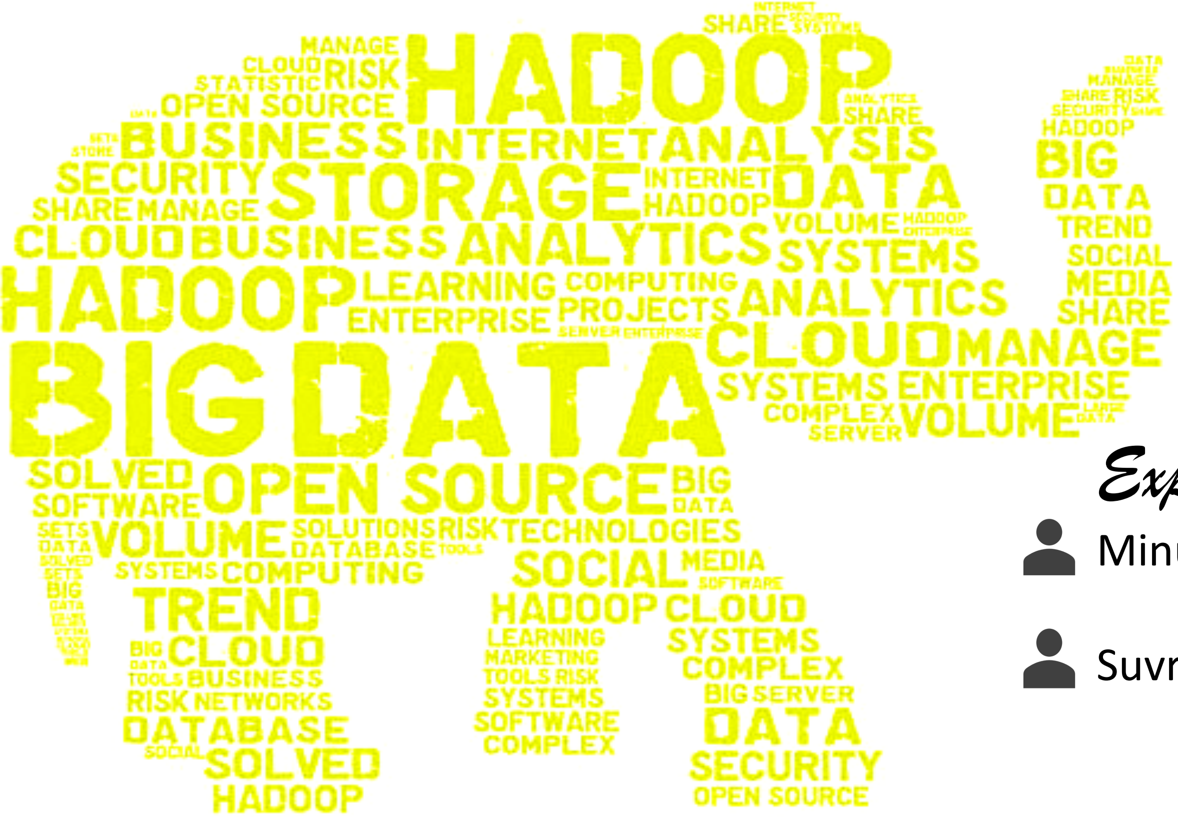
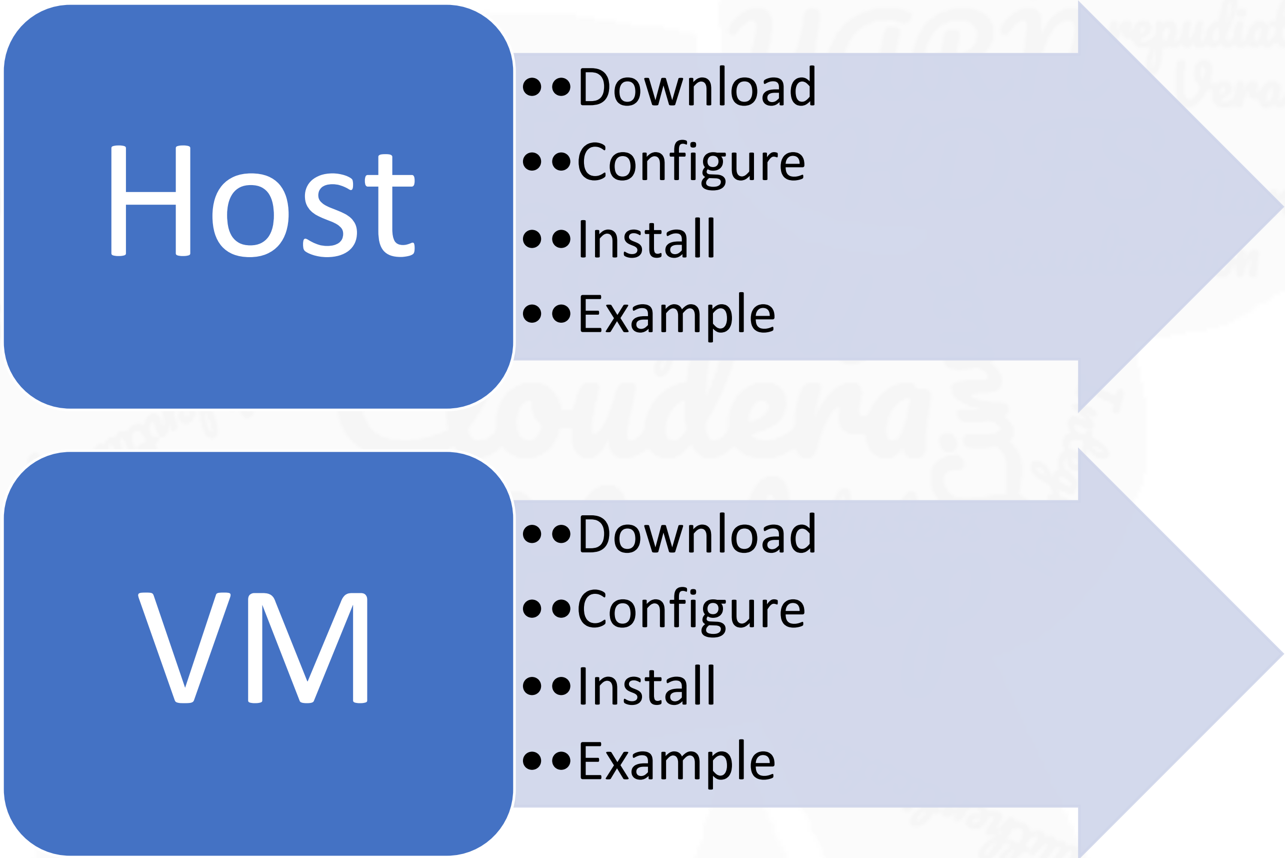
How To *Download*, *Configure* and *Run MapReduce* Program in cloudera VM?


INTERNET
SHARE SECURITY SYSTEMS
MANAGE
CLOUD RISK
STATISTIC
HADOOP
OPEN SOURCE
ANALYTICS SHARE
BUSINESS INTERNET ANALYSIS
SECURITY STORAGE INTERNET HADOOP DATA
SHARE MANAGE
CLOUD BUSINESS ANALYTICS VOLUME HADOOP ENTERPRISE SYSTEMS
HADOOP LEARNING COMPUTING ENTERPRISE PROJECTS ANALYTICS
SERVER ENTERPRISE
BIG DATA CLOUD MANAGE
SYSTEMS ENTERPRISE
COMPLEX SERVER VOLUME LARGE DATA
DATA MANAGE SHARE RISK SECURITY HADOOP BIG DATA TREND SOCIAL MEDIA SHARE
SOLVED SOFTWARE OPEN SOURCE BIG DATA
SETS DATA SOLVED SETS BIG DATA VOLUME SOLUTIONS RISK TECHNOLOGIES DATABASE TOOLS
SYSTEMS COMPUTING SOCIAL MEDIA SOFTWARE
TREND HADOOP CLOUD
BIG DATA CLOUD LEARNING MARKETING SYSTEMS
TOOLS BUSINESS TOOLS RISK COMPLEX
RISK NETWORKS SYSTEMS BIG SERVER
DATABASE SOFTWARE DATA
SOCIAL SOLVED COMPLEX SECURITY
HADOOP OPEN SOURCE

Expounded by:

- Minu Sunny (*10106288б*)
- Suvrojeet Kumar Ghosh (*8635364*)

