**Hadoop MapReduce WordCount Job Execution (Pseudo-Distributed Mode)**

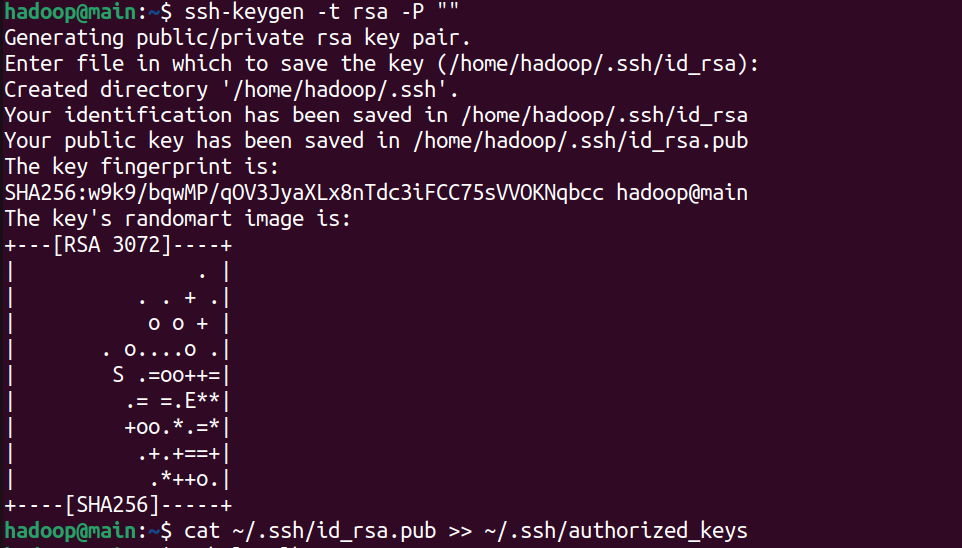
**Objective**

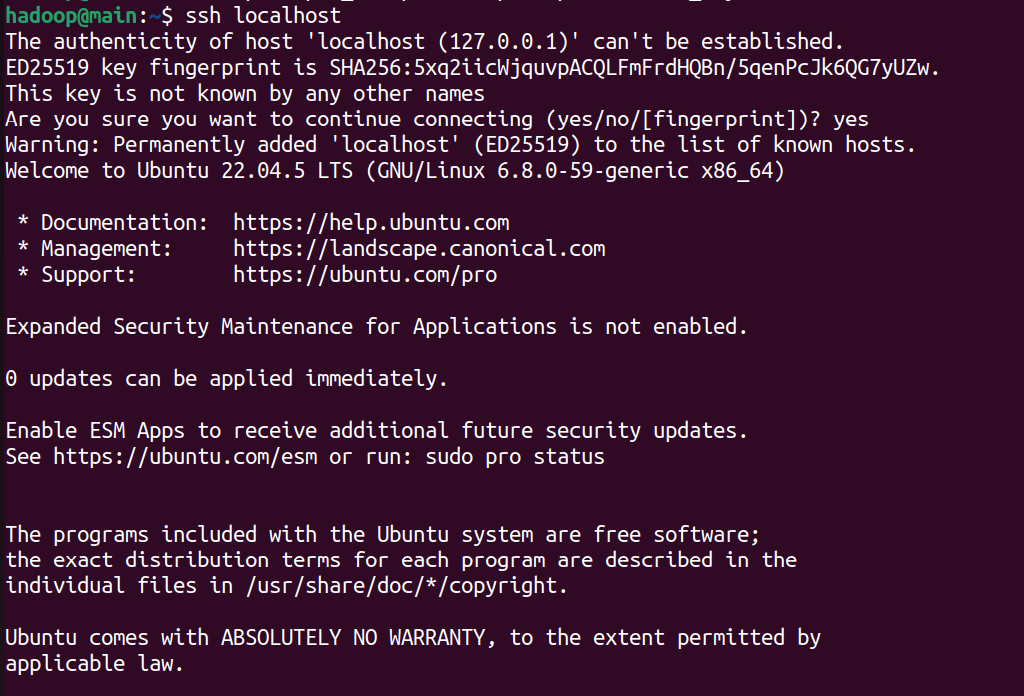
To configure a single-node Hadoop environment in pseudo-distributed mode and execute a built-in MapReduce WordCount job using Hadoop configuration XML files as input data.

