Assignment 11

Make sure you have set up Hadoop on your Linux machine.

You can check if Hadoop is installed properly by using the following command in the terminal:

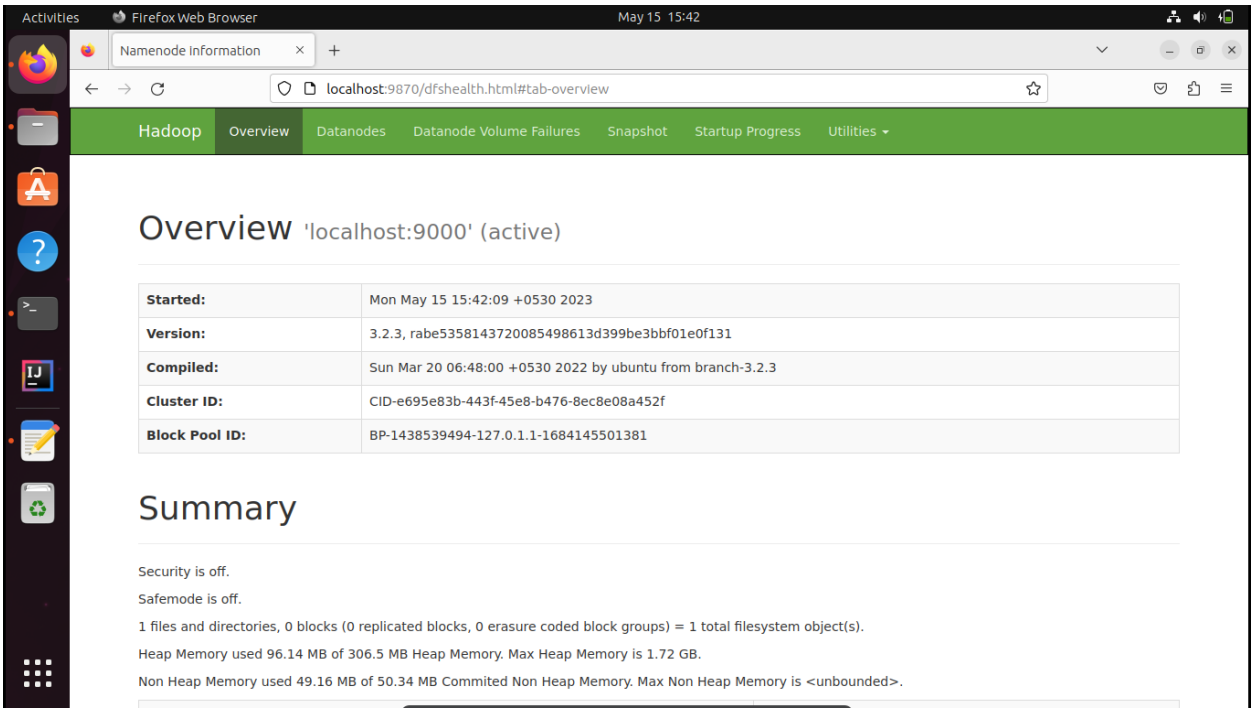
hadoop version

Start the Hadoop server using the following command:

start-all.sh

After the server boots up you can open any browser and enter the following URL to view the Hadoop user interface.

URL: localhost:/9870



Open Ubuntu Software Store and install IntelliJ Idea Community Edition

Open IntelliJ Idea Community Edition and click on 'Create New Project.'

Enter the name of the project as 'WordCountExample'

Select 'Maven' as the build system while creating new project.

Click on 'Advanced Settings' and keep note of your 'GroupId' which will be required later.

Click on Create then the maven project will be created.

