Work Integrated
Learning Programmes

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Computer Science & Information Systems

Real Time Analytics / Stream Processing & Analytics

Apache Storm Lab Sheet 3

WordCount Application with Storm

1. Objective:

Students should be able to

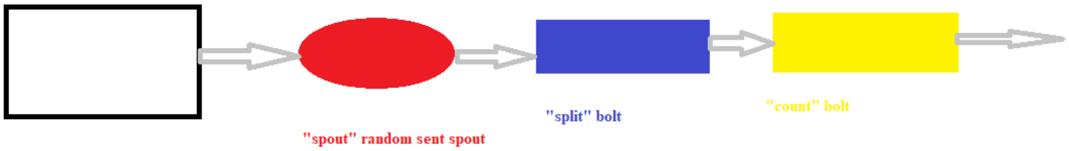
A. Get familiarity with the working of Storm Application

B. Get hands-on experience writing Java program for Streams processing using Storm Topology consisting of Spout and Bolts

The logic for a real-time application is packaged into a Storm topology. A Storm topology is analogous to a MapReduce job. One key difference is that a MapReduce job eventually finishes, whereas a topology runs forever (or until you kill it, of course). A topology is a graph of spouts and bolts that are connected with stream groupings. The stream is the core abstraction in Storm. A stream is an unbounded sequence of tuples that is processed and created in parallel in a distributed fashion. Streams are defined with a schema that names the fields in the stream's tuples. By default, tuples can contain integers, longs, shorts, bytes, strings, doubles, floats, booleans, and byte arrays. You can also define your own sterilizers so that custom types can be used natively within tuples.

A spout is a source of streams in a topology. Generally spouts will read tuples from an external source and emit them into the topology (e.g. a Kestrel queue or the Twitter API). Spouts can either be reliable or unreliable. A reliable spout is capable of replaying a tuple if it failed to be processed by Storm, whereas an unreliable spout forgets about the tuple as soon as it is emitted. All processing in topologies is done in bolts. Bolts can do anything from filtering, functions, aggregations, joins, talking to databases, and more. Bolts can do simple stream transformations. Doing complex stream transformations often requires multiple steps and thus multiple bolts. Part of defining a topology is specifying for each bolt which streams it should receive as input. A stream grouping defines how that stream should be partitioned among the bolt's tasks.

This lab sheet will introduce students with usage of Storm Topology with Java. The application that will be taken as example is word count application. The topology will consist of a spout and two bolts. The spout will serve as abstraction for the real world. Its responsibility is to provide a sentence at random from the given list of sentences. The assumption made here is that the sentences are the records received from the real world. Spout is helping us to accept them and making it available for further processing. The first bolt is the topology will be responsible for splitting the sentence into words and forwarding the words to the next bolt. The last bolt will keep track of the count of the words it has received as input. The topology of the word count application can be visualized as shown below.

