## **DS&BDA:-12**

## Steps:-

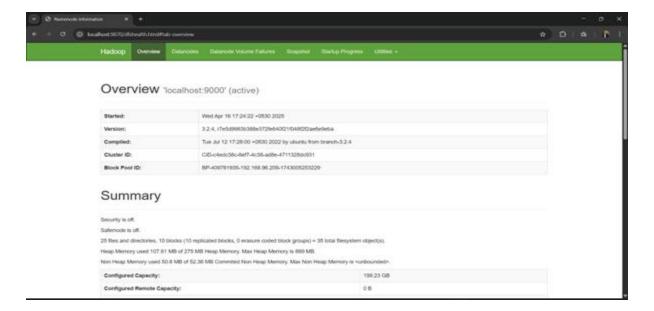
- Check version
  - > java -version
  - > hadoop version
- **4** Make Log Folder
  - > InputLog
    - 10.223.157.186 - [15/Jul/2009:20:50:32 -0700] "GET /assets/js/the-associates.js HTTP/1.1" 304 -
    - 10.223.157.186 - [15/Jul/2009:20:50:33 -0700] "GET /assets/img/home-logo.png HTTP/1.1" 304 -
    - 10.223.157.186 - [15/Jul/2009:20:50:33 -0700] "GET /assets/img/dummy/primary-news-2.jpg HTTP/1.1" 304 -
    - 10.223.157.186 - [15/Jul/2009:20:50:33 -0700] "GET /assets/img/dummy/primary-news-1.jpg HTTP/1.1" 304 -
    - 10.223.157.186 - [15/Jul/2009:20:50:33 -0700] "GET /assets/img/home-media-block-placeholder.jpg HTTP/1.1" 304 -
- **♣** Open Eclipse IDE for Java Developers 2025-03
  - **➤** Click File Option
    - New-> Java Project
      - Project Name-> ProcessLog -> Select JavaSE-1.8 -> Next -> Finish
    - Right Click on ProcessLog -> new -> Package -> name for package : com.mapreduce.lf -> Finish
    - Right Click on ProcessLog -> Build Path -> configure BuildPath -> Select Libraries -> Add external JARS Files -> c drive -> Hadoop folder -> share -> Hadoop -> Client -> Common -> common : lib -> Yarn -> mapreduce -> Hadoop: hdfs -> Apply and Changes
    - Right Click on com.mapreduce.lf -> new -> Class -> class name:
       Process -> Finish
    - o Add java code Here

```
package com.mapreduce.lf;
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class Process{
 public static class IPMapper
    extends Mapper<Object, Text, Text, IntWritable>{
  private final static IntWritable one = new IntWritable(1);
  private Text ip = new Text();
  public void map(Object key, Text value, Context context
            ) throws IOException, InterruptedException {
   // Assuming the IP address is the first token in each line
   StringTokenizer itr = new StringTokenizer(value.toString());
   if (itr.hasMoreTokens()) {
     ip.set(itr.nextToken());
     context.write(ip, one);
 public static class IntSumReducer
    extends Reducer<Text,IntWritable,Text,IntWritable> {
  private IntWritable result = new IntWritable();
  public void reduce(Text key, Iterable<IntWritable> values,
              Context context
              ) throws IOException, InterruptedException {
   int sum = 0:
   for (IntWritable val : values) {
```

```
sum += val.get();
}
result.set(sum);
context.write(key, result);
}

public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "IP address count");
    job.setJarByClass(Process.class);
    job.setMapperClass(IPMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
```

- Right Click On ProcessLog -> Export : java -> Jar files -> next -> Browse :- Make new Folder With Name : JARFILES : Open-> file name as Process
- Go to CMD AS Run As administrator
  - Start-all.cmd
  - Go to Chrome -> http://localhost:9870



- Utilities -> Browse the file
- Create a input file -> Hadoop fs -mkdir /InputDir
  - ♦ Delete a input file -> hadoop fs -rm -r / InputDir
- Uploading Input Log File -> hadoop fs -put
   C:\Users\Omkar\Documents\Log\InputLog.txt /InputDir
  - ◆ Retrive File From Local Host To CMD :- hadoop fs -ls /InputDir1
- For running the jarFiles-> Copy The JARFILE Path-> hadoop jar C:\Users\Omkar\Documents\JARFILES\Process.jar com.mapreduce.lf/Process /InputDir1/InputLog.txt /OutputDir
- For Display OutPut -> Hadoop dfs -cat /OutputDir1/\*
- Create a New Folder Name OutputDir1 for add output in this file
   -> hdfs dfs -get /OutputDir1/part-r-00000
   C:\Users\Omkar\Documents\OutputDir1\OutputLog.txt
- **♣** Stop All Services :- stop-all.cmd