```
import java.io.IOException;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class LogFileMapper extends Mapper<LongWritable, Text, Text, Text> {

    @Override
    public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException {
        // Convert the input value (log line) to string
        String line = value.toString();

        // Split the log line by space or any other delimiter as per your log format
        String[] parts = line.split("\\s+");

        // Extract the relevant information, such as URL, IP address, timestamp, etc.
        // For example, if the URL is in the third position in the log format:
        String url = parts[2];

        // Emit the URL as the output key and any other relevant information as the output value
        context.write(new Text(url), new Text("1")); // Assuming we're counting occurrences of each
URL
    }
}
```

```
import java.io.IOException;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class LogFileReducer extends Reducer<Text, Text, Text, Text> {

    @Override
    public void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
InterruptedException {
        int count = 0;

        // Iterate through the values and count occurrences
        for (Text value : values) {
            count++;
        }

        // Emit the URL and its count as the output
        context.write(key, new Text(Integer.toString(count)));
    }
}
```

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class LogFileAnalyzer {

    public static void main(String[] args) throws Exception {
        // Create a new job configuration
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Log File Analysis");

        // Set the main class
        job.setJarByClass(LogFileAnalyzer.class);

        // Set Mapper and Reducer classes
        job.setMapperClass(LogFileMapper.class);
        job.setReducerClass(LogFileReducer.class);

        // Set input and output format classes
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        // Set output key and value classes
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

        // Set input and output paths
        TextInputFormat.setInputPaths(job, new Path(args[0]));
```

```
        TextOutputFormat.setOutputPath(job, new Path(args[1]));

        // Submit the job and wait for completion
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

192.168.0.1 - - [25/Mar/2024:12:34:56 +0000] "GET /example-page HTTP/1.1" 200 1234

192.168.0.2 - - [25/Mar/2024:12:35:02 +0000] "POST /submit-form HTTP/1.1" 302 -

192.168.0.3 - - [25/Mar/2024:12:35:10 +0000] "GET /images/logo.png HTTP/1.1" 304 0

192.168.0.4 - - [25/Mar/2024:12:35:15 +0000] "GET /example-page HTTP/1.1" 200 5678

```
javac -cp "/home/kumawat_sunil/hadoop-
3.4.0/share/hadoop/common/*:/home/kumawat_sunil/hadoop-
3.4.0/share/hadoop/hdfs/*:/home/kumawat_sunil/Hadoop-
3.4.0/share/hadoop/mapreduce/*:/home/kumawat_sunil/hadoop-3.4.0/share/hadoop/yarn/*"
LogFileMapper.java LogFileReducer.java LogFileAnalyzer.java
```

```
jar -cvf log_analysis.jar LogFileMapper.class LogFileReducer.class LogFileAnalyzer.class
```

```
hadoop jar log_analysis.jar LogFileAnalyzer /input/data.txt /output
```