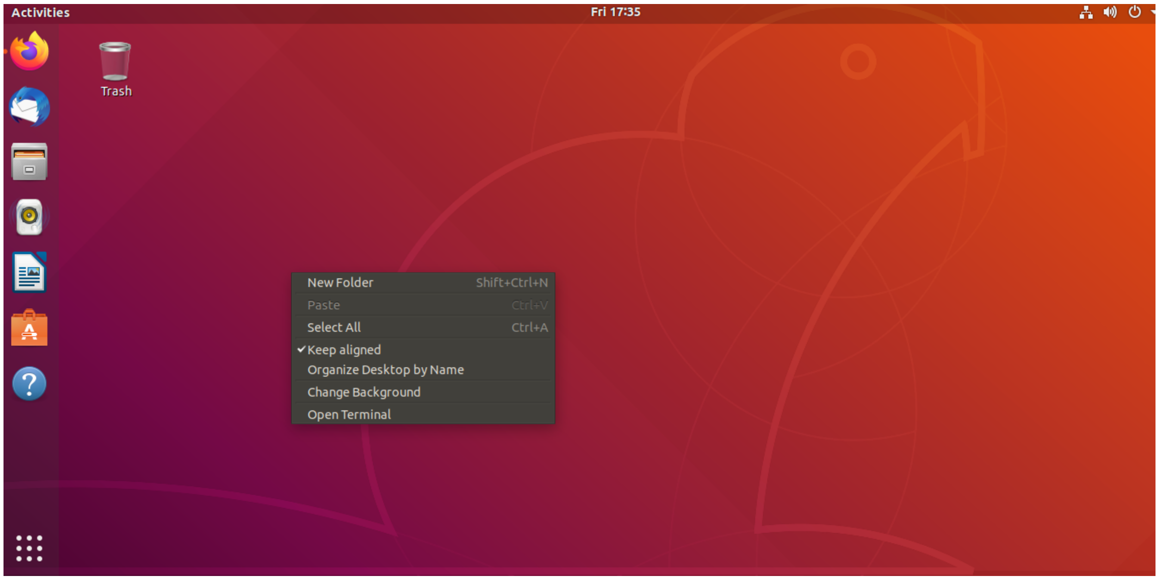


2. Steps to be performed:

Note - It's assumed that student has made a slot reservation using the slot booking interface where Apache Storm framework was selected. The details of the Apache Strom systems to be used is received through an email. If not, please contact the administrators for the same.

Also it's assumed that students are aware of the process of logging into these virtual machines. If not, then get access to the user manual maintained for the usage of remote lab setup.

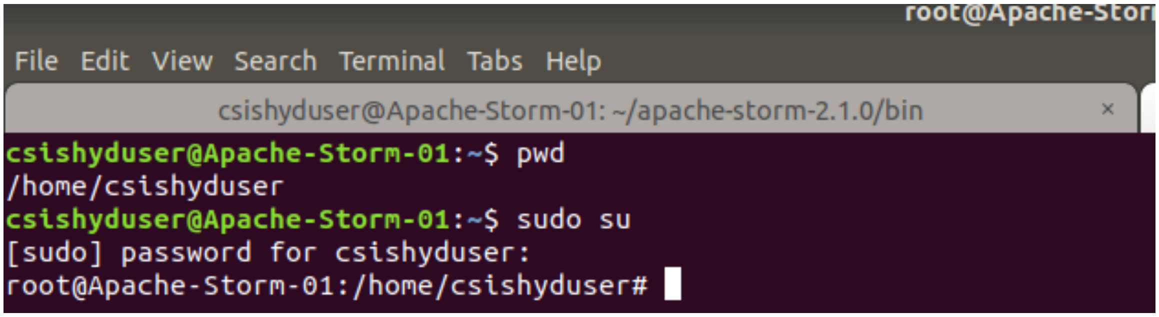
A. Open the terminal by right clicking on the desktop of the virtual machine.



B. Login as sudo user.

>>> sudo su

Provide the password provided in the email received from BITS remote lab team.

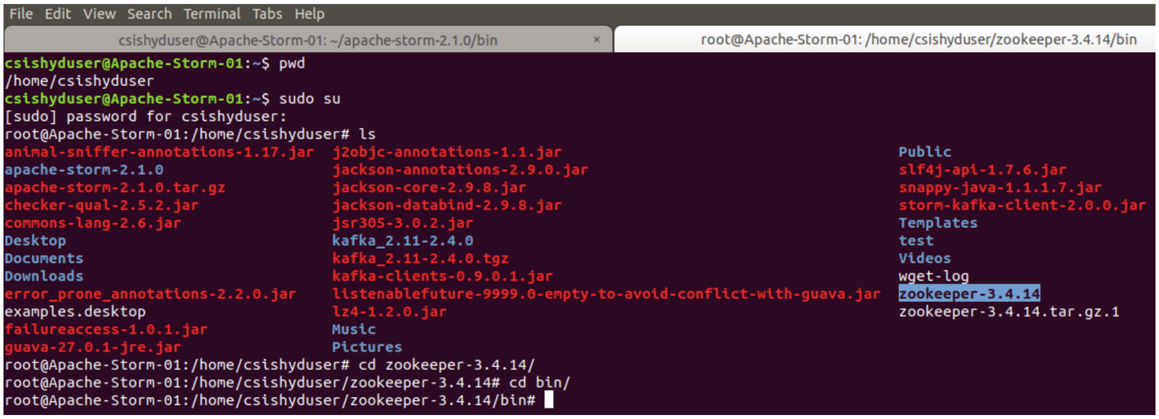

root@Apache-Stor
File Edit View Search Terminal Tabs Help
csishyduser@Apache-Storm-01: ~/apache-storm-2.1.0/bin

```
csishyduser@Apache-Storm-01:~$ pwd
/home/csishyduser
csishyduser@Apache-Storm-01:~$ sudo su
[sudo] password for csishyduser:
root@Apache-Storm-01:/home/csishyduser#
```

C. Look at the current working directory using the “pwd” command. Then change the directory to the Zookeepers bin directory.