**Environment Details**

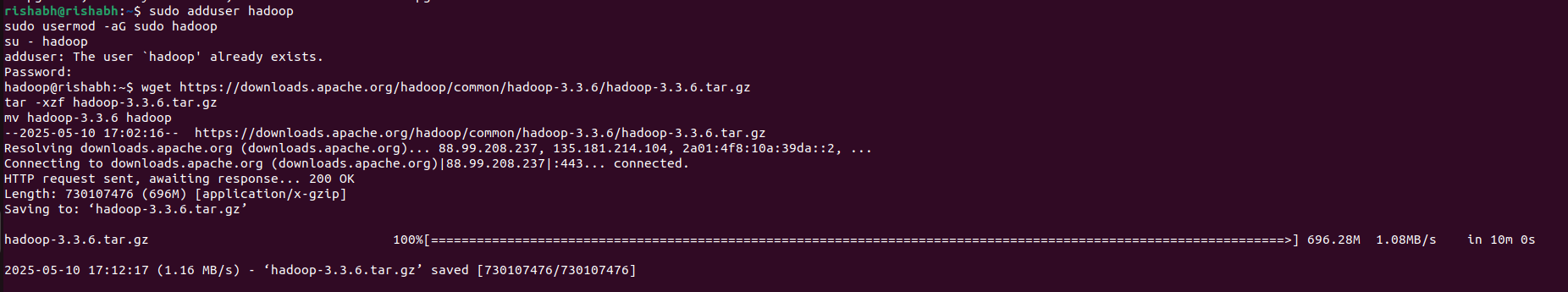
* **Execution Mode:** Pseudo-Distributed Mode
* **User:** hadoop
* **OS:** Ubuntu 22.04 LTS
* **Hadoop Version:** 3.3.6
* **Java Version:** OpenJDK 11
* **Hostname:** main
* **HDFS URI:** hdfs://localhost:9000

**SSH server setup for the communication between Hadoop Daemons**

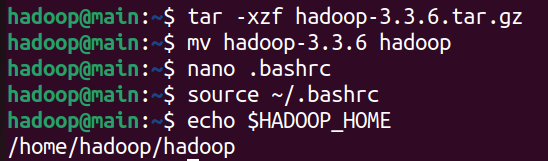




**Hadoop download and Environment variable setup**

**Environment variables:**

* export HADOOP\_HOME=~/hadoop
* export HADOOP\_INSTALL=$HADOOP\_HOME
* export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME
* export HADOOP\_COMMON\_HOME=$HADOOP\_HOME
* export HADOOP\_HDFS\_HOME=$HADOOP\_HOME
* export YARN\_HOME=$HADOOP\_HOME
* export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native
* export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64
* export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin



**Pseudo-Distributed Mode Setup Summary**

Each Hadoop daemon (NameNode, DataNode, ResourceManager, NodeManager) runs as a separate Java process on the same node.

**Configurations modified across the following files:**

* core-site.xml
* hdfs-site.xml
* yarn-site.xml
* mapred-site.xml

**Key settings include:**

* fs.defaultFS = hdfs://localhost:9000
* mapreduce.framework.name = yarn
* Proper paths set for HADOOP\_MAPRED\_HOME in mapred-site.xml

**Execution Steps**

**1. Prepare Local Input Directory**

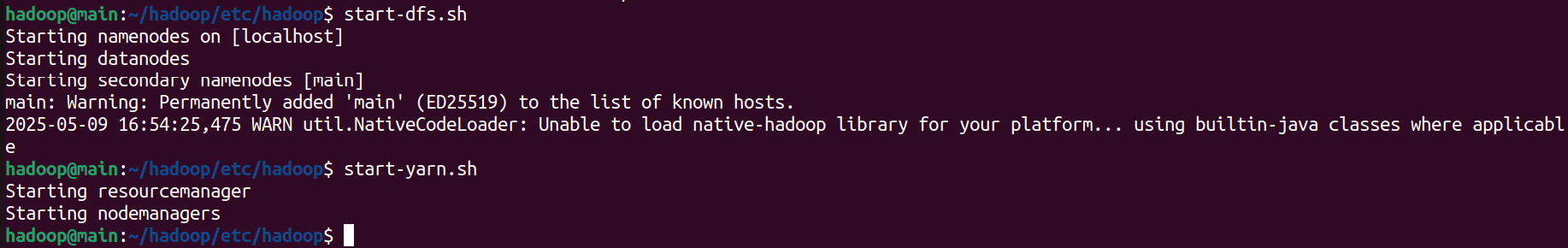
Create a directory and copy Hadoop’s XML files:

* ***mkdir -p ~/input***
* ***cp $HADOOP\_HOME/etc/hadoop/\*.xml ~/input/***

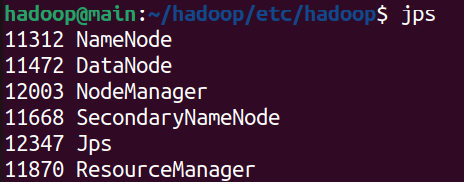
**2. Start Hadoop Services**

Start the HDFS and YARN daemons:

* ***start-dfs.sh***
* ***start-yarn.sh***



Verify all processes with jps.