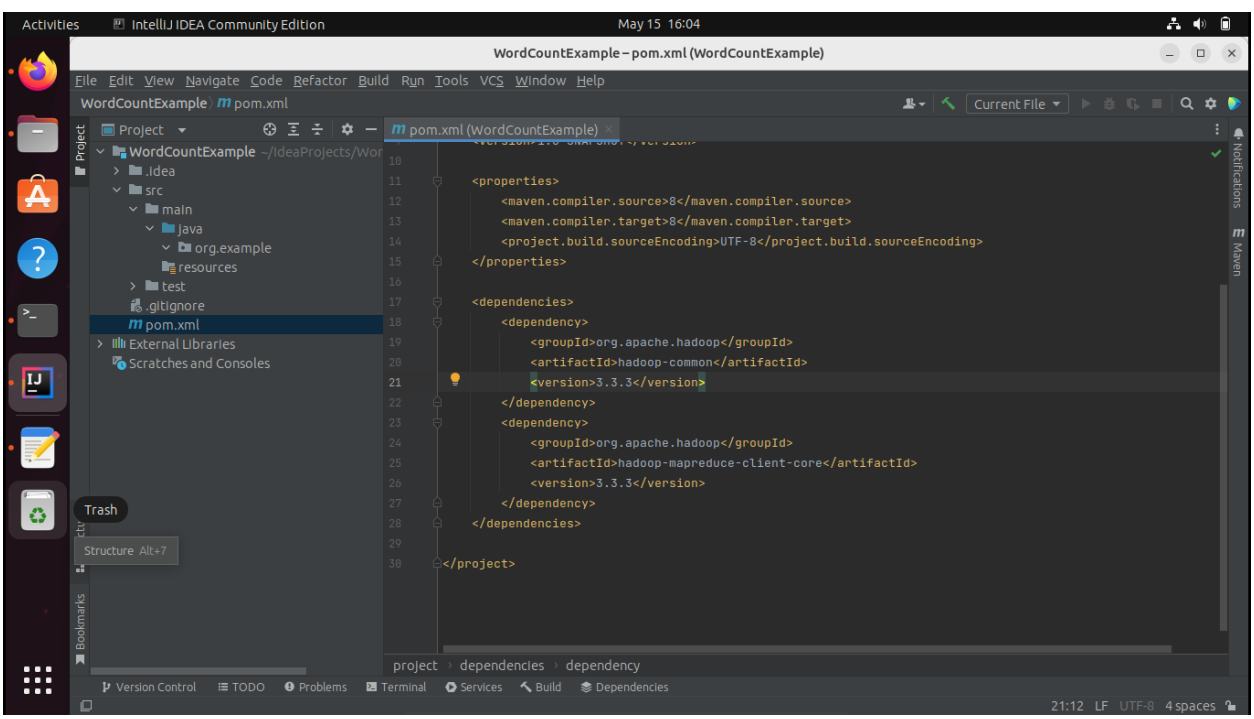
On the left hand side of the file explorer, Delete Main.java which is present inside

src/java/org.example/Main.java

Open pom.xml

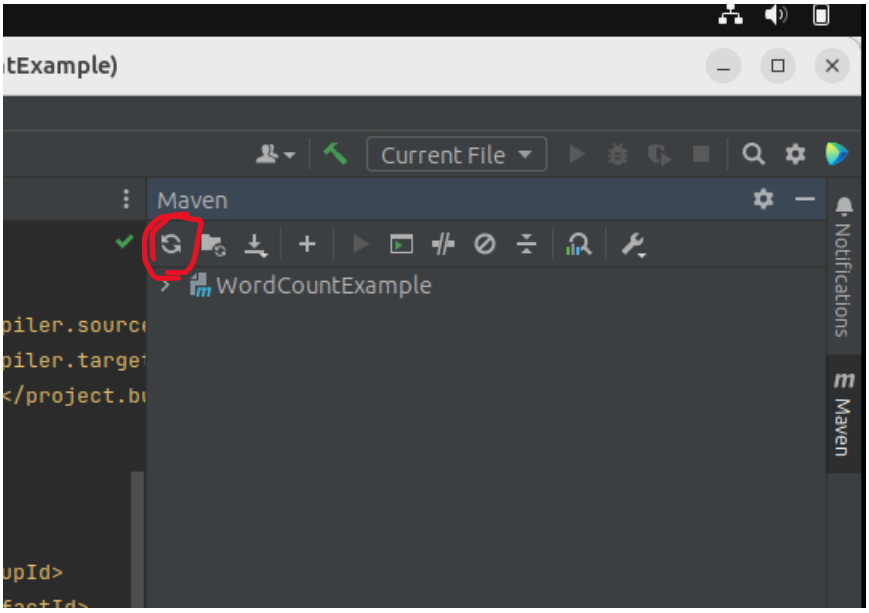
Add the following dependencies:

```
<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>
```