>>> pwd

>>> cd zookeeper-3.4.14/bin/

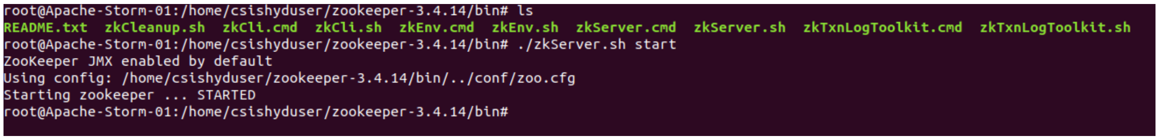


File Edit View Search Terminal Tabs Help
csishyduser@Apache-Storm-01: ~/apache-storm-2.1.0/bin
root@Apache-Storm-01: /home/csishyduser/zookeeper-3.4.14/bin

```
csishyduser@Apache-Storm-01:~$ pwd
/home/csishyduser
csishyduser@Apache-Storm-01:~$ sudo su
[sudo] password for csishyduser:
root@Apache-Storm-01:/home/csishyduser# ls
animal-sniffer-annotations-1.17.jar  j2objc-annotations-1.1.jar                                      Public
apache-storm-2.1.0                   jackson-annotations-2.9.0.jar                                   slf4j-api-1.7.6.jar
apache-storm-2.1.0.tar.gz            jackson-core-2.9.8.jar                                          snappy-java-1.1.1.7.jar
checker-qual-2.5.2.jar               jackson-databind-2.9.8.jar                                      storm-kafka-client-2.0.0.jar
commons-lang-2.6.jar                 jsr305-3.0.2.jar                                                Templates
Desktop                              kafka_2.11-2.4.0                                                test
Documents                            kafka_2.11-2.4.0.tgz                                            Videos
Downloads                            kafka-clients-0.9.0.1.jar                                       wget-log
error_prone_annotations-2.2.0.jar    listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar  zookeeper-3.4.14
examples.desktop                     lz4-1.2.0.jar                                                   zookeeper-3.4.14.tar.gz.1
failureaccess-1.0.1.jar              Music
guava-27.0.1-jre.jar                 Pictures
root@Apache-Storm-01:/home/csishyduser# cd zookeeper-3.4.14/
root@Apache-Storm-01:/home/csishyduser/zookeeper-3.4.14# cd bin/
root@Apache-Storm-01:/home/csishyduser/zookeeper-3.4.14/bin# 
```

D.   Start the zookeeper.

>>> ./zkServer.sh start

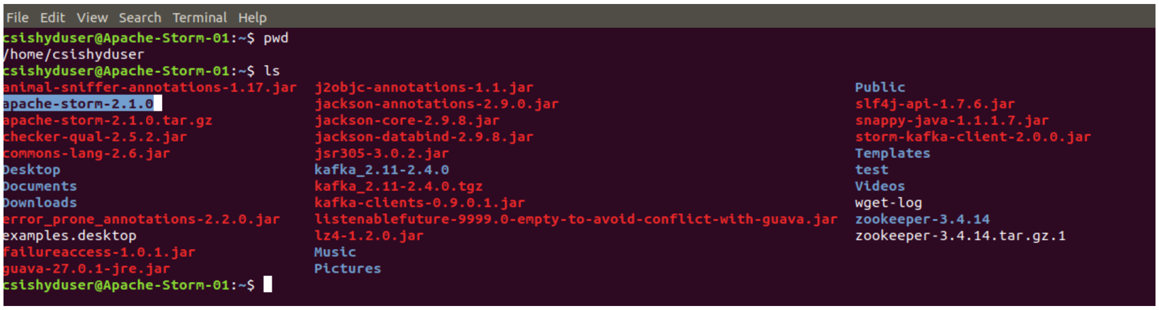


```
root@Apache-Storm-01:/home/csishyduser/zookeeper-3.4.14/bin# ls
README.txt  zkCleanup.sh  zkCli.cmd  zkCli.sh  zkEnv.cmd  zkEnv.sh  zkServer.cmd  zkServer.sh  zkTxnLogToolkit.cmd  zkTxnLogToolkit.sh
root@Apache-Storm-01:/home/csishyduser/zookeeper-3.4.14/bin# ./zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /home/csishyduser/zookeeper-3.4.14/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
root@Apache-Storm-01:/home/csishyduser/zookeeper-3.4.14/bin#
```

E.   Open another terminal. Look at the current working directory using the “pwd” command. Then change the directory to the Storms directory.