Methods



Host

••Download

••Configure

••Install

••Example

VM

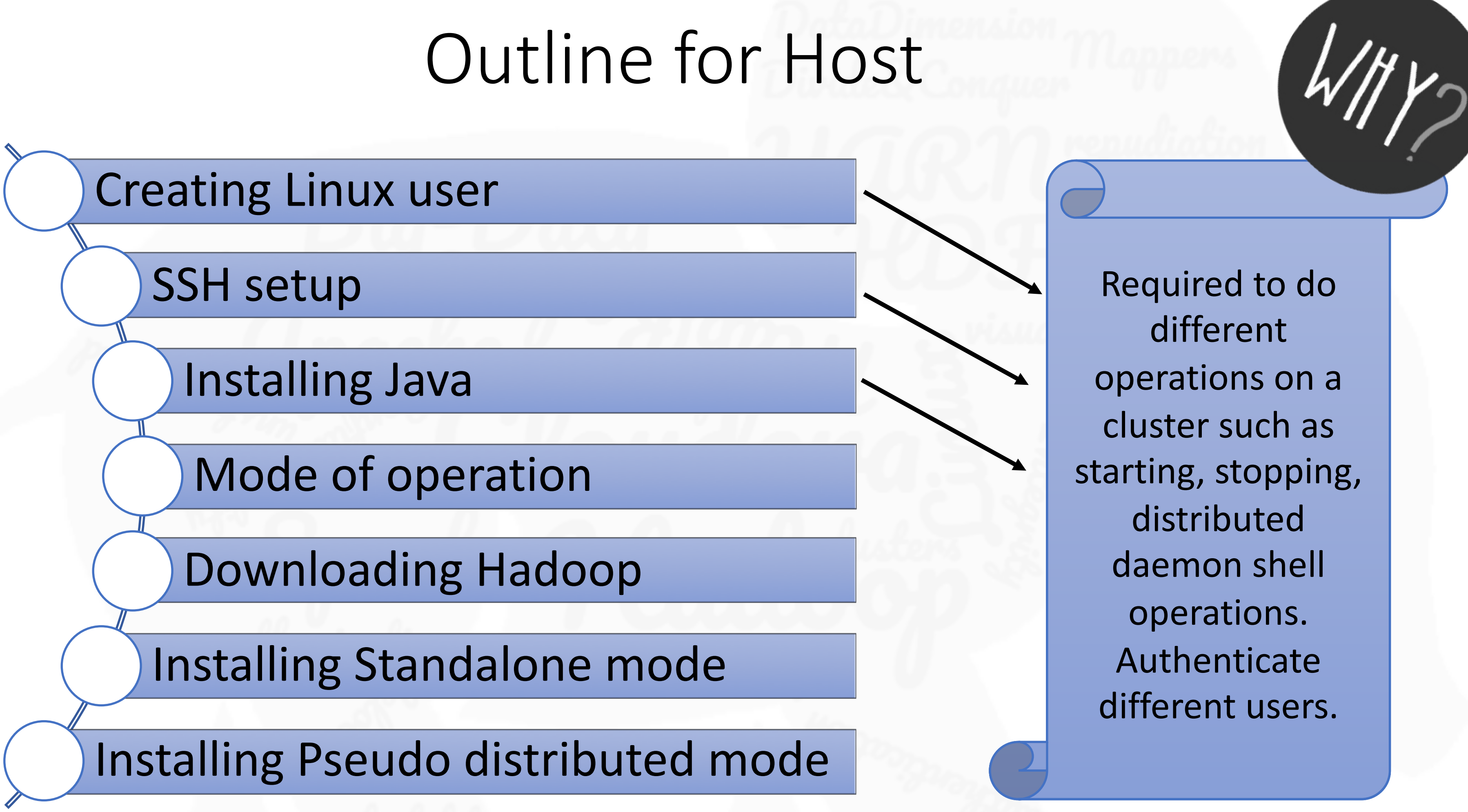
••Download

••Configure

••Install

••Example

Outline for Host

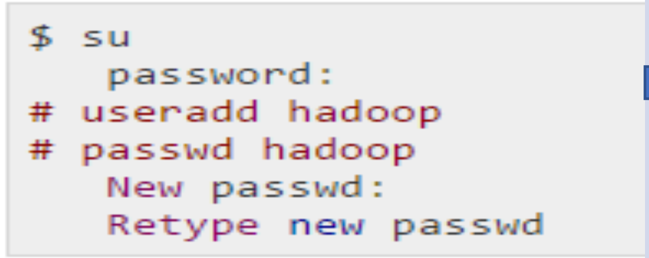


- Creating Linux user
- SSH setup
- Installing Java
- Mode of operation
- Downloading Hadoop
- Installing Standalone mode
- Installing Pseudo distributed mode

Required to do different operations on a cluster such as starting, stopping, distributed daemon shell operations. Authenticate different users.

Creating Linux User

Command (&/ description)



```
$ su
    password:
# useradd hadoop
# passwd hadoop
    New passwd:
    Retype new passwd
```

- adduser hadoop
- Perl script which creates all home directories, etc automatically

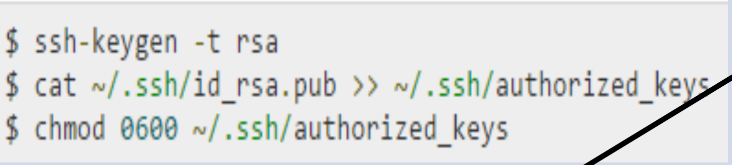
Problem faced

- Didn’t create home directory, Had to manually create directory

SSH Setup

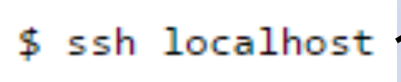
Command (&/ description)

- Generating keys using rsa

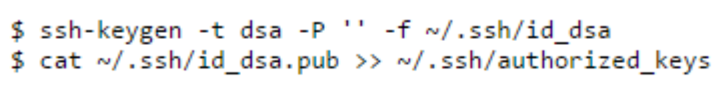


```
$ ssh-keygen -t rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
```

- To test



```
$ ssh localhost
```



```
$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

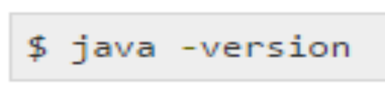
Problem faced (&/ Solution)

- Connection refused at port 22
- Solution tried is different algo for encryption. – but didn't work.

✓ Finally figured out I had created it for a different user. I had to create it for user "hadoop". Also didn't "sudo service start ssh"

Installing Java

Check already installed or not?

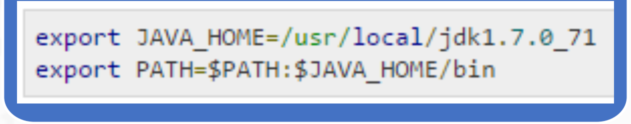


```
$ java -version
```

NO

Install Java

Set up PATH and JAVA_HOME variables in ~/.bashrc



```
export JAVA_HOME=/usr/local/jdk1.7.0_71
export PATH=$PATH:$JAVA_HOME/bin
```

YES

Modes of operations

1. **Local/Standalone Mode** : After downloading Hadoop in your system, by default, it is configured in a standalone mode and can be run as a single java process.

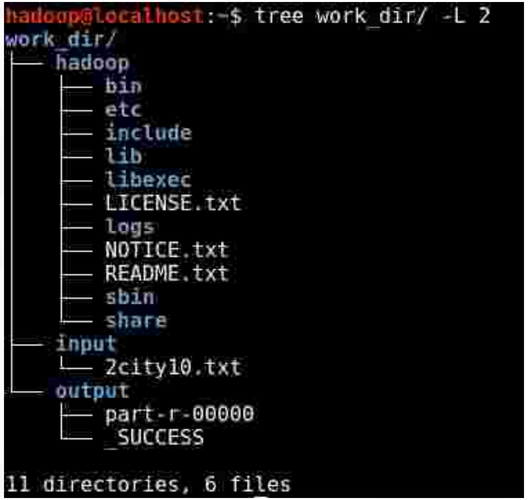
2. **Pseudo Distributed Mode** : It is a distributed simulation on single machine. Each Hadoop daemon such as hdfs, yarn, MapReduce etc., will run as a separate java process. This mode is useful for development.

3. **Fully Distributed Mode** : This mode is fully distributed with minimum two or more machines as a cluster. We will come across this mode in detail in the coming chapters. Used in “productions”.

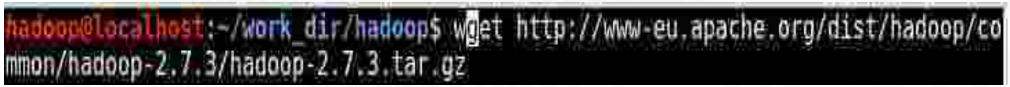
- We will see examples with mode 1 and 2 in host computer. Later In Cloudera we will see example only in mode 1 and further in the end we will see examples in Redhat using mode 2 only.

Downloading Hadoop (*This step was smooth)

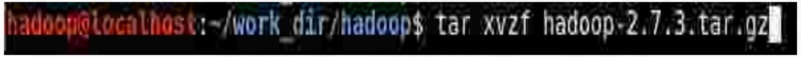
- At this point I was a working as “hadoop” user
- And my working directory was work_dir.
- This working directory was also Hadoop installation



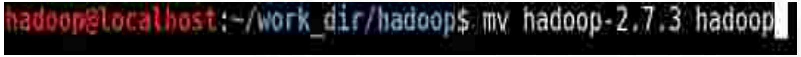
```
hadoop@localhost:~$ tree work_dir/ -L 2
work_dir/
├── hadoop
│   ├── bin
│   ├── etc
│   ├── include
│   ├── lib
│   ├── libexec
│   ├── LICENSE.txt
│   ├── logs
│   ├── NOTICE.txt
│   ├── README.txt
│   ├── sbin
│   └── share
├── input
│   └── 2city10.txt
└── output
    ├── part-r-00000
    └── _SUCCESS

11 directories, 6 files
```



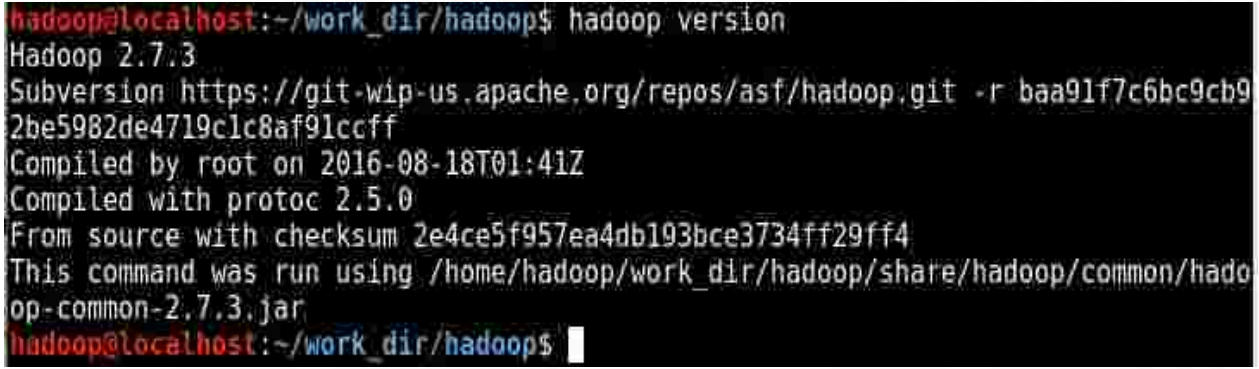
```
hadoop@localhost:~/work_dir/hadoop$ wget http://www-eu.apache.org/dist/hadoop/co
mmon/hadoop-2.7.3/hadoop-2.7.3.tar.gz
```