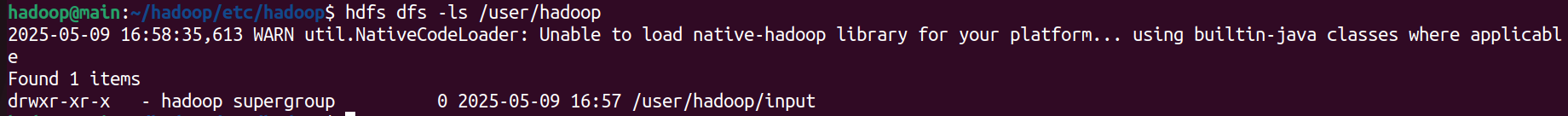


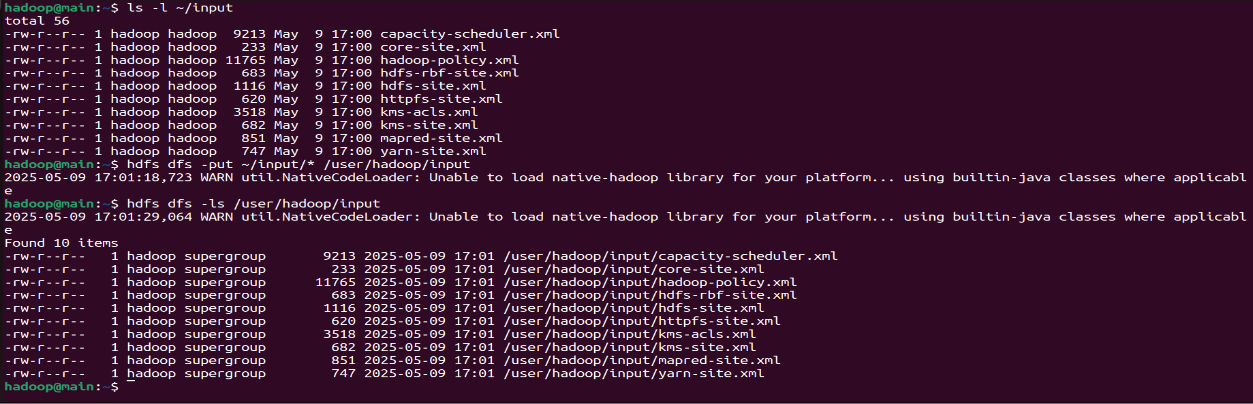
**3. Create HDFS Directories and Upload Files**

* ***hdfs dfs -mkdir -p /user/hadoop/input***
* ***hdfs dfs -put ~/input/\*.xml /user/hadoop/input/***

Confirm upload:

* ***hdfs dfs -ls /user/hadoop/input***





**4. Configure mapred-site.xml (Critical for YARN)**

Edit $HADOOP\_HOME/etc/hadoop/mapred-site.xml to include:

***<configuration>***

***<property>***

***<name>mapreduce.framework.name</name>***

***<value>yarn</value>***

***</property>***

***<property>***

***<name>yarn.app.mapreduce.am.env</name>***

***<value>HADOOP\_MAPRED\_HOME=/home/hadoop/hadoop</value>***

***</property>***

***<property>***

***<name>mapreduce.map.env</name>***

***<value>HADOOP\_MAPRED\_HOME=/home/hadoop/hadoop</value>***

***</property>***

***<property>***

***<name>mapreduce.reduce.env</name>***

***<value>HADOOP\_MAPRED\_HOME=/home/hadoop/hadoop</value>***

***</property>***

***</configuration>***

Replace /home/hadoop/hadoop with your actual Hadoop install path.