>>> pwd

>>> cd apache-storm-2.1.0/

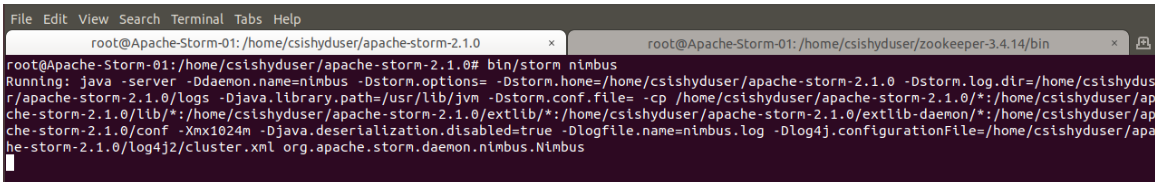


```
csishyduser@Apache-Storm-01:~$ pwd
/home/csishyduser
csishyduser@Apache-Storm-01:~$ ls
animal-sniffer-annotations-1.17.jar  j2objc-annotations-1.1.jar                                      Public
apache-storm-2.1.0                   jackson-annotations-2.9.0.jar                                   slf4j-api-1.7.6.jar
apache-storm-2.1.0.tar.gz            jackson-core-2.9.8.jar                                          snappy-java-1.1.1.7.jar
checker-qual-2.5.2.jar               jackson-databind-2.9.8.jar                                      storm-kafka-client-2.0.0.jar
commons-lang-2.6.jar                 jsr305-3.0.2.jar                                                Templates
Desktop                              kafka_2.11-2.4.0                                                test
Documents                            kafka_2.11-2.4.0.tgz                                            Videos
Downloads                            kafka-clients-0.9.0.1.jar                                       wget-log
error_prone_annotations-2.2.0.jar    listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar  zookeeper-3.4.14
examples.desktop                     lz4-1.2.0.jar                                                   zookeeper-3.4.14.tar.gz.1
failureaccess-1.0.1.jar              Music
guava-27.0.1-jre.jar                 Pictures
csishyduser@Apache-Storm-01:~$ 
```

F.   Start the nimbus node using the storm command.

>>> bin/strom nimbus

Note – if the command fails, then login as sudo su and then try again.