```
hadoop@localhost:~/work_dir/hadoop$ tar xvzf hadoop-2.7.3.tar.gz
```



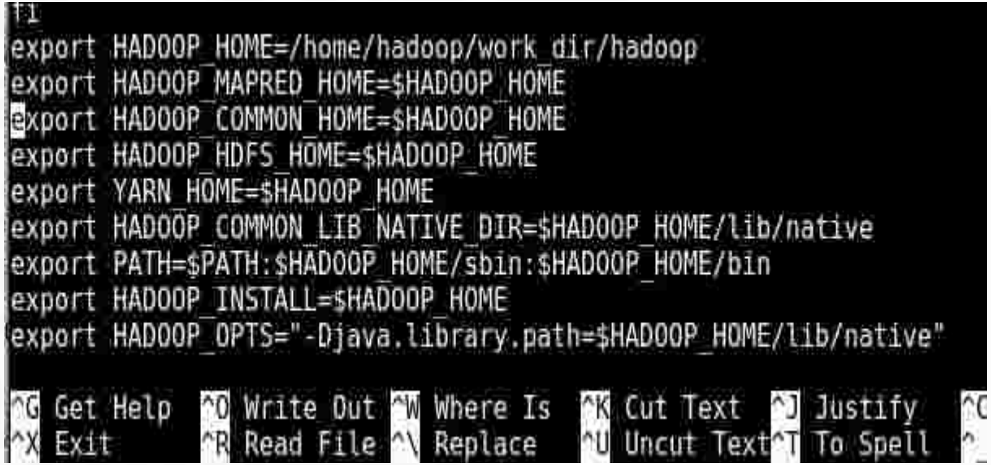
```
hadoop@localhost:~/work_dir/hadoop$ mv hadoop-2.7.3 hadoop
```



```
hadoop@localhost:~/work_dir/hadoop$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb9
2be5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/hadoop/work_dir/hadoop/share/hadoop/common/hado
op-common-2.7.3.jar
hadoop@localhost:~/work_dir/hadoop$
```

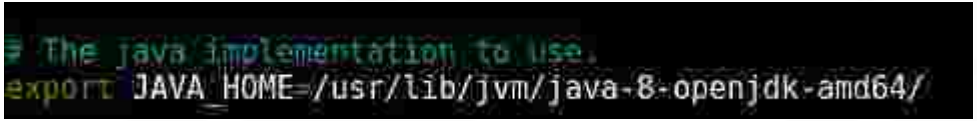
Installing Hadoop (*standalone mode)

- Standalone mode doesn’t require any install apart from few configuration below: ~/.bashrc



```
fi
export HADOOP_HOME=/home/hadoop/work_dir/hadoop
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=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

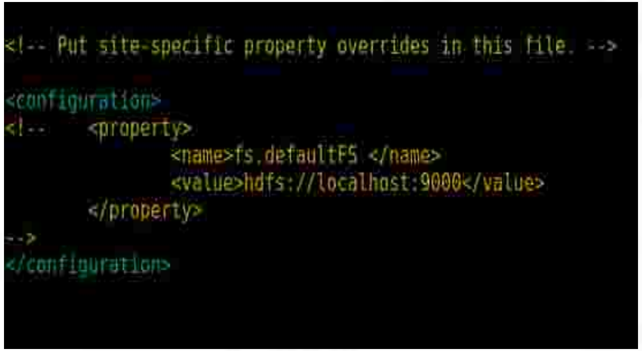
- hadoop-env.sh (mandatory)



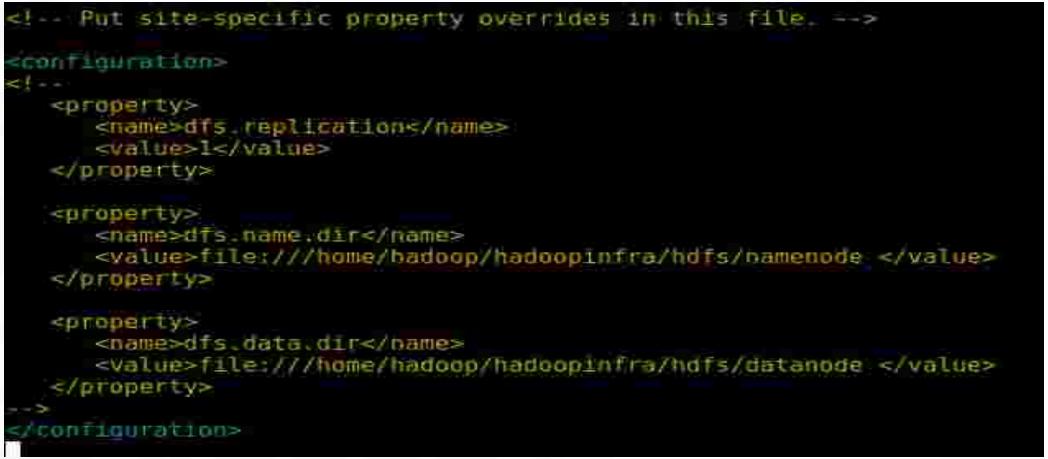
```
# The java implementation to use.
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
```

Installing Hadoop (*pseudomode)

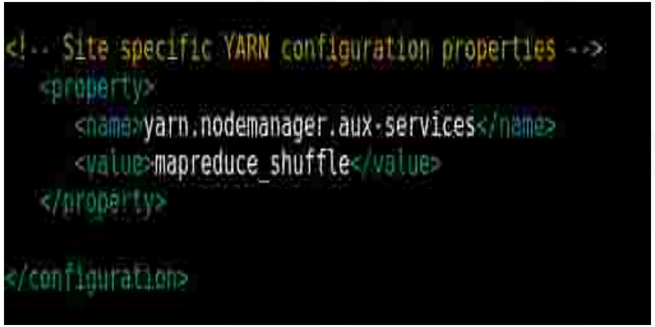
- Pseudo distributed mode …. (Lots of work…). Mainly editing four config files of Hadoop installation **core-site.xml, hdfs-site.xml, yarn-site.xml, mapred-site.xml**



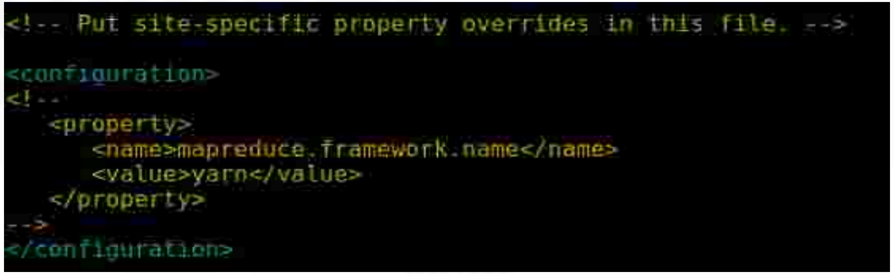
```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<!--     <property>
                <name>fs.defaultFS </name>
                <value>hdfs://localhost:9000</value>
         </property>
-->
</configuration>
```



```
<!-- Put site-specific property overrides in this file. -->

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

   <property>
      <name>dfs.name.dir</name>
      <value>file:///home/hadoop/hadoopinfra/hdfs/namenode </value>
   </property>

   <property>
      <name>dfs.data.dir</name>
      <value>file:///home/hadoop/hadoopinfra/hdfs/datanode </value>
   </property>
-->
</configuration>
```

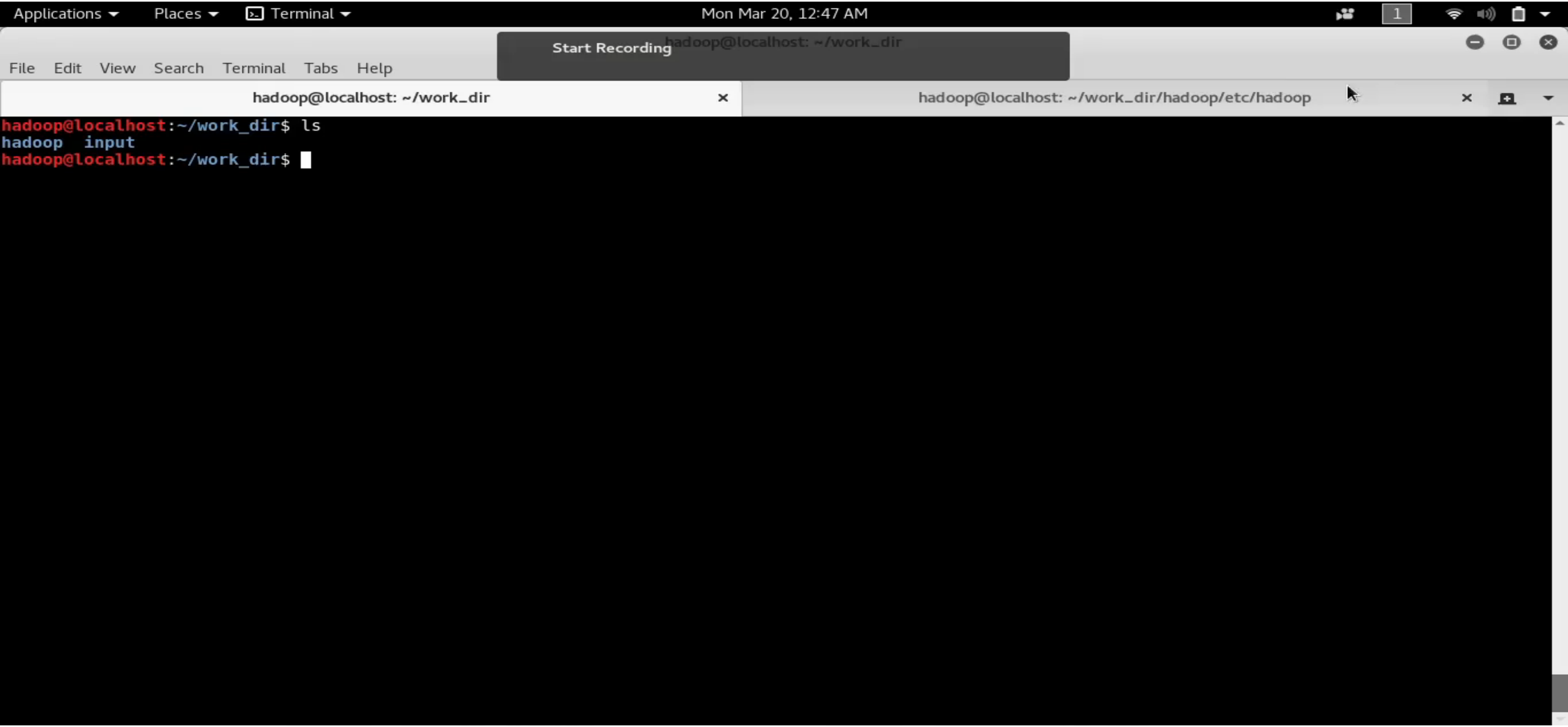


```
<!-- Site specific YARN configuration properties -->
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>

</configuration>
```



```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<!--
  <property>
     <name>mapreduce.framework.name</name>
     <value>yarn</value>
  </property>
-->
</configuration>
```