Click on Maven tab on the right hand side of the IDE.

Click on ‘reload’ icon to download all the dependencies.



Right click on your package name and select New > Java Class

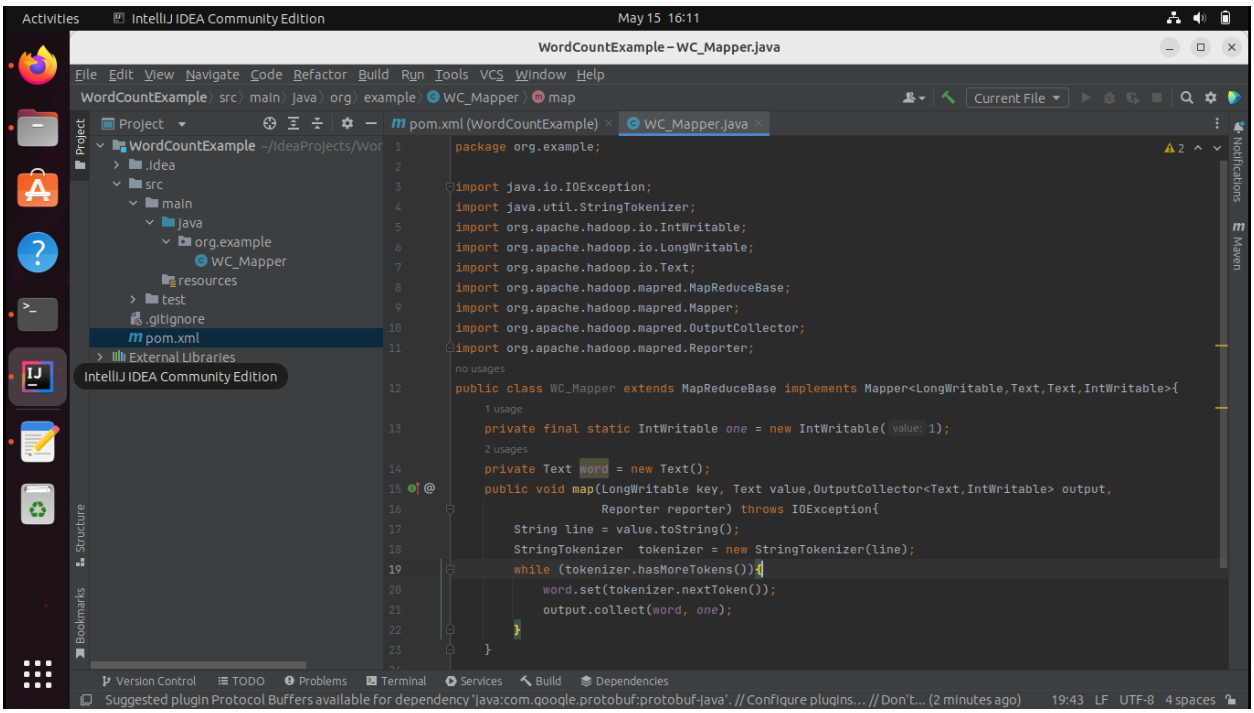
Enter the class name as ‘WC_Mapper’ and hit enter.

