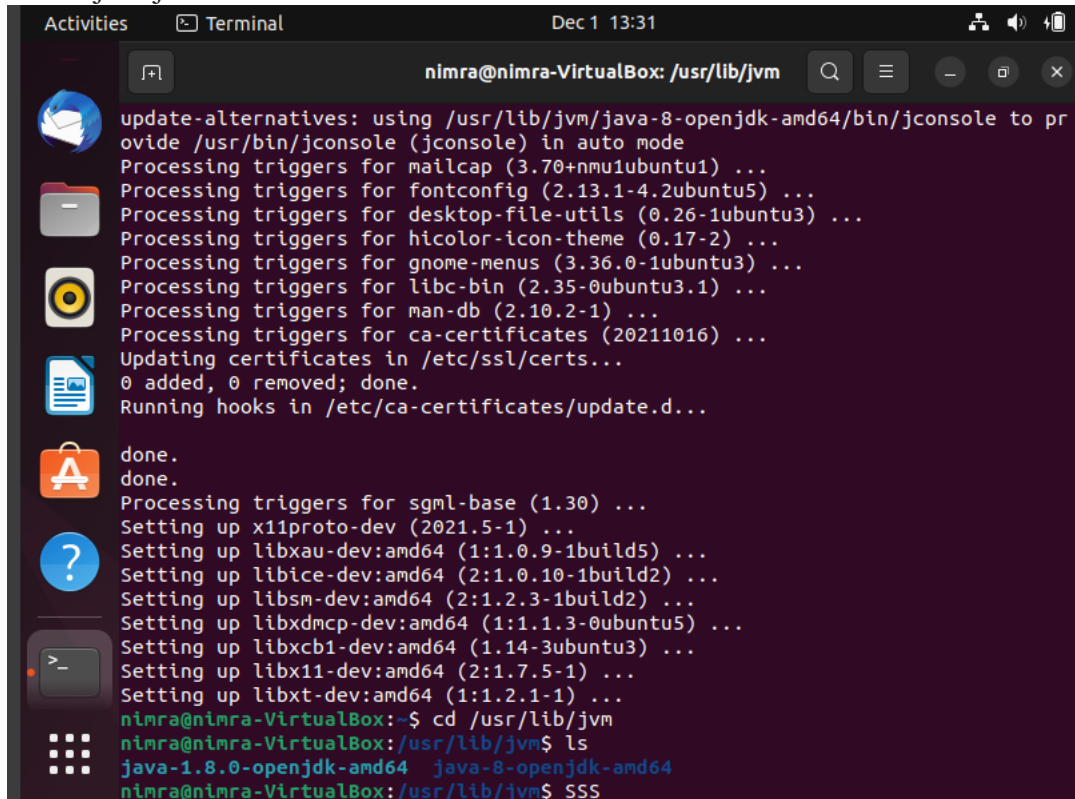


Installing Software

Step 1:

Install java jdk 8



```
nimra@nimra-VirtualBox: /usr/lib/jvm
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jconsole to provide /usr/bin/jconsole (jconsole) in auto mode
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ca-certificates (20211016) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
done.
Processing triggers for sgml-base (1.30) ...
Setting up x11proto-dev (2021.5-1) ...
Setting up libxau-dev:amd64 (1:1.0.9-1build5) ...
Setting up libice-dev:amd64 (2:1.0.10-1build2) ...
Setting up libsm-dev:amd64 (2:1.2.3-1build2) ...
Setting up libxdmcp-dev:amd64 (1:1.1.3-0ubuntu5) ...
Setting up libxcb1-dev:amd64 (1.14-3ubuntu3) ...
Setting up libx11-dev:amd64 (2:1.7.5-1) ...
Setting up libxt-dev:amd64 (1:1.2.1-1) ...
nimra@nimra-VirtualBox:~$ cd /usr/lib/jvm
nimra@nimra-VirtualBox:/usr/lib/jvm$ ls
java-1.8.0-openjdk-amd64  java-8-openjdk-amd64
nimra@nimra-VirtualBox:/usr/lib/jvm$ SSS
```

Step 2:

To check it's there **cd /usr/lib/jvm**

(ssh — secure shell — protocol used to securely connect to remote server/system — transfers data in encrypted form)

now go to hadoop.apache.org website download the tar file
(hadoop.apache.org — download tar file of hadoop.)