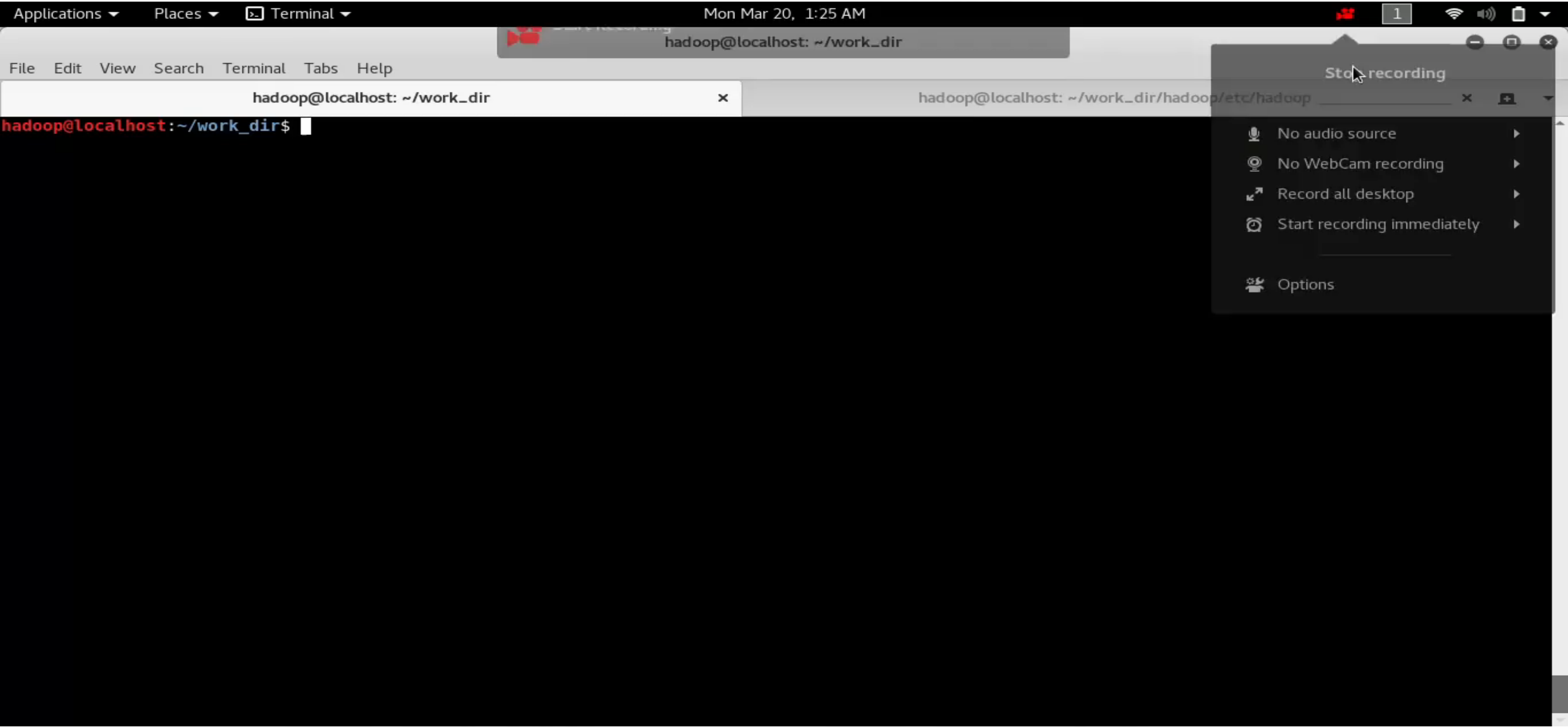
Running word-count Program (*standalone mode)



Applications
Places
Terminal
Mon Mar 20, 12:47 AM
Start Recording
hadoop@localhost: ~/work_dir
File
Edit
View
Search
Terminal
Tabs
Help
hadoop@localhost: ~/work_dir
hadoop@localhost: ~/work_dir/hadoop/etc/hadoop

```
hadoop@localhost:~/work_dir$ ls
hadoop  input
hadoop@localhost:~/work_dir$ 
```

Running word-count Program (*pseudo distributed mode)



```
hadoop@localhost:~/work_dir$ 
```

Outline for Cloudera VM

- ➤Virtual machine (VM).
- About Cloudera
- Description Of Cloudera VM
- Download Cloudera VM.
- How to configure and warmup of the VM.
- Action Time : Examples
- Problem faced and alternative chosen.

Virtual Machine (VM)

- Runs in a host computer as a normal virtual computer and create it’s own workspace by sharing the resources of host computer.
- For example, we can run a Linux OS in a Windows OS platform by installing a VM.
- To run the virtual machine VMware, KVM or Virtualbox can be used.

About Cloudera

- Cloudera Inc. is a US based company that provides Apache Hadoop based software, support and services, and training to business customers.
- Architect Doug Cutting, also a former chairman of the Apache Software Foundation, he wrote the initial Hadoop software in 2004, Joined Cloudera in 2009.
- There Products are **Cloudera Manager, Cloudera Navigator, Gazzang, Cloudera Navigator Optimizer, Impala** can be found in there VM, known as “QuickStart VM” based on CentOS distribution.
- The VM is pre - built Hadoop stack and applications related to it.

Description of Cloudera VM

Requirements

- •64 bit host OS
- •Min. 4GB RAM.
- •Updated Version of Virtualization software.
- •**Oracle VM Virtual Box** from the link. https://www.virtualbox.org/wiki/Downloads

Download Zip

- •OVF (Open Virtualization Format) file.
- •VMDK (Virtual Machine Disk) file.
- •*VM is freely distributed

Contains

- •HDFS.
- •MapReduce Framework.
- •Supporting applications from Apache foundation.
- •Pre-Built with Big Data ecosystem consisting of Hive, Impala, HBase , Sqoop.

Download Cloudera VM

- Link to download Cloudera VM:

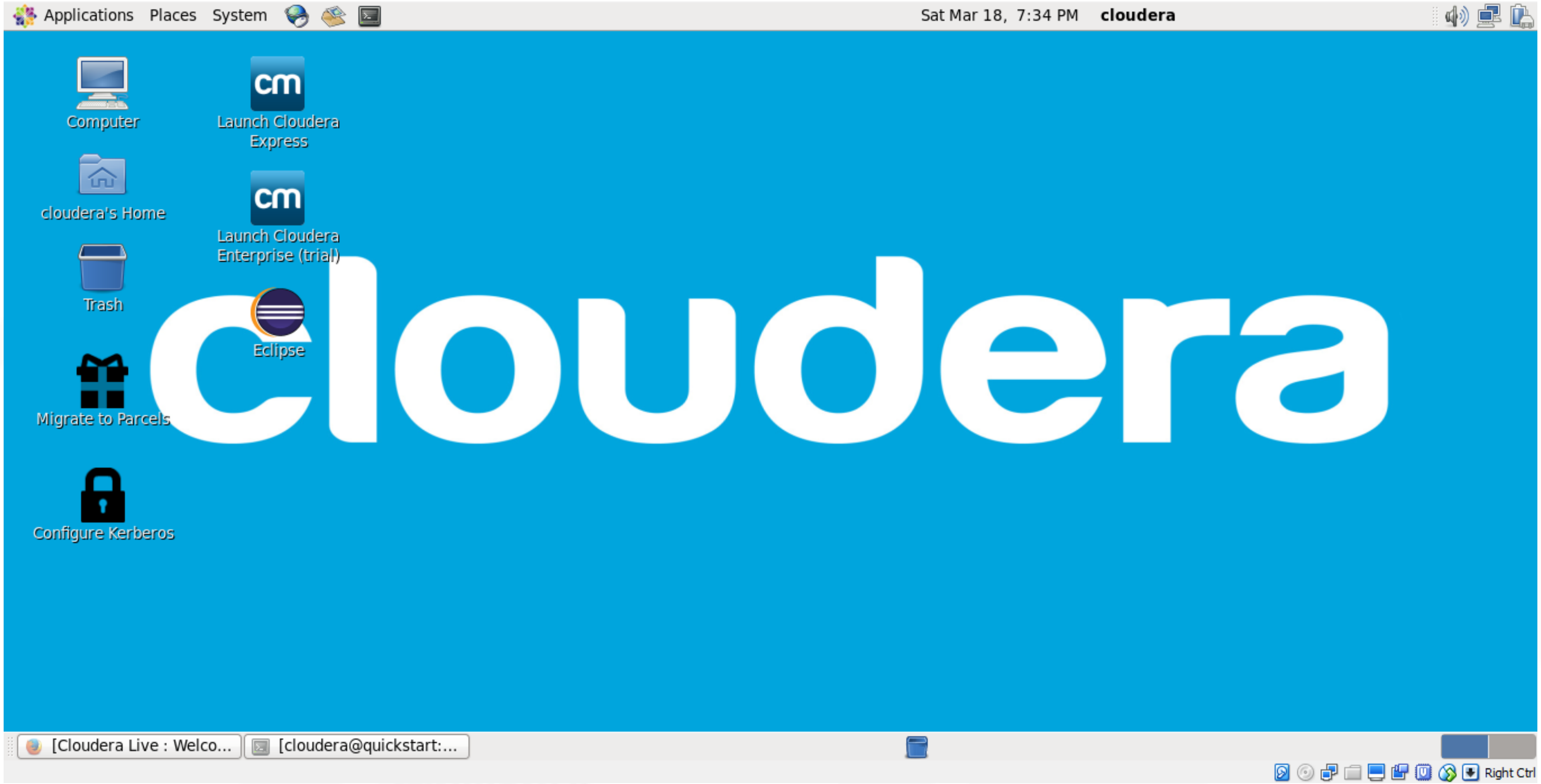
  https://www.cloudera.com/downloads/quickstart_vms/5-8.html

- Select the version as **QuickStarts for CDH 5.8** and select platform as **Virtual Box.**
- **Sign In** or Complete **Product interest form**.
- **Download** the ZIP file.
- **Extract** the ZIP file.