Restart Hadoop services to apply changes:

* stop-yarn.sh
* stop-dfs.sh
* start-dfs.sh
* start-yarn.sh

**5. Run the WordCount MapReduce Job**

Execute the job:

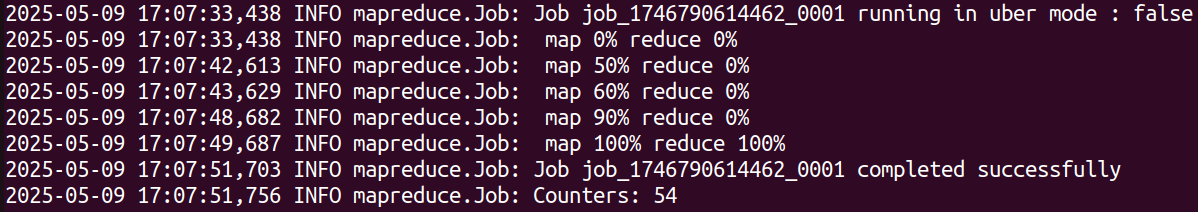
***hadoop jar \***

***$HADOOP\_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar \***

***wordcount \***

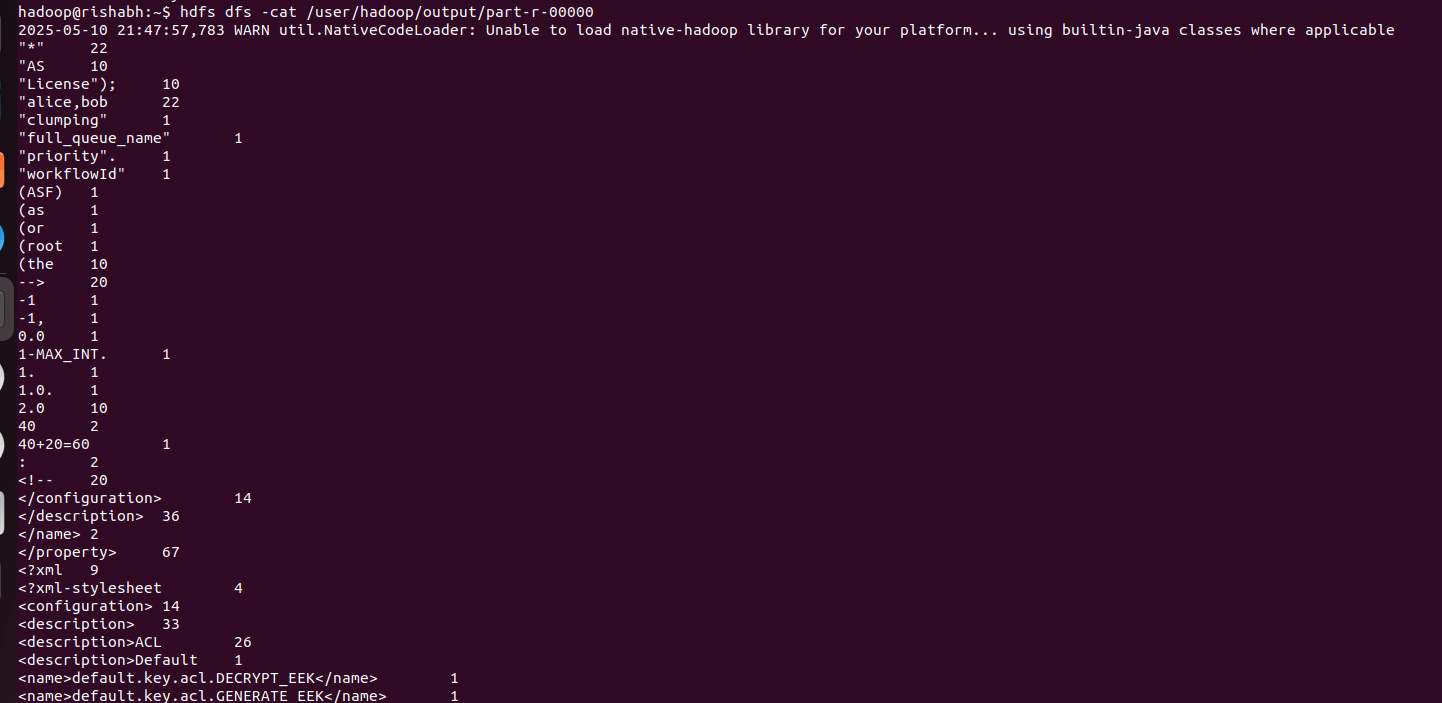
***/user/hadoop/input \***

***/user/hadoop/output***



**6. View Output**

***hdfs dfs -cat /user/hadoop/output/part-r-00000***

**Summary**

1. Configured Hadoop in pseudo-distributed mode on a single Ubuntu VM.
2. Prepared input data from Hadoop’s own configuration files.
3. Uploaded data to HDFS and ran the WordCount job using Hadoop’s built-in example JAR.
4. Ensured proper YARN and MapReduce configuration, resolving the MRAppMaster class error by correctly setting HADOOP\_MAPRED\_HOME.
5. Successfully verified the output through HDFS CLI.

This process validates that the pseudo-distributed Hadoop setup is correctly installed and capable of running MapReduce jobs end-to-end using YARN.