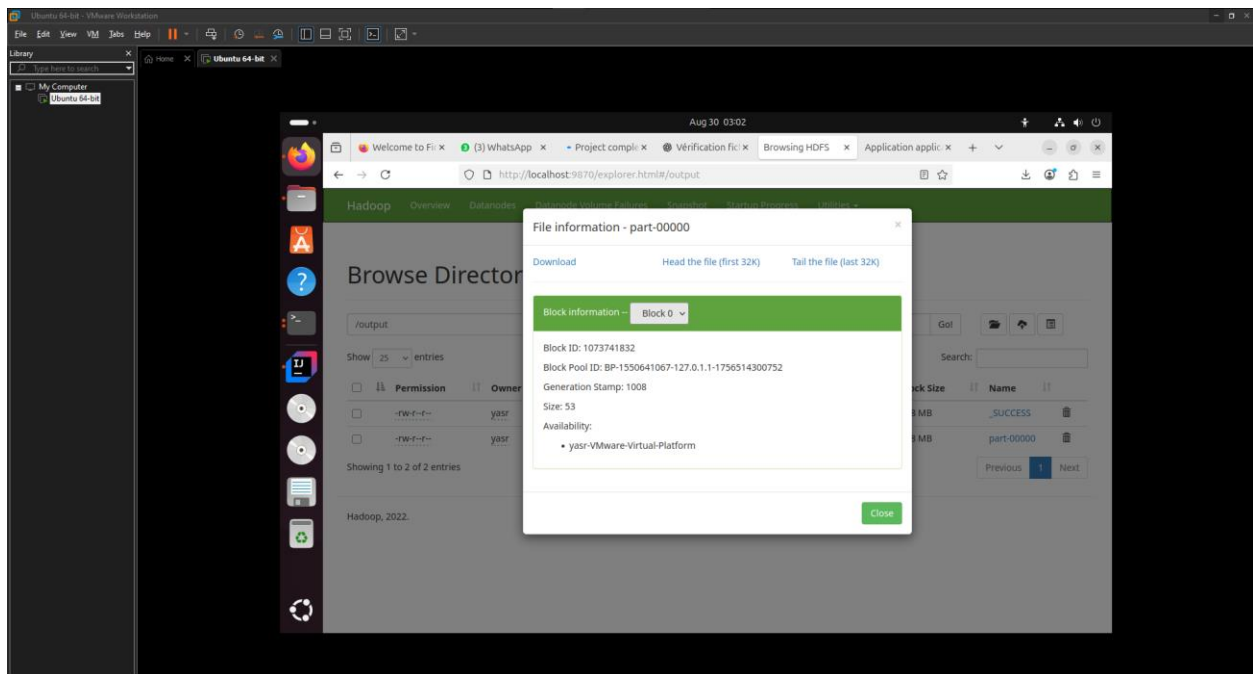
1. Lancer le job MapReduce :



```
yasr@yasr-VMware-Virtual-Platform:~/IdeaProjects/WordCount$ hadoop jar target/WordCount-1.0-SNAPSHOT.jar org.codewithjun.WC_Runner /input/input.txt /output
2025-08-30 02:30:54,454 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2025-08-30 02:30:55,056 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2025-08-30 02:30:55,696 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2025-08-30 02:30:55,878 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/yasr/.staging/job_1756514374740_0001
2025-08-30 02:30:56,777 INFO mapred.FileInputFormat: Total input files to process : 1
2025-08-30 02:30:57,351 INFO mapreduce.JobSubmitter: number of splits:2
2025-08-30 02:30:57,940 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1756514374740_0001
2025-08-30 02:30:57,950 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-08-30 02:30:58,141 INFO conf.Configuration: resource-types.xml not found
2025-08-30 02:30:58,142 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-08-30 02:30:59,677 INFO impl.YarnClientImpl: Submitted application application_1756514374740_0001
2025-08-30 02:30:59,745 INFO mapreduce.Job: The url to track the job: http://yasr-VMware-Virtual-Platform:8088/proxy/application_1756514374740_0001/
2025-08-30 02:30:59,747 INFO mapreduce.Job: Running job: job_1756514374740_0001
2025-08-30 02:31:17,226 INFO mapreduce.Job: Job job_1756514374740_0001 running in user mode : false
2025-08-30 02:31:17,227 INFO mapreduce.Job: map 0% reduce 0%
2025-08-30 02:31:36,084 INFO mapreduce.Job: map 50% reduce 0%
2025-08-30 02:31:37,091 INFO mapreduce.Job: map 100% reduce 0%
2025-08-30 02:31:43,130 INFO mapreduce.Job: map 100% reduce 100%
2025-08-30 02:31:43,139 INFO mapreduce.Job: Job job_1756514374740_0001 completed successfully
2025-08-30 02:31:43,141 INFO mapreduce.Job: Counters: 54
```

```
hadoop jar target/WordCount-1.0-SNAPSHOT.jar org.codewithjun.WC_Runner /input/input.txt /output
```





Realiser Par :

Yassir El ghriissi