Add the code given below to the WC_Mapper class

****

```
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);
    }
  }
}
```



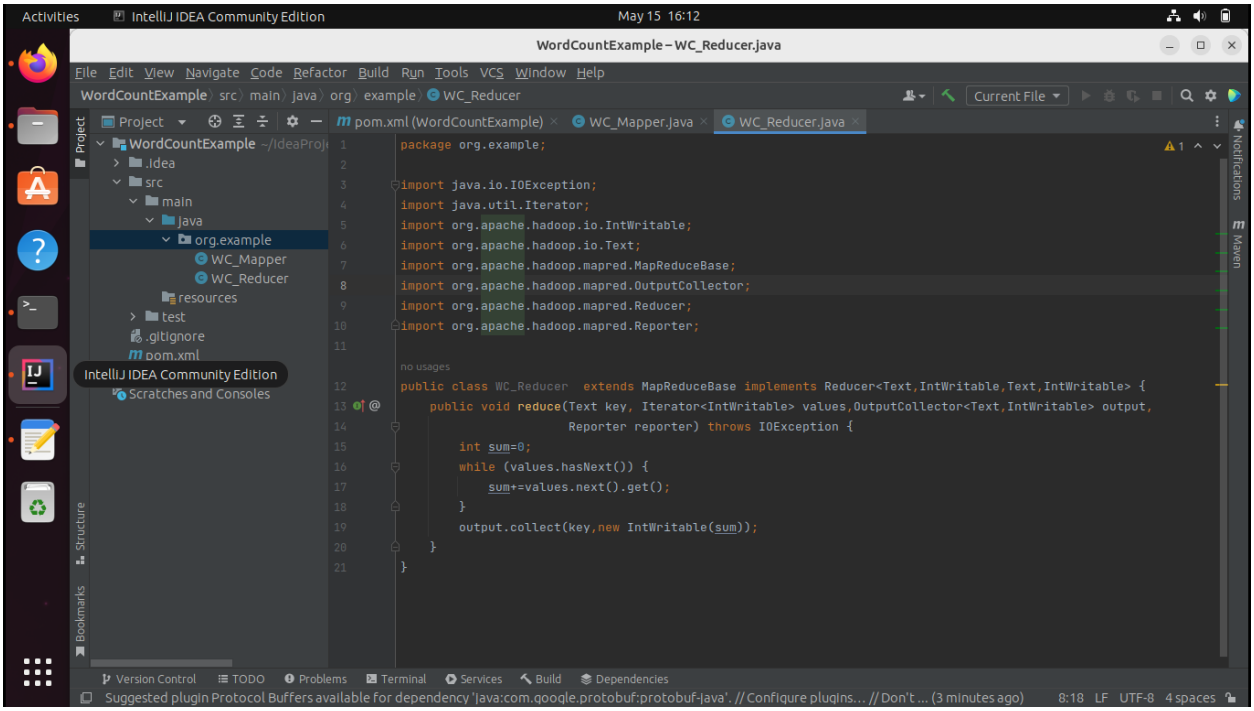
****

Create new class WC_Reducer and add the following code inside it:

****

```
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));

   }

}
```



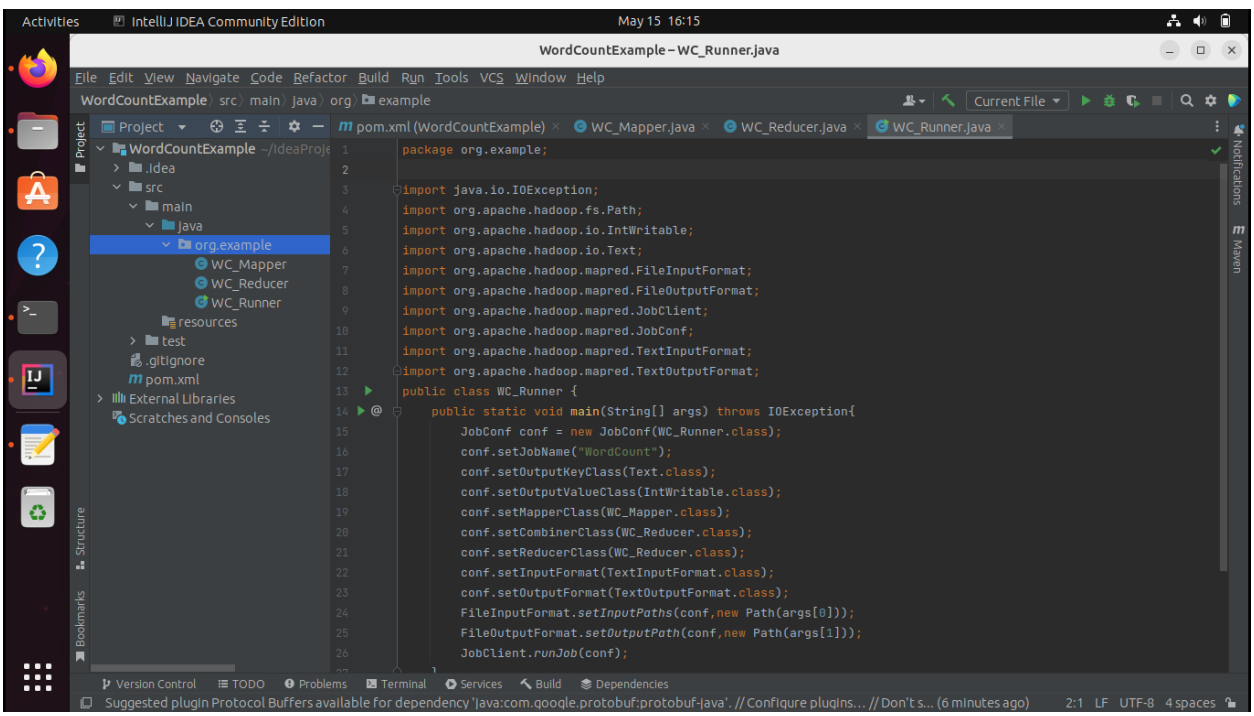
****

Create new class 'WC_Runner' and add the following code:

****

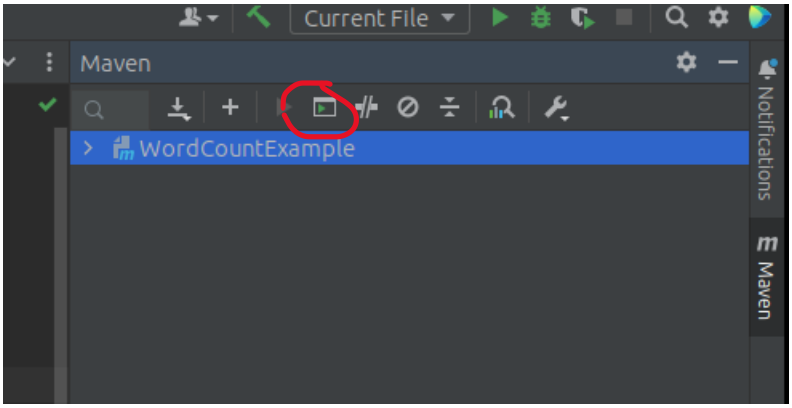
```
import java.io.IOException;

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);
    }
}
```



****

Now, Click on maven tab and select ‘Execute Maven Goal’