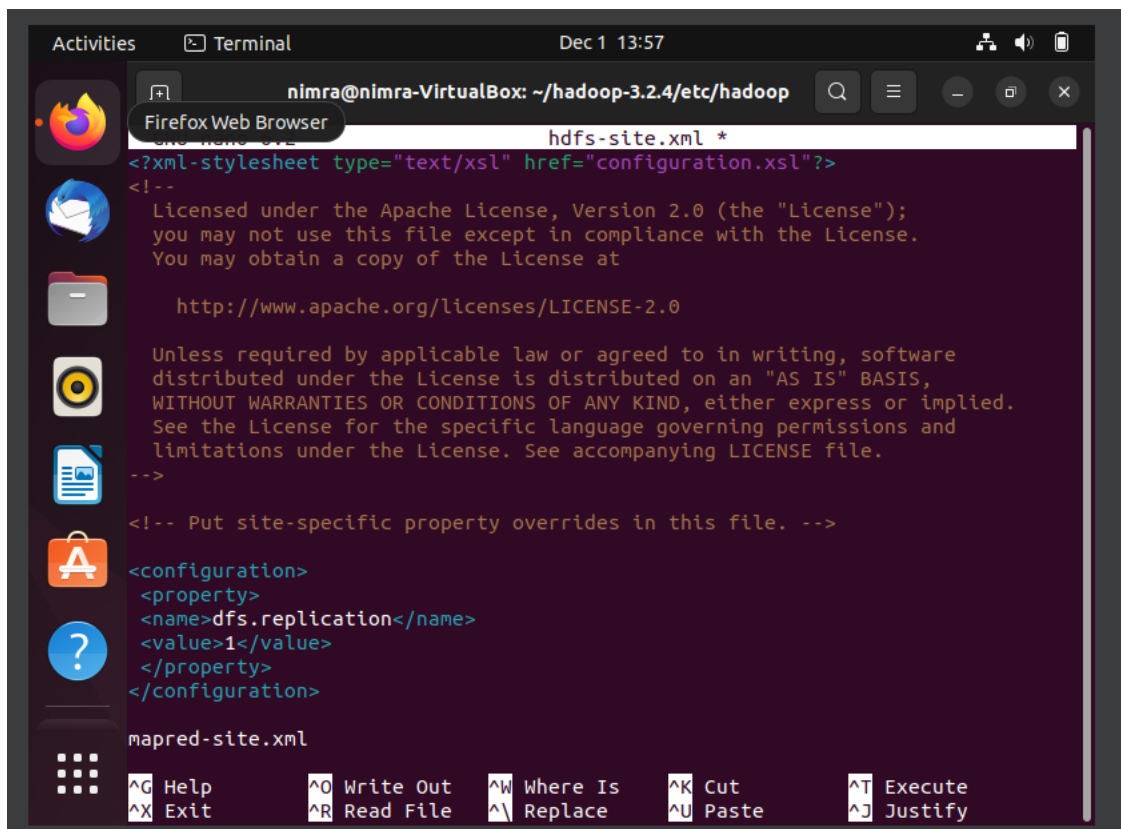
Execution

Step 3:

Format the filesystem:

(Extract the tar file)