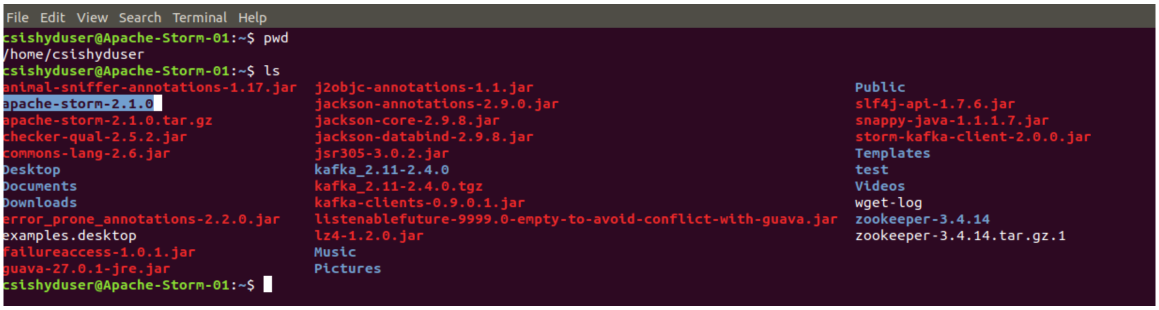

File Edit View Search Terminal Tabs Help
root@Apache-Storm-01: /home/csishyduser/apache-storm-2.1.0
root@Apache-Storm-01: /home/csishyduser/zookeeper-3.4.14/bin

```
root@Apache-Storm-01:/home/csishyduser/apache-storm-2.1.0# bin/storm nimbus
Running: java -server -Ddaemon.name=nimbus -Dstorm.options= -Dstorm.home=/home/csishyduser/apache-storm-2.1.0 -Dstorm.log.dir=/home/csishydus
r/apache-storm-2.1.0/logs -Djava.library.path=/usr/lib/jvm -Dstorm.conf.file= -cp /home/csishyduser/apache-storm-2.1.0/*:/home/csishyduser/ap
che-storm-2.1.0/lib/*:/home/csishyduser/apache-storm-2.1.0/extlib/*:/home/csishyduser/apache-storm-2.1.0/extlib-daemon/*:/home/csishyduser/ap
che-storm-2.1.0/conf -Xmx1024m -Djava.deserialization.disabled=true -Dlogfile.name=nimbus.log -Dlog4j.configurationFile=/home/csishyduser/apa
he-storm-2.1.0/log4j2/cluster.xml org.apache.storm.daemon.nimbus.Nimbus
```

G. Open another terminal. Look at the current working directory using the “pwd” command. Then change the directory to the Storms directory.

>>> pwd

>>> cd apache-storm-2.1.0/

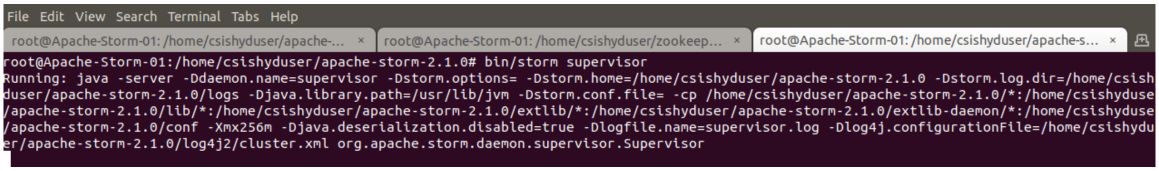


```
csishyduser@Apache-Storm-01:~$ pwd
/home/csishyduser
csishyduser@Apache-Storm-01:~$ ls
animal-sniffer-annotations-1.17.jar  j2objc-annotations-1.1.jar                                        Public
apache-storm-2.1.0                   jackson-annotations-2.9.0.jar                                     slf4j-api-1.7.6.jar
apache-storm-2.1.0.tar.gz            jackson-core-2.9.8.jar                                            snappy-java-1.1.1.7.jar
checker-qual-2.5.2.jar               jackson-databind-2.9.8.jar                                        storm-kafka-client-2.0.0.jar
commons-lang-2.6.jar                 jsr305-3.0.2.jar                                                  Templates
Desktop                              kafka_2.11-2.4.0                                                  test
Documents                            kafka_2.11-2.4.0.tgz                                              Videos
Downloads                            kafka-clients-0.9.0.1.jar                                         wget-log
error_prone_annotations-2.2.0.jar    listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar  zookeeper-3.4.14
examples.desktop                     lz4-1.2.0.jar                                                     zookeeper-3.4.14.tar.gz.1
failureaccess-1.0.1.jar              Music
guava-27.0.1-jre.jar                 Pictures
csishyduser@Apache-Storm-01:~$
```

H. Start the supervisor node using the storm command.

>>> bin/strom suporvisor

Note – if the command fails, then login as sudo su and then try again.


File Edit View Search Terminal Tabs Help
root@Apache-Storm-01: /home/csishyduser/apache-...
root@Apache-Storm-01: /home/csishyduser/zookeep...
root@Apache-Storm-01: /home/csishyduser/apache-s...

```
root@Apache-Storm-01:/home/csishyduser/apache-storm-2.1.0# bin/storm supervisor
Running: java -server -Ddaemon.name=supervisor -Dstorm.options= -Dstorm.home=/home/csishyduser/apache-storm-2.1.0 -Dstorm.log.dir=/home/csish
duser/apache-storm-2.1.0/logs -Djava.library.path=/usr/lib/jvm -Dstorm.conf.file= -cp /home/csishyduser/apache-storm-2.1.0/*:/home/csishyduse
/apache-storm-2.1.0/lib/*:/home/csishyduser/apache-storm-2.1.0/extlib/*:/home/csishyduser/apache-storm-2.1.0/extlib-daemon/*:/home/csishyduse
/apache-storm-2.1.0/conf -Xmx256m -Djava.deserialization.disabled=true -Dlogfile.name=supervisor.log -Dlog4j.configurationFile=/home/csishydu
er/apache-storm-2.1.0/log4j2/cluster.xml org.apache.storm.daemon.supervisor.Supervisor
```

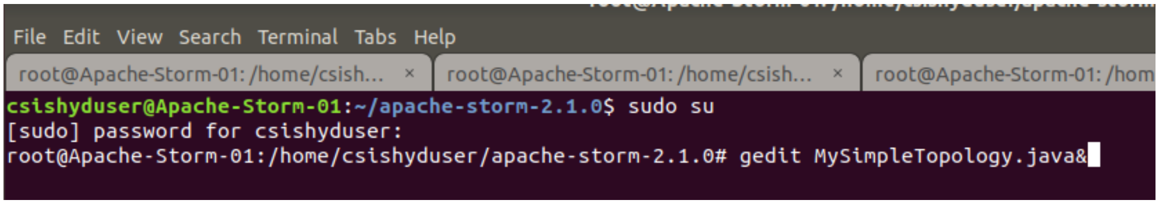
I. Open another terminal. Look at the current working directory using the “pwd” command. Then change the directory to the Storms directory.

>>> pwd

>>> cd apache-storm-2.1.0/

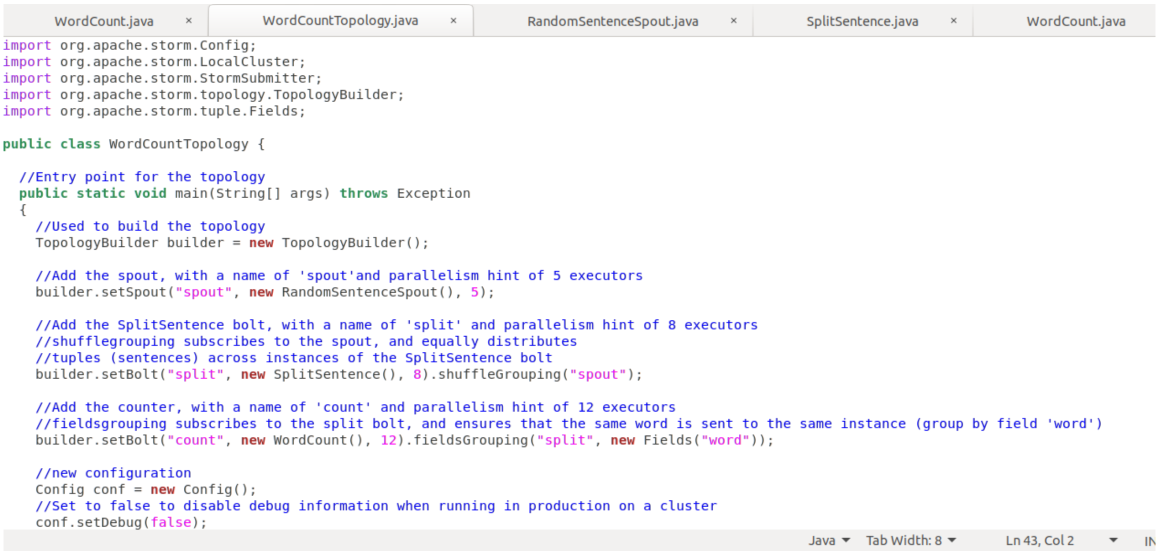
J. Open up gedit editor for writing the Java code.

>>> gedit WordCountTopology.java&


File Edit View Search Terminal Tabs Help
root@Apache-Storm-01: /home/csish... ×
root@Apache-Storm-01: /home/csish... ×
root@Apache-Storm-01: /hom

```
csishyduser@Apache-Storm-01:~/apache-storm-2.1.0$ sudo su
[sudo] password for csishyduser:
root@Apache-Storm-01:/home/csishyduser/apache-storm-2.1.0# gedit MySimpleTopology.java&
```

K. Copy paste the content of attached WordCountTopology.java file into the file opened in the geditor.



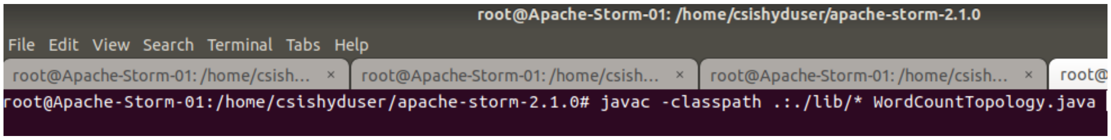
```
import org.apache.storm.Config;
import org.apache.storm.LocalCluster;
import org.apache.storm.StormSubmitter;
import org.apache.storm.topology.TopologyBuilder;
import org.apache.storm.tuple.Fields;

public class WordCountTopology {

  //Entry point for the topology
  public static void main(String[] args) throws Exception
  {
    //Used to build the topology
    TopologyBuilder builder = new TopologyBuilder();

    //Add the spout, with a name of 'spout'and parallelism hint of 5 executors
    builder.setSpout("spout", new RandomSentenceSpout(), 5);

    //Add the SplitSentence bolt, with a name of 'split' and parallelism hint of 8 executors
    //shufflegrouping subscribes to the spout, and equally distributes
    //tuples (sentences) across instances of the SplitSentence bolt
    builder.setBolt("split", new SplitSentence(), 8).shuffleGrouping("spout");

    //Add the counter, with a name of 'count' and parallelism hint of 12 executors
    //fieldsgrouping subscribes to the split bolt, and ensures that the same word is sent to the same instance (group by field 'word')
    builder.setBolt("count", new WordCount(), 12).fieldsGrouping("split", new Fields("word"));

    //new configuration
    Config conf = new Config();
    //Set to false to disable debug information when running in production on a cluster
    conf.setDebug(false);
```

L. Repeat the step J and K for two other java files namely

- RandomSentenceSpout.java
- SplitSentence.java
- WordCount.java

M. Compile the MySimpleTopology.java class which has the topology definition.

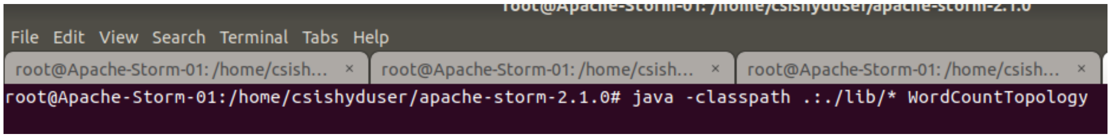
>>> javac -classpath .:./lib/* WordCountTopology.java


root@Apache-Storm-01: /home/csishyduser/apache-storm-2.1.0
File Edit View Search Terminal Tabs Help
root@Apache-Storm-01: /home/csish... ×
root@Apache-Storm-01: /home/csish... ×
root@Apache-Storm-01: /home/csish... ×
root@

```
root@Apache-Storm-01:/home/csishyduser/apache-storm-2.1.0# javac -classpath .:./lib/* WordCountTopology.java
```

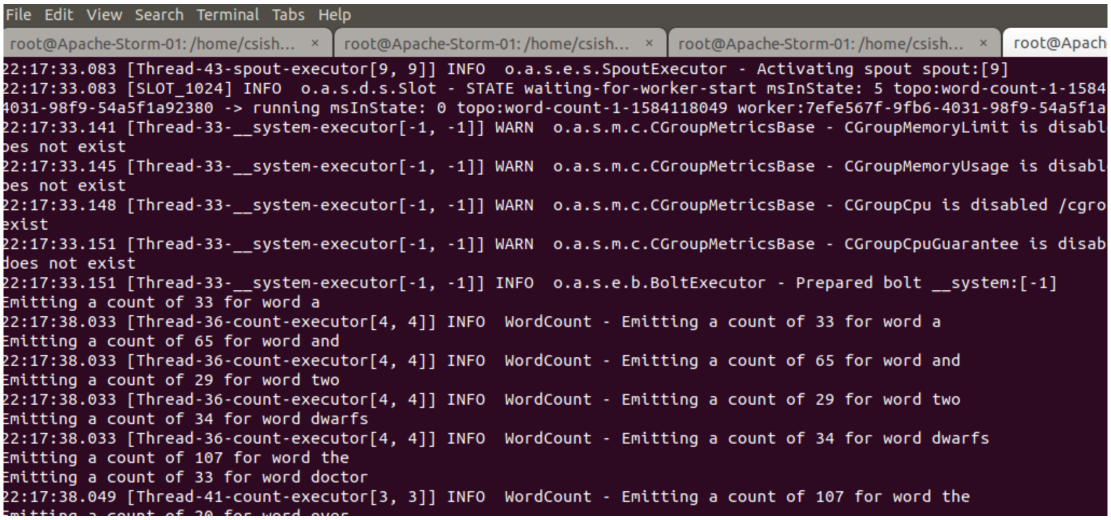
N. Run the MySimpleTopology Storm application and observe the output.

>>> java -classpath .:./lib/* WordCountTopology


root@Apache-Storm-01: /home/csishyduser/apache-storm-2.1.0
File Edit View Search Terminal Tabs Help
root@Apache-Storm-01: /home/csish... ×
root@Apache-Storm-01: /home/csish... ×
root@Apache-Storm-01: /home/csish... ×

```
root@Apache-Storm-01:/home/csishyduser/apache-storm-2.1.0# java -classpath .:./lib/* WordCountTopology
```

1. In the output you must be seeing the lines shown below which shows that the random sentences is generated in the Spout and getting processed with the Bolt defined where its broken down into the words and then the count of them is getting printed on the console.


File Edit View Search Terminal Tabs Help
root@Apache-Storm-01: /home/csish... ×
root@Apache-Storm-01: /home/csish... ×
root@Apache-Storm-01: /home/csish... ×
root@Apach

```
22:17:33.083 [Thread-43-spout-executor[9, 9]] INFO  o.a.s.e.s.SpoutExecutor - Activating spout spout:[9]
22:17:33.083 [SLOT_1024] INFO  o.a.s.d.s.Slot - STATE waiting-for-worker-start msInState: 5 topo:word-count-1-1584
4031-98f9-54a5f1a92380 -> running msInState: 0 topo:word-count-1-1584118049 worker:7efe567f-9fb6-4031-98f9-54a5f1a
22:17:33.141 [Thread-33-__system-executor[-1, -1]] WARN  o.a.s.m.c.CGroupMetricsBase - CGroupMemoryLimit is disabl
oes not exist
22:17:33.145 [Thread-33-__system-executor[-1, -1]] WARN  o.a.s.m.c.CGroupMetricsBase - CGroupMemoryUsage is disabl
oes not exist
22:17:33.148 [Thread-33-__system-executor[-1, -1]] WARN  o.a.s.m.c.CGroupMetricsBase - CGroupCpu is disabled /cgro
exist
22:17:33.151 [Thread-33-__system-executor[-1, -1]] WARN  o.a.s.m.c.CGroupMetricsBase - CGroupCpuGuarantee is disab
does not exist
22:17:33.151 [Thread-33-__system-executor[-1, -1]] INFO  o.a.s.e.b.BoltExecutor - Prepared bolt __system:[-1]
Emitting a count of 33 for word a
22:17:38.033 [Thread-36-count-executor[4, 4]] INFO  WordCount - Emitting a count of 33 for word a
Emitting a count of 65 for word and
22:17:38.033 [Thread-36-count-executor[4, 4]] INFO  WordCount - Emitting a count of 65 for word and
Emitting a count of 29 for word two
22:17:38.033 [Thread-36-count-executor[4, 4]] INFO  WordCount - Emitting a count of 29 for word two
Emitting a count of 34 for word dwarfs
22:17:38.033 [Thread-36-count-executor[4, 4]] INFO  WordCount - Emitting a count of 34 for word dwarfs
Emitting a count of 107 for word the
Emitting a count of 33 for word doctor
22:17:38.049 [Thread-41-count-executor[3, 3]] INFO  WordCount - Emitting a count of 107 for word the
Emitting a count of 20 for word over
```

O. In order to run the code continuously comment out the following line in the WordCountTopology.java file, recompile and execute the program to see the continuous output.

- cluster.shutdown();

3. Outputs/Results:

Students should be able to write a Storm application

- To read the tuples / records from the external sources through Spout
- To do word counting on the tuples through the Bolt logic
- To execute the Storm Topology on the local cluster

4. Observations:

Students carefully needs to observe the code written for the usage of Storm API for

- Building the Spout and data handling with it
- Writing the Bolt and process the data with it
- Building the topology with Spout and Bolt and executing it on cluster

5. References:

a. Storm Documentation

b. Strom Tutorial