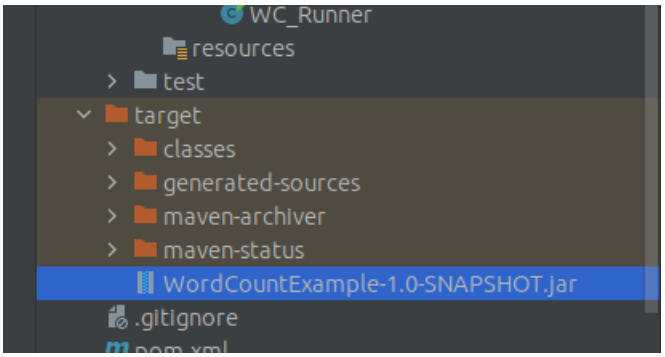
Run

mvn clean

Again click on ‘Execute Maven Goal’

mvn install

This will create a .jar file which we will use later.

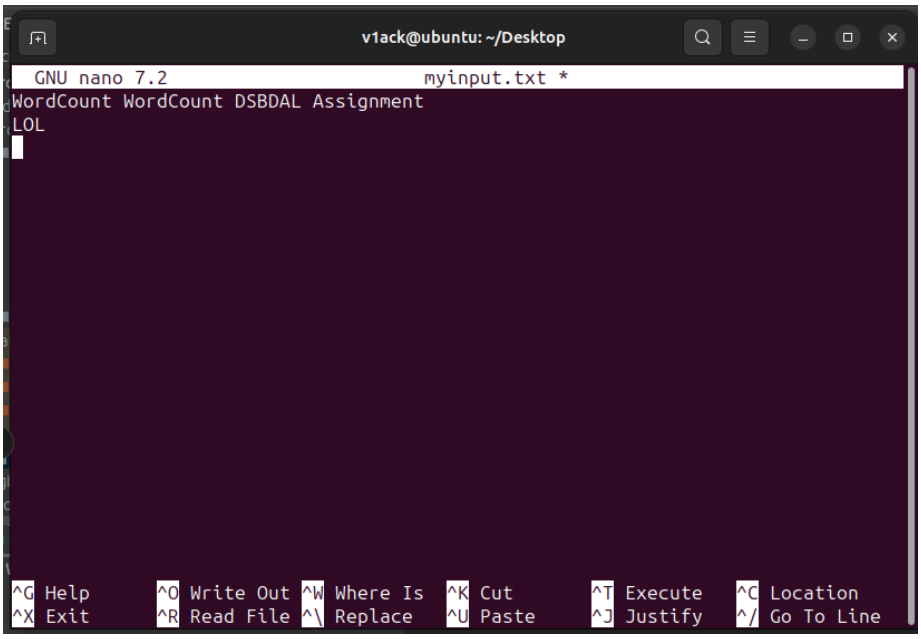


Open your terminal, move to your Desktop.

cd Desktop

nano myinput.txt

Add anything that you want.



After adding the words or sentences, type

Ctrl + O

Enter

Ctrl + X

Enter

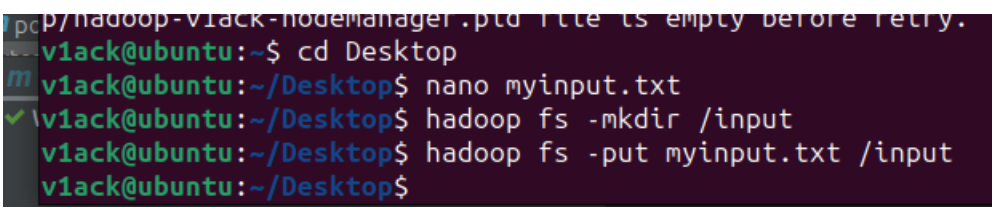
Above commands will then save the text that you wrote into myinput.txt file.

Run the below command to create a new directory inside Hadoop file system.