cd hadoop-3.2.3/etc/hadoop



The screenshot shows a terminal window titled "nimra@nimra-VirtualBox: ~/hadoop-3.2.4/etc/hadoop". The terminal displays the contents of the "hdfs-site.xml" file. The file is an XML configuration file for Hadoop Distributed File System (HDFS). It includes a license notice for Apache License, Version 2.0, and a configuration section for "dfs.replication" set to 1. The terminal also shows a list of keyboard shortcuts at the bottom.

```
nimra@nimra-VirtualBox: ~/hadoop-3.2.4/etc/hadoop
hdfs-site.xml *
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->

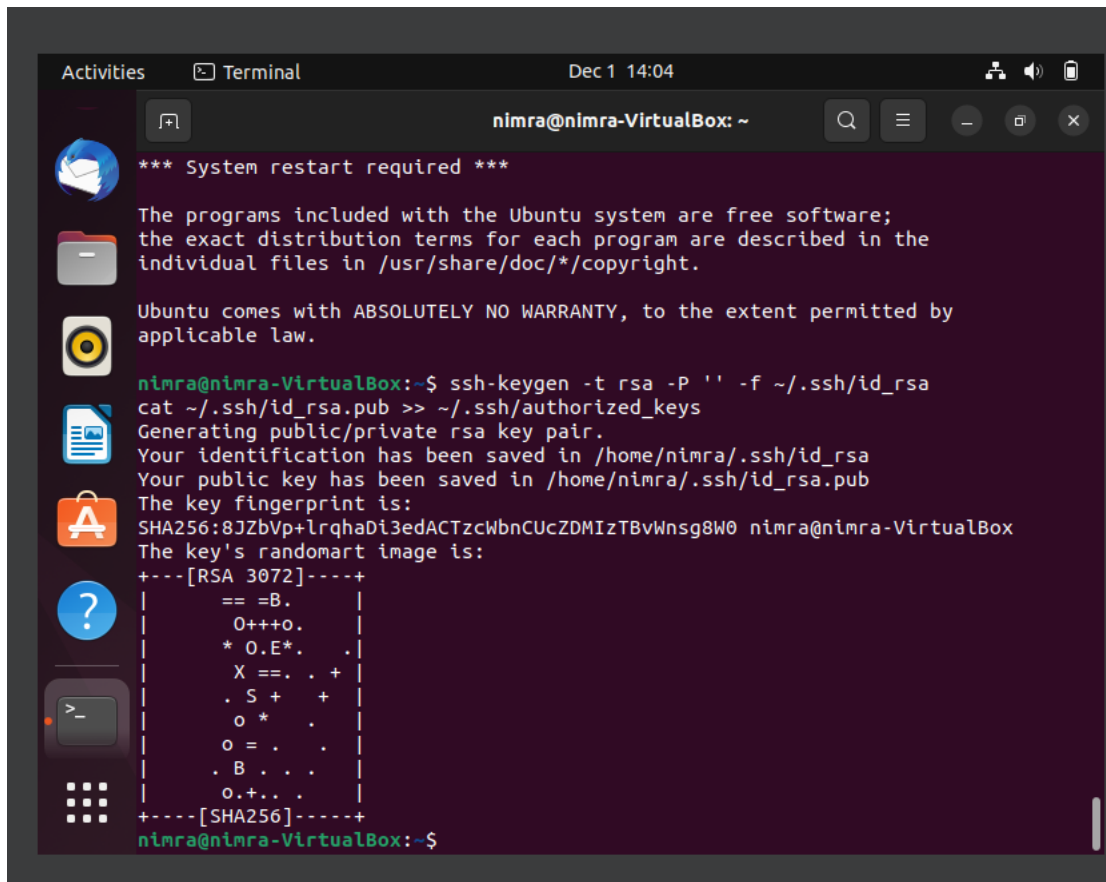
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>

mapred-site.xml
```

Keyboard shortcuts:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^N Replace	^U Paste	^J Justify

Start NameNode daemon and DataNode daemon:



A terminal window titled 'Terminal' with the date 'Dec 1 14:04' and the user 'nimra@nimra-VirtualBox: ~'. The terminal shows the following output:

```
*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

nimra@nimra-VirtualBox:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
Generating public/private rsa key pair.
Your identification has been saved in /home/nimra/.ssh/id_rsa
Your public key has been saved in /home/nimra/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8JZbVp+lrqhaDi3edACTzcWbnCUCZDMizTBvWnsg8W0 nimra@nimra-VirtualBox
The key's randomart image is:
+----[RSA 3072]-----+
  == =B.
  0+++o.
  * O.E*.
  X ==. . +
  . S + +
  o * .
  o = . .
  . B . . .
  o.+...
+-----[SHA256]-----+
nimra@nimra-VirtualBox:~$
```

Step 4:

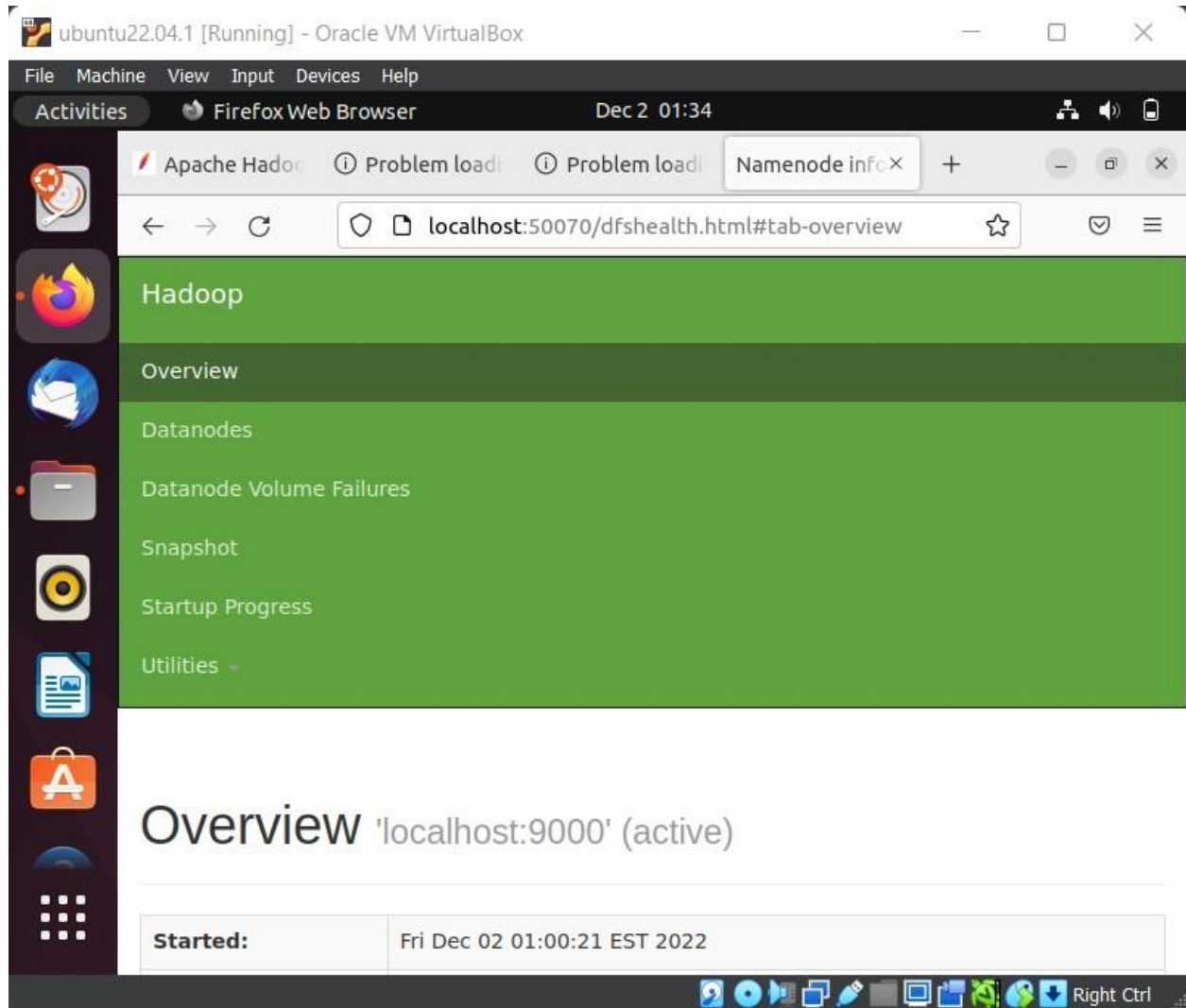
1. Browse the web interface for the NameNode; by default it is available at:
 - o NameNode - <http://localhost:9870/>

1. Make the HDFS directories required to execute MapReduce jobs:

2. `$ bin/hdfs dfs -mkdir /user`
3. `$ bin/hdfs dfs -mkdir /user/<username>`

4. Copy the input files into the distributed filesystem:

5. `$ bin/hdfs dfs -mkdir input`



Step 5:

Run a wordcount MapReduce program using either the example wordcount program in the Hadoop package or write and compile your own wordcount program.

The MapReduce and wordcount can be found at:

<https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>

Use the following file as your input file:

<https://www.gutenberg.org/files/4300/old/ulyss11.txt>

1. Make the HDFS directories required to execute MapReduce jobs:

```
2. export PATH=${JAVA_HOME}/bin:${PATH}
3. export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

copy this command and save this and run the command below.

Compile WordCount.java and create a jar:

```
$ bin/hadoop com.sun.tools.javac.Main WordCount.java
$ jar cf wc.jar WordCount*.class
```

1. Copy the input files into the distributed filesystem:

```
2. $ bin/hdfs dfs -mkdir input
$ bin/hdfs dfs -put etc/hadoop/*.xml input
```

Apache Hadoop Problem load Problem load Namenode info X

localhost:50070/dfshealth.html#tab-overview

Summary

Security is off.

Safemode is off.

3 files and directories, 0 blocks = 3 total filesystem object(s).

Heap Memory used 94.95 MB of 182.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 43.81 MB of 44.59 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	2.34 GB
DFS Used:	8 KB (0%)
Non DFS Used:	1.66 MB
DFS Remaining:	2.34 GB (99.93%)
Block Pool Used:	8 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0)

Activities

Firefox Web Browser

Dec 2 01:35

Apache Hadoop

Problem load

Problem load

Namenode info x

localhost:50070/dfshealth.html#tab-overview

Number of blocks pending deletion

0

Block Deletion Start Time

12/2/2022, 1:00:21 AM

NameNode Journal Status

Current transaction ID: 5

Journal Manager	State
FileJournalManager(root=/tmp/hadoop-ubuntu/dfs/name)	EditLogFileOutputStream(/tmp/hadoop-ubuntu/dfs/name/current/edits_inprogress_0000000000000000003)

NameNode Storage

Storage Directory	Type	State
/tmp/hadoop-ubuntu/dfs/name	IMAGE_AND_EDITS	Active

Right Ctrl

localhost:50070/dfshealth.html#tab-datanode

Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes
ubuntu:50010 (127.0.0.1:50010)	0	In Service	2.34 GB	8 KB	1.66 MB	2.34 GB	0	8 KB (0%)	0

Decomissioning

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks In files under construction
------	--------------	-------------------------	------------------------------	---

Step 6:

Create a Mapper class within the WordCount class which extends MapReduceBase Class to implement mapper interface. The mapper class will contain -

1. Code to implement "map" method.
2. Code for implementing the mapper-stage business logic should be written within this method.

ubuntu@ubuntu: ~/Downloads/hadoop-2.7.3

```
GNU nano 6.2 WordCount.java
    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, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.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);
}

export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

ubuntu22.04.1 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Dec 2 14:37

ubuntu@ubuntu: ~/Downloads/hadoop-2.7.3

```
at org.apache.hadoop.mapreduce.Job.submit(Job.java:1287)
at org.apache.hadoop.mapreduce.Job.waitForCompletion(Job.java:1308)
at WordCount.main(WordCount.java:59)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl
.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcce
ssorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.apache.hadoop.util.RunJar.run(RunJar.java:221)
at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
Caused by: java.net.ConnectException: Connection refused
at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)
at sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:71
6)
at org.apache.hadoop.net.SocketIOWithTimeout.connect(SocketIOWithTimeou
t.java:206)
at org.apache.hadoop.net.NetUtils.connect(NetUtils.java:531)
at org.apache.hadoop.net.NetUtils.connect(NetUtils.java:495)
at org.apache.hadoop.ipc.Client$Connection.setupConnection(Client.java:
614)
at org.apache.hadoop.ipc.Client$Connection.setupIOstreams(Client.java:7
12)
at org.apache.hadoop.ipc.Client$Connection.access$2900(Client.java:375)
at org.apache.hadoop.ipc.Client.getConnection(Client.java:1528)
at org.apache.hadoop.ipc.Client.call(Client.java:1451)
... 29 more
```