How to Configure a VM

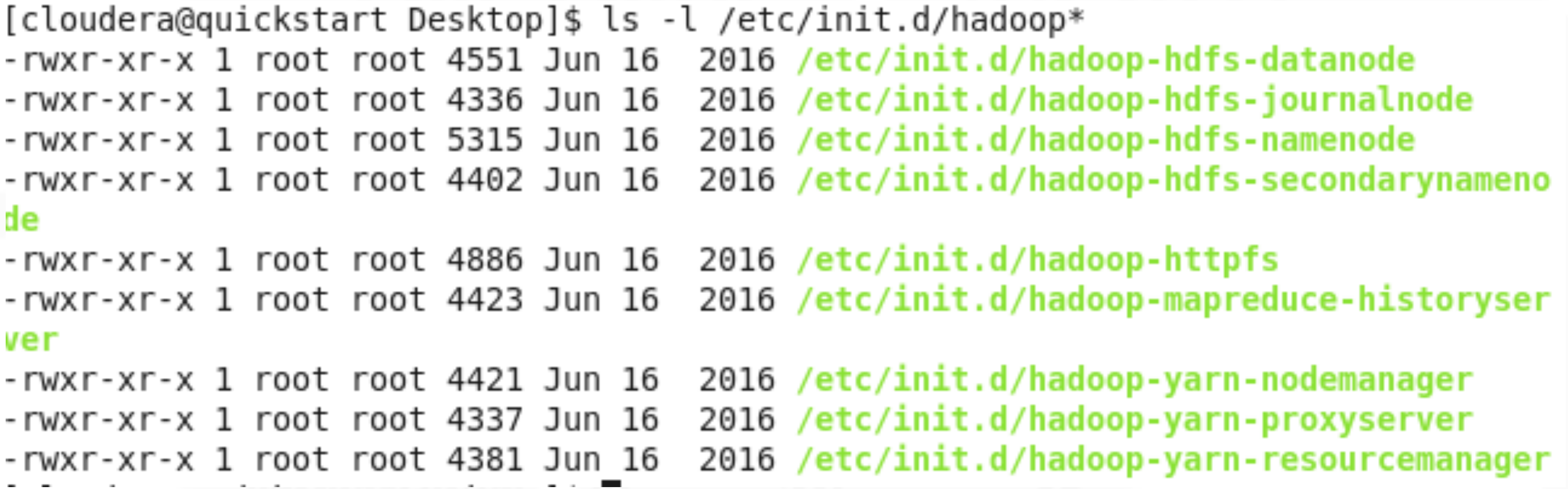
- Open **Oracle VM Virtual Box Manager.**
- Click on **New** to create new virtual box .
- Give name for new virtual machine and select type as **Linux** and version according to VM available.
- Select Memory Size as 4GB and click Next.
- Select Hard Drive for new VirualBox . Select **Use an existing virtual hard drive file** option.
- Click **Start**

After the VM Warmed up



Configurations *good to know

- All the “downloading and configurations” for hadoop seen in the “host computer installation section” is abstracted in the below listed scripts which is located in “init.d” folder



```
[cloudera@quickstart Desktop]$ ls -l /etc/init.d/hadoop*
-rwxr-xr-x 1 root root 4551 Jun 16  2016 /etc/init.d/hadoop-hdfs-datanode
-rwxr-xr-x 1 root root 4336 Jun 16  2016 /etc/init.d/hadoop-hdfs-journalnode
-rwxr-xr-x 1 root root 5315 Jun 16  2016 /etc/init.d/hadoop-hdfs-namenode
-rwxr-xr-x 1 root root 4402 Jun 16  2016 /etc/init.d/hadoop-hdfs-secondarynameno
de
-rwxr-xr-x 1 root root 4886 Jun 16  2016 /etc/init.d/hadoop-httpfs
-rwxr-xr-x 1 root root 4423 Jun 16  2016 /etc/init.d/hadoop-mapreduce-historyser
ver
-rwxr-xr-x 1 root root 4421 Jun 16  2016 /etc/init.d/hadoop-yarn-nodemanager
-rwxr-xr-x 1 root root 4337 Jun 16  2016 /etc/init.d/hadoop-yarn-proxyserver
-rwxr-xr-x 1 root root 4381 Jun 16  2016 /etc/init.d/hadoop-yarn-resourcemanager
```

- These all are loaded as services in Linux which can be stopped using the command in this format “sudo service *select select* stop"

What you can expect in this VM

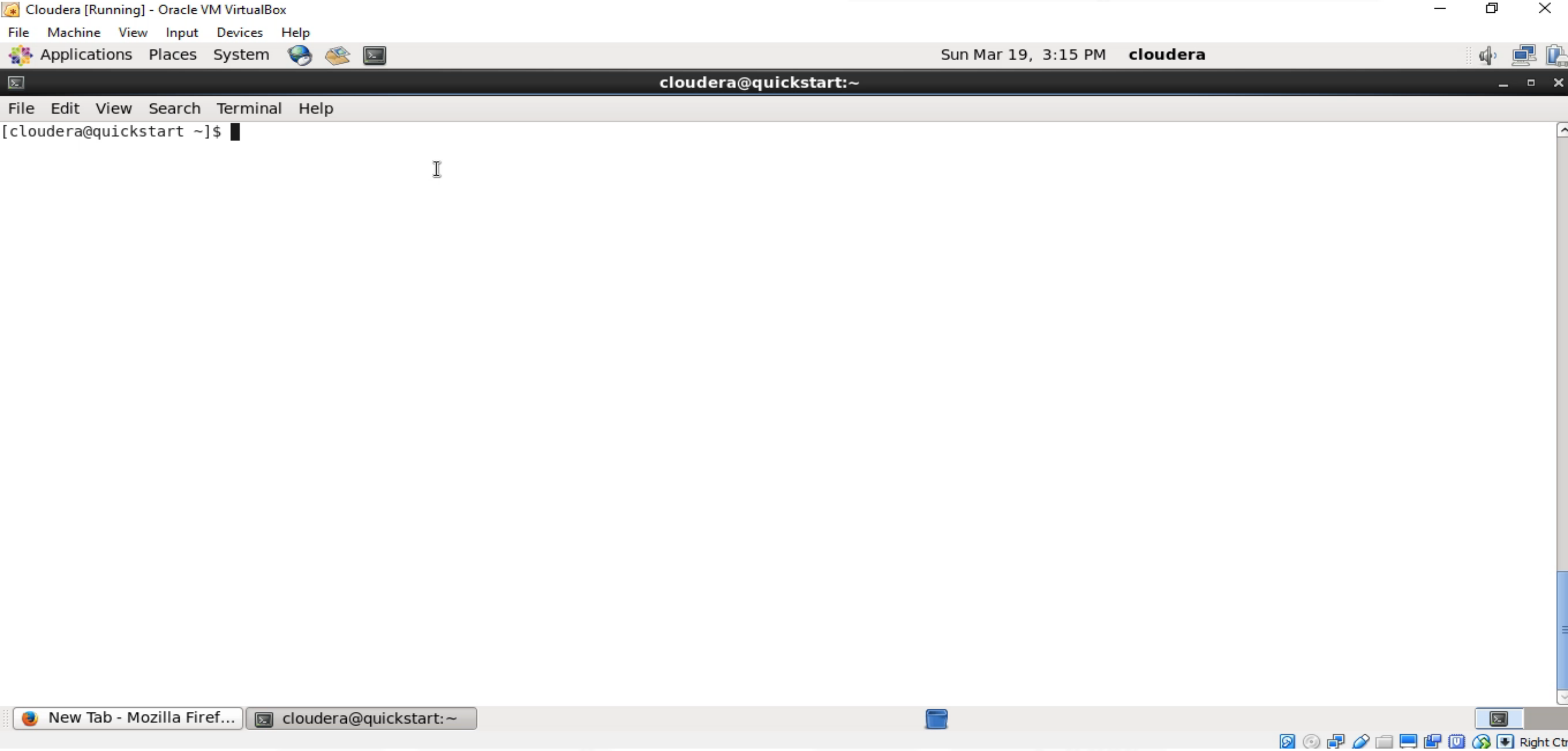
Interesting tutorials are given in Cloudera VM based on business scenarios and corresponding Hadoop solutions.

- Example of analyzing data of products interested by customers gives idea about the method to feed data from relational databases to HDFS .
- Processing the available data.
- Usage of Impala and construct the graph.
- Combining web access logs.
- Analytics using Spark.

Action time

(in Cloudera)

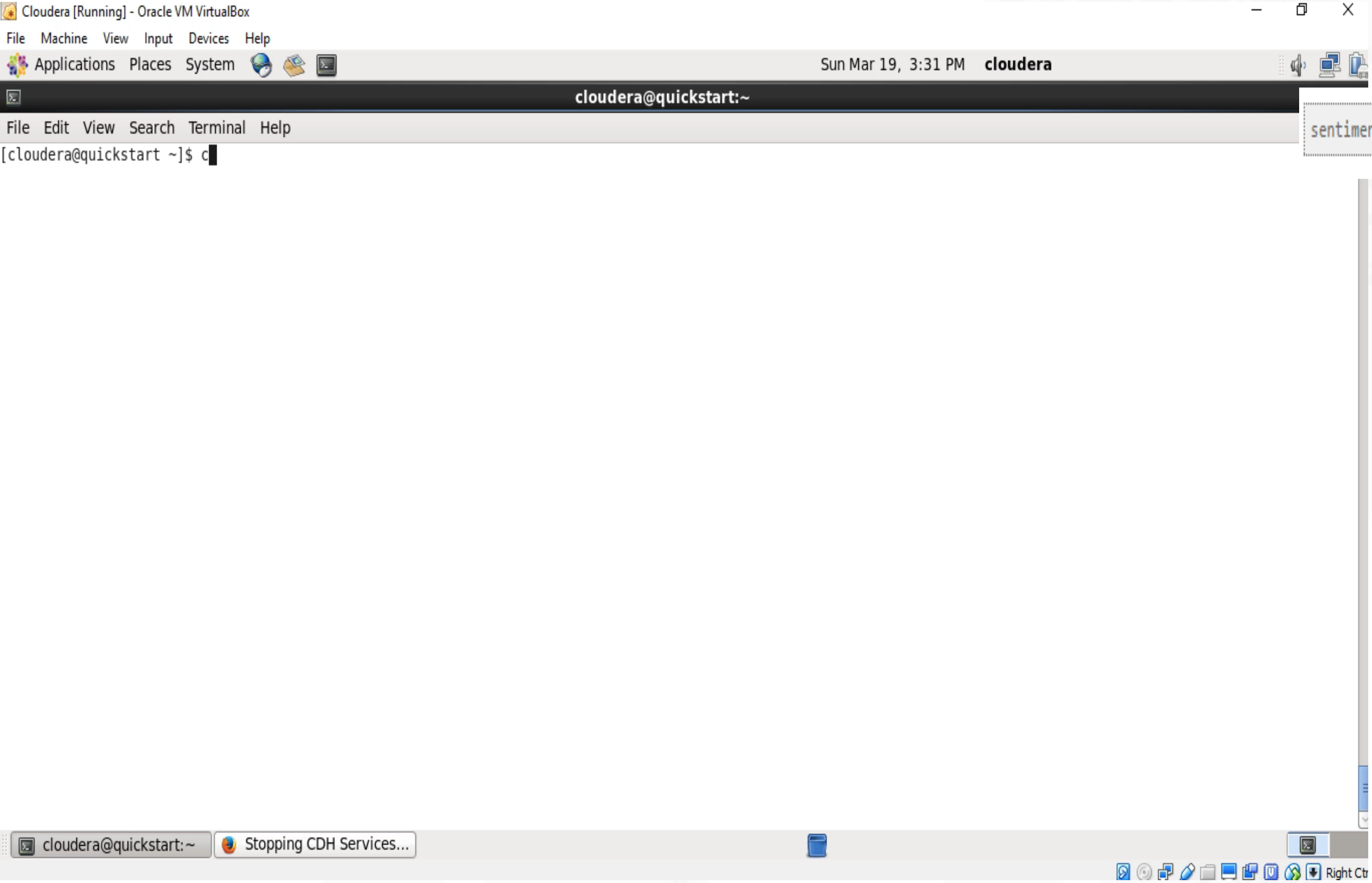
Example of Simple wordcount program.



Cloudera [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Sun Mar 19, 3:15 PM
cloudera
cloudera@quickstart:~
File Edit View Search Terminal Help
New Tab - Mozilla Firef...
cloudera@quickstart:~
Right Ctr

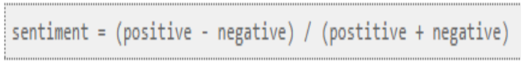
```
[cloudera@quickstart ~]$ 
```

Example of Sentiment Analysis Program


Cloudera [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Sun Mar 19, 3:31 PM cloudera
cloudera@quickstart:~
File Edit View Search Terminal Help
cloudera@quickstart:~
Stopping CDH Services...
Right Ctrl

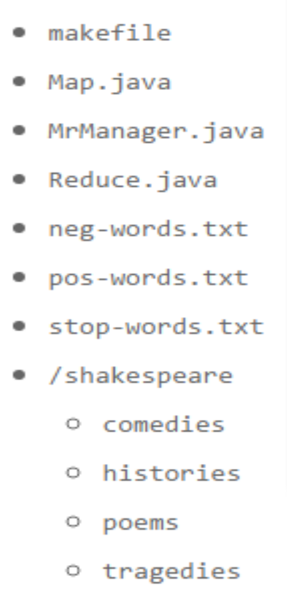
```
[cloudera@quickstart ~]$ c
```

Formula used



```
sentiment = (positive - negative) / (postitive + negative)
```

Directory Structure

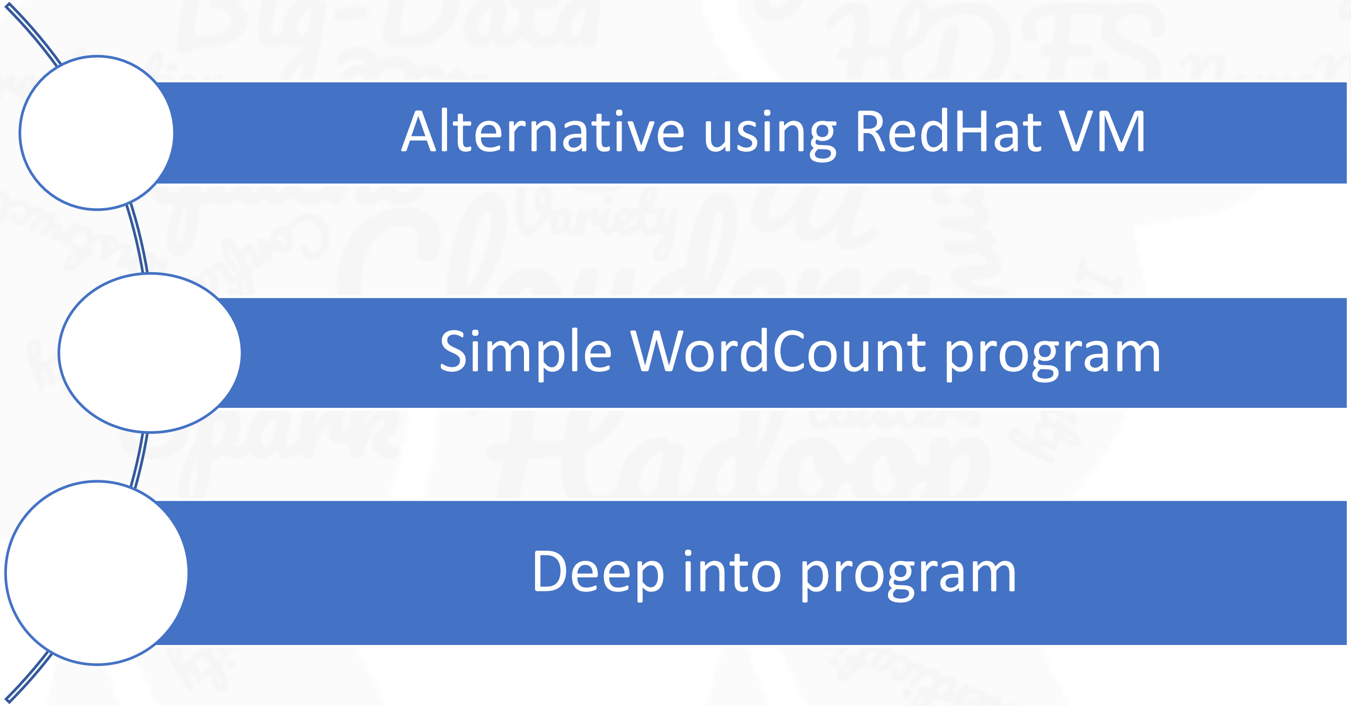


- makefile
- Map.java
- MrManager.java
- Reduce.java
- neg-words.txt
- pos-words.txt
- stop-words.txt
- /shakespeare
  - comedies
  - histories
  - poems
  - tragedies

Problem Faced and Alternative Chosen

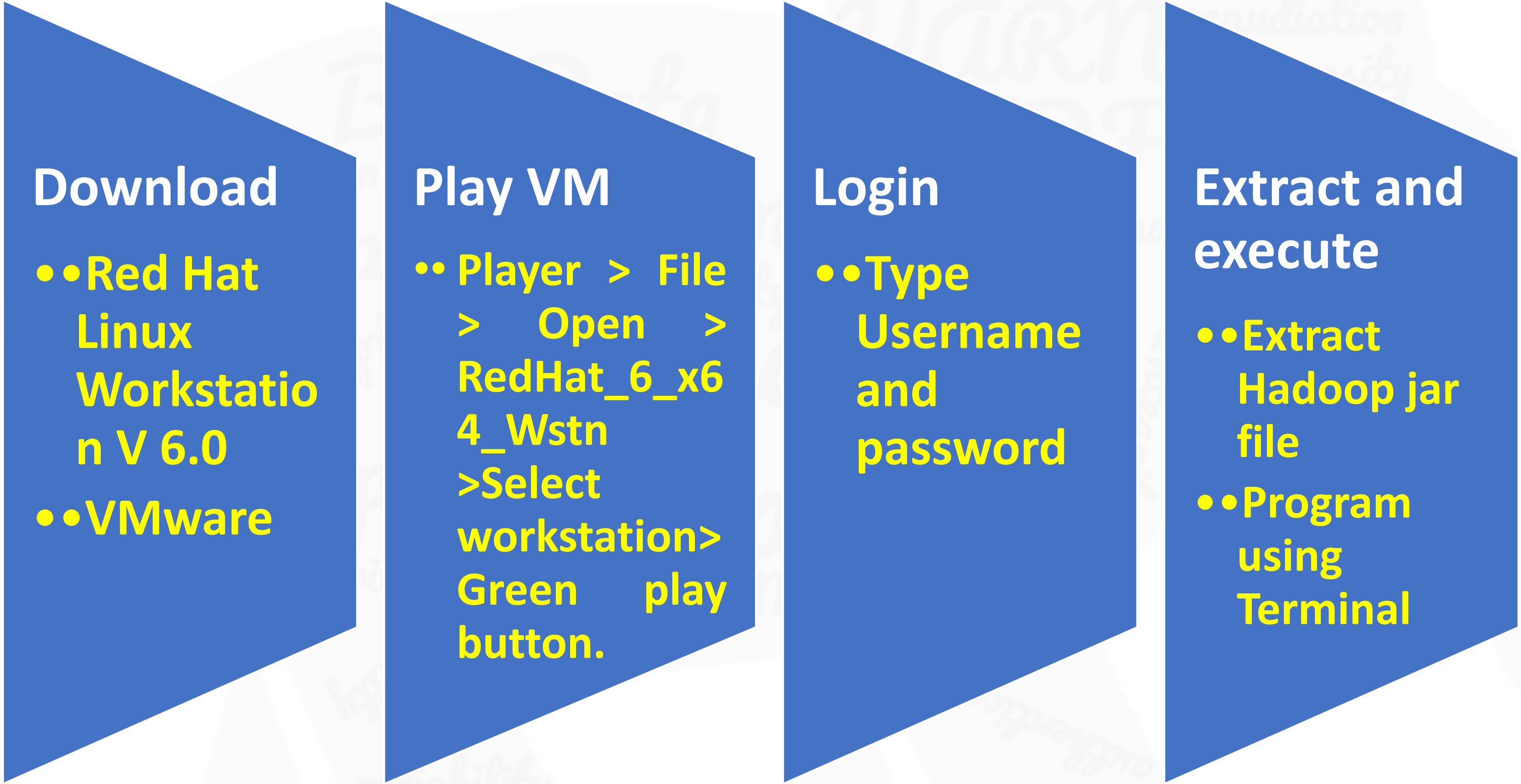
- Problem : Failed to compile JAVA source code.
- Alternative chosen: Tried Red Hat Linux Workstation v 6.0.

Outline for RedHat VM



- Alternative using RedHat VM
- Simple WordCount program
- Deep into program

Alternative using Red Hat Workstation



Download

- •Red Hat Linux Workstation V 6.0
- •VMware

Play VM

- • Player > File > Open > RedHat_6_x64_Wstn >Select workstation> Green play button.

Login

- •Type Username and password

Extract and execute

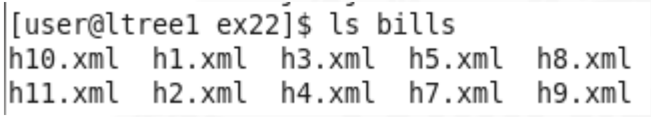
- •Extract Hadoop jar file
- •Program using Terminal

Simple Word Count Program

```
$ cd ex/ex22
$ ls
$ ls bills
$ start-dfs.sh
$ start-yarn.sh
$ jps
$ hadoop fs -mkdir -p ex22/bills
$ hadoop fs -put bills ex22
$ hadoop jar wordcount.jar wordcount ex22/bills ex22/word_frequency
$ hadoop fs -rm -r ex22/word_frequency
$ hadoop fs -ls ex22
$ mr- jobhistory -daemon.sh start historyserver
$ hadoop jar wordcount.jar ex22/bills ex22/word_frequency
$ ~/stop-hadoop.sh
$ exit
```

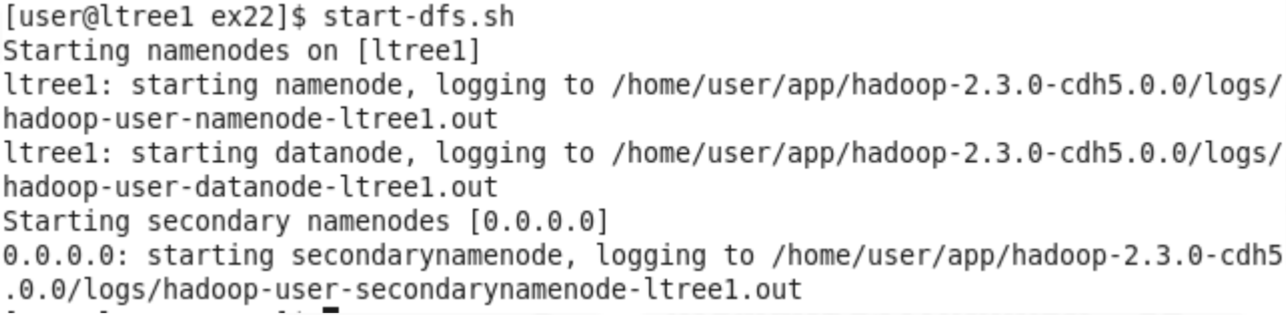
Deep into program

```
$ cd ex/ex22
$ ls
$ ls bills
```



```
[user@ltree1 ex22]$ ls bills
h10.xml  h1.xml  h3.xml  h5.xml  h8.xml
h11.xml  h2.xml  h4.xml  h7.xml  h9.xml
```

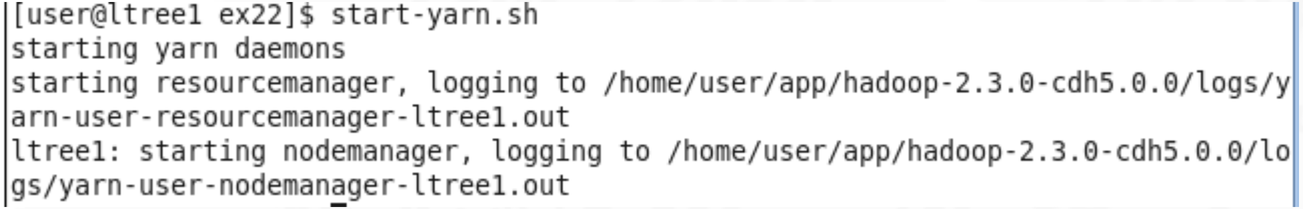
```
$ start-dfs.sh
```



```
[user@ltree1 ex22]$ start-dfs.sh
Starting namenodes on [ltree1]
ltree1: starting namenode, logging to /home/user/app/hadoop-2.3.0-cdh5.0.0/logs/
hadoop-user-namenode-ltree1.out
ltree1: starting datanode, logging to /home/user/app/hadoop-2.3.0-cdh5.0.0/logs/
hadoop-user-datanode-ltree1.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/user/app/hadoop-2.3.0-cdh5
.0.0/logs/hadoop-user-secondarynamenode-ltree1.out
```

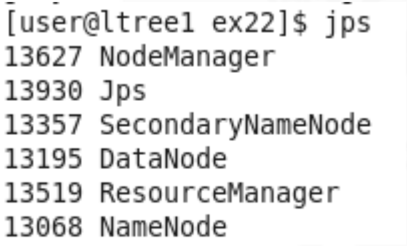
Continued….

$ start-yarn.sh



```
[user@ltree1 ex22]$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/user/app/hadoop-2.3.0-cdh5.0.0/logs/y
arn-user-resourcemanager-ltree1.out
ltree1: starting nodemanager, logging to /home/user/app/hadoop-2.3.0-cdh5.0.0/lo
gs/yarn-user-nodemanager-ltree1.out
```

$jps



```
[user@ltree1 ex22]$ jps
13627 NodeManager
13930 Jps
13357 SecondaryNameNode
13195 DataNode
13519 ResourceManager
13068 NameNode
```

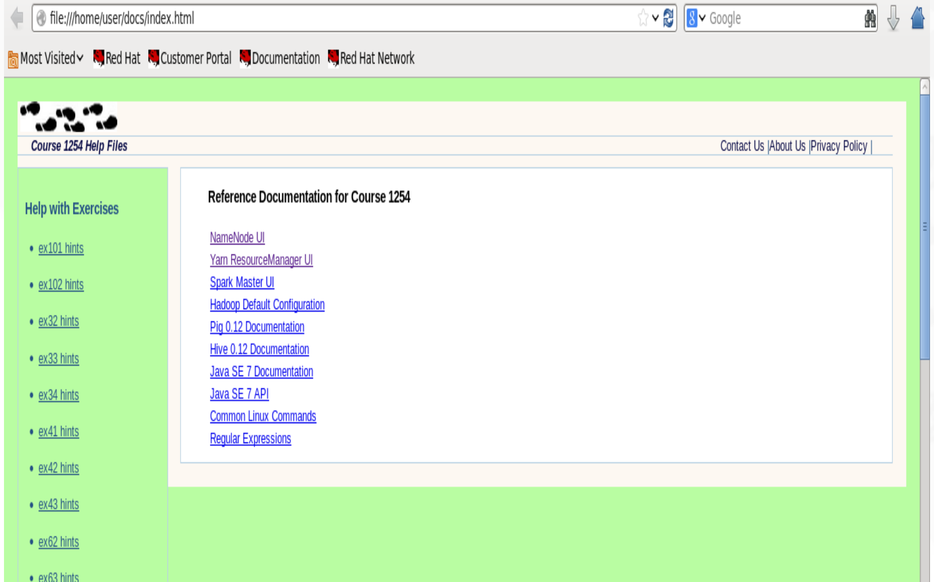
Continued….

```
$ hadoop fs -mkdir -p ex22/bills
```

Created directory called ex22 with a sub directory bills in HDFS.

```
$ hadoop fs -put bills ex22
```

Homepage>NamenodeUI>Go to directory>Type /user/user/ex22/bills.



Course 1254 Help Files

Contact Us |About Us |Privacy Policy |

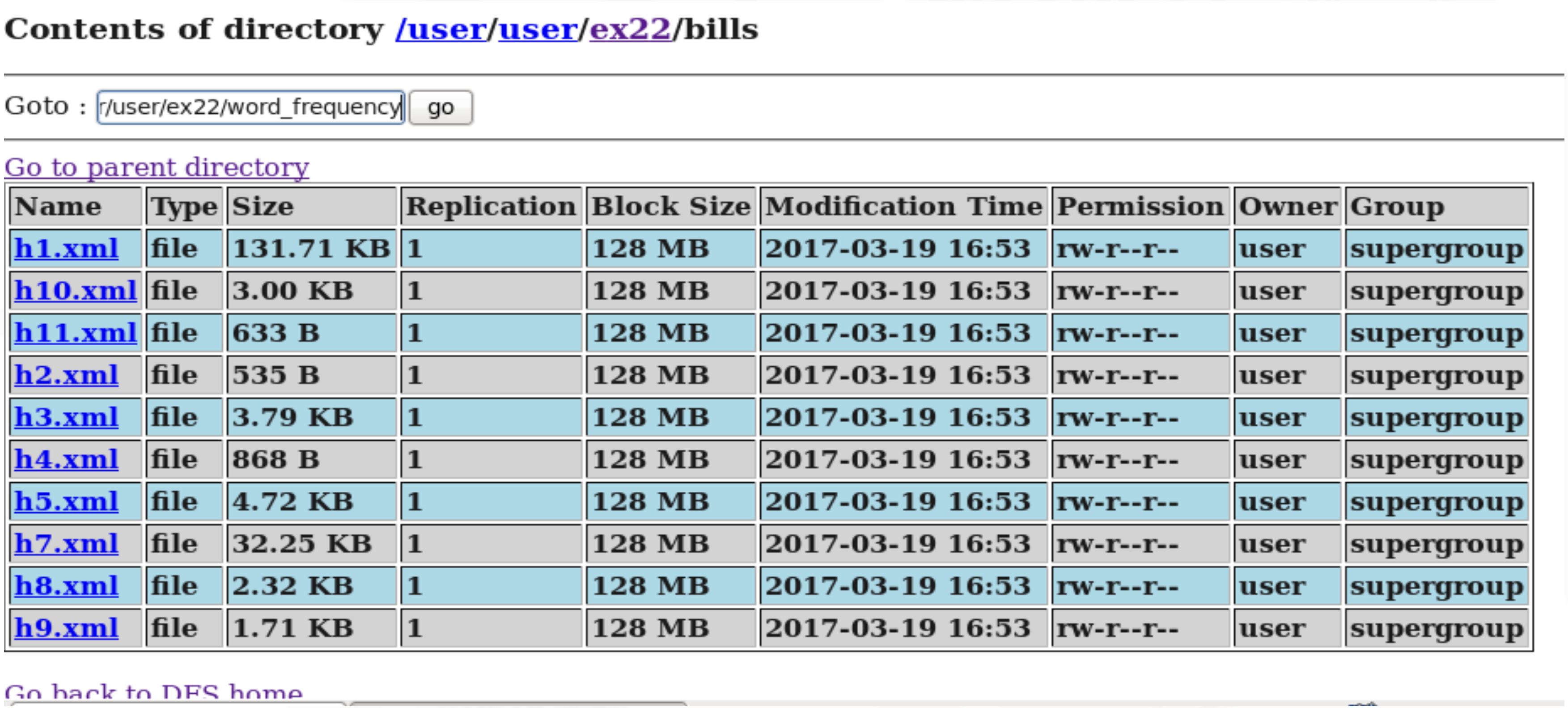
Help with Exercises

- ex101 hints
- ex102 hints
- ex32 hints
- ex33 hints
- ex34 hints
- ex41 hints
- ex42 hints
- ex43 hints
- ex62 hints
- ex63 hints

Reference Documentation for Course 1254

NameNode UI
Yarn ResourceManager UI
Spark Master UI
Hadoop Default Configuration
Pig 0.12 Documentation
Hive 0.12 Documentation
Java SE 7 Documentation
Java SE 7 API
Common Linux Commands
Regular Expressions

Deep into program



Contents of directory /user/user/ex22/bills

Go to parent directory

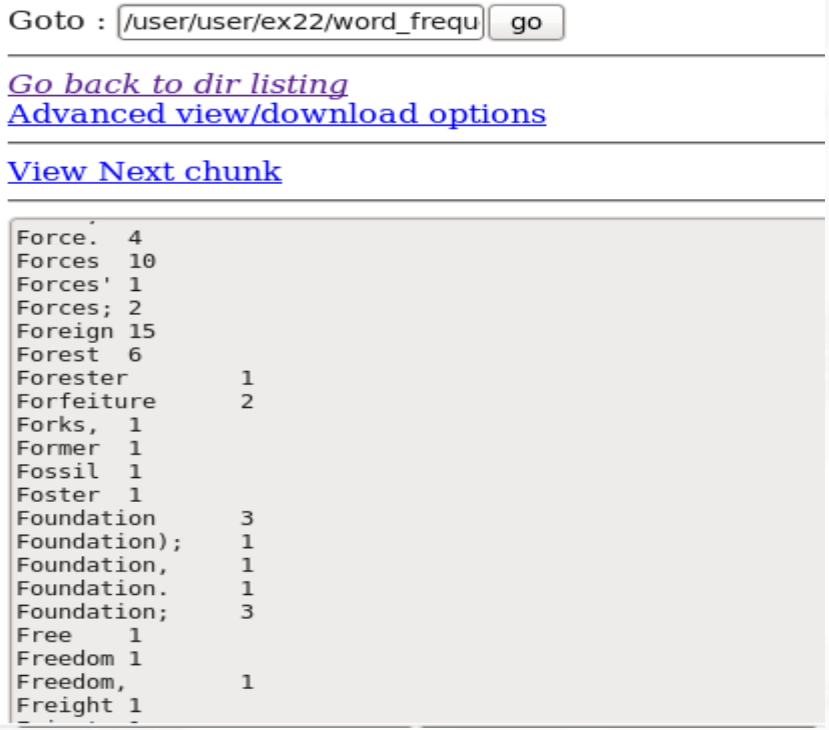
Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
h1.xml	file	131.71 KB	1	128 MB	2017-03-19 16:53	rw-r--r--	user	supergroup
h10.xml	file	3.00 KB	1	128 MB	2017-03-19 16:53	rw-r--r--	user	supergroup
h11.xml	file	633 B	1	128 MB	2017-03-19 16:53	rw-r--r--	user	supergroup
h2.xml	file	535 B	1	128 MB	2017-03-19 16:53	rw-r--r--	user	supergroup
h3.xml	file	3.79 KB	1	128 MB	2017-03-19 16:53	rw-r--r--	user	supergroup
h4.xml	file	868 B	1	128 MB	2017-03-19 16:53	rw-r--r--	user	supergroup
h5.xml	file	4.72 KB	1	128 MB	2017-03-19 16:53	rw-r--r--	user	supergroup
h7.xml	file	32.25 KB	1	128 MB	2017-03-19 16:53	rw-r--r--	user	supergroup
h8.xml	file	2.32 KB	1	128 MB	2017-03-19 16:53	rw-r--r--	user	supergroup
h9.xml	file	1.71 KB	1	128 MB	2017-03-19 16:53	rw-r--r--	user	supergroup

Go back to DFS home

Continued….

```
$ hadoop jar wordcount.jar wordcount ex22/bills ex22/word_frequency
```

Execution of wordcount program. Output is obtained as:



Goto : /user/user/ex22/word_frequ go

Go back to dir listing
Advanced view/download options

View Next chunk

```
Force.  4
Forces  10
Forces' 1
Forces; 2
Foreign 15
Forest  6
Forester        1
Forfeiture      2
Forks,  1
Former  1
Fossil  1
Foster  1
Foundation      3
Foundation);    1
Foundation,     1
Foundation.     1
Foundation;     3
Free    1
Freedom 1
Freedom,        1
Freight 1
```

Continued….

```
$ hadoop fs –rm –r ex22/word_frequency
```

Removes the file to monitor using YARN Interface.

```
$ hadoop fs –ls ex22
```

Describes the directory where input and output is placed.

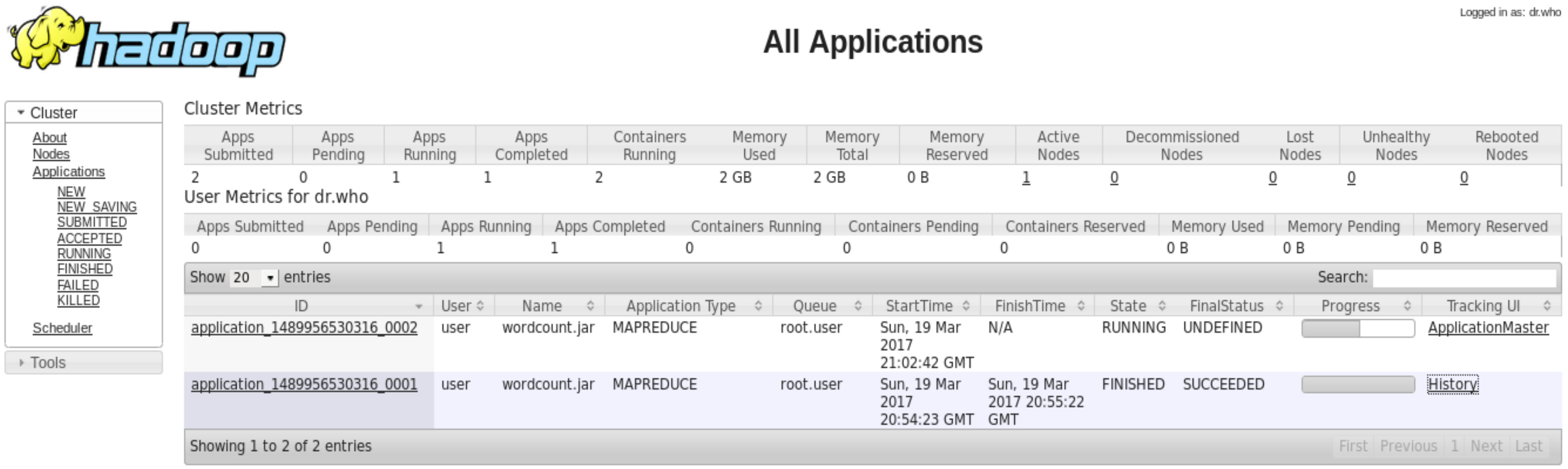
```
$ mr- jobhistory –daemon.sh start historyserver
```

Starts the daemon

```
$ hadoop jar wordcount.jar wordcount ex22/bills ex22/word_frequency
```

Execution of program.

By going to Yarn resource manger in the home page the process happening can be examined.



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
2	0	1	1	2	2 GB	2 GB	0 B	1	0	0	0	0

User Metrics for dr.who