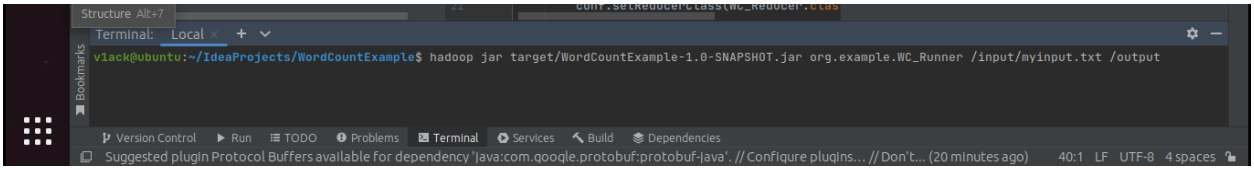
hadoop fs -mkdir /input

Run the below command to move the myinput.txt to your newly crated input folder inside hadoop file system.

hadoop fs -put myinput.txt /input



Now, open IntelliJ IDE and select terminal from the bottom bar and run the following command.



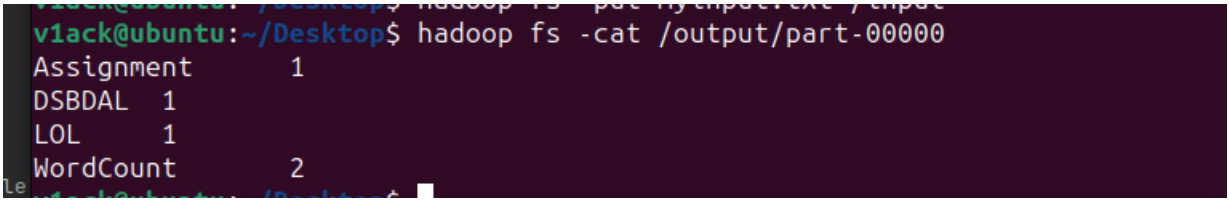
hadoop jar target/WordCountExample-1.0-SNAPSHOT.jar org.example.WC_Runner /input/myinput.txt /output

This will take 2-3 minutes depending on your processing capabilities.

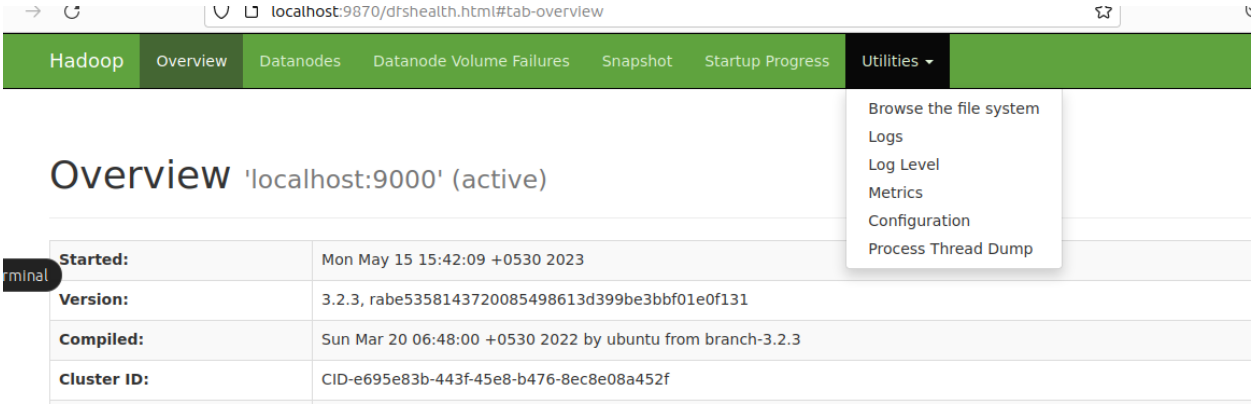
After successfully executing your output will be ready.

To check the output, run the following command in your terminal:

hadoop fs -cat /output/part-00000



You can also check the output from the Hadoop user interface by going into the Utilities/Browse the file system



Type '/' in the Browse Directory

Select output

Click on part-00000

Click on ‘Head the file (first 32K)