The Deliverables folder contains the following artifacts-

* CS6240\_Rakesh\_Buchan Krishna Murthy\_1 – zip folder containing the source code for weather analysis programs
* Report File
* Log files generated at EMR on executing WordCount.java – syslog & controller
* Output files generated at EMR on executing WordCount.java – part-r-00000, part-r-00001, part-r-00002
* Readme file

**To execute Weather Analysis code:**

1. Create a java project
2. Copy the folder src in the CS6240\_Rakesh\_Buchan Krishna Murthy\_1 folder into the newly created java project
3. Run the FileLoader.java class
4. Enter the input file name with the path when prompted for
5. The Max, Min, Average Calculation times for each of the approaches will be printed on the console. (with Fibonacci – iterative)

**To Execute Word Count Program on Standalone:**

1. Create a Maven Project
2. Set groupId to MR and artifactId to Assignment1
3. Copy the WordCount.java file extracted from Hadoop download and paste it under src/main/java folder location
4. Update the pom.xml as per the embedded document



1. Create a makefile in the same location as pom.xml and add the content as per the embedded document



1. Open the terminal at the folder location of makefile.txt and run command “make run”
2. Run the WordCount.java by choosing Run as – Run Configurations; provide the path of input file and the path where the output file needs to be generated in Arguments tab
3. Creates the output files at the specified location
4. Also generates a WordCount.jar at the location specified in the makefile

**To execute WordCount on AWS EMR:**

1. Login to the account
2. Create a Bucket in S3
3. Upload WordCount.jar
4. Create Input, Output and logs folder
5. Upload Input file in Input Directory
6. Open EMR Console
7. Create Cluster – provide a name, check logging, select the previously configured s3 log folder for logging and also select hardware configurations as needed
8. Set Launch mode as Step Execution and Step Type would be Custom JAR and configure the jar location from S3, add s3 input/output folders to arguments and action on failure would be to terminate
9. The program would start executing and we can track the progress too.
10. The output files and log files will be created on completion of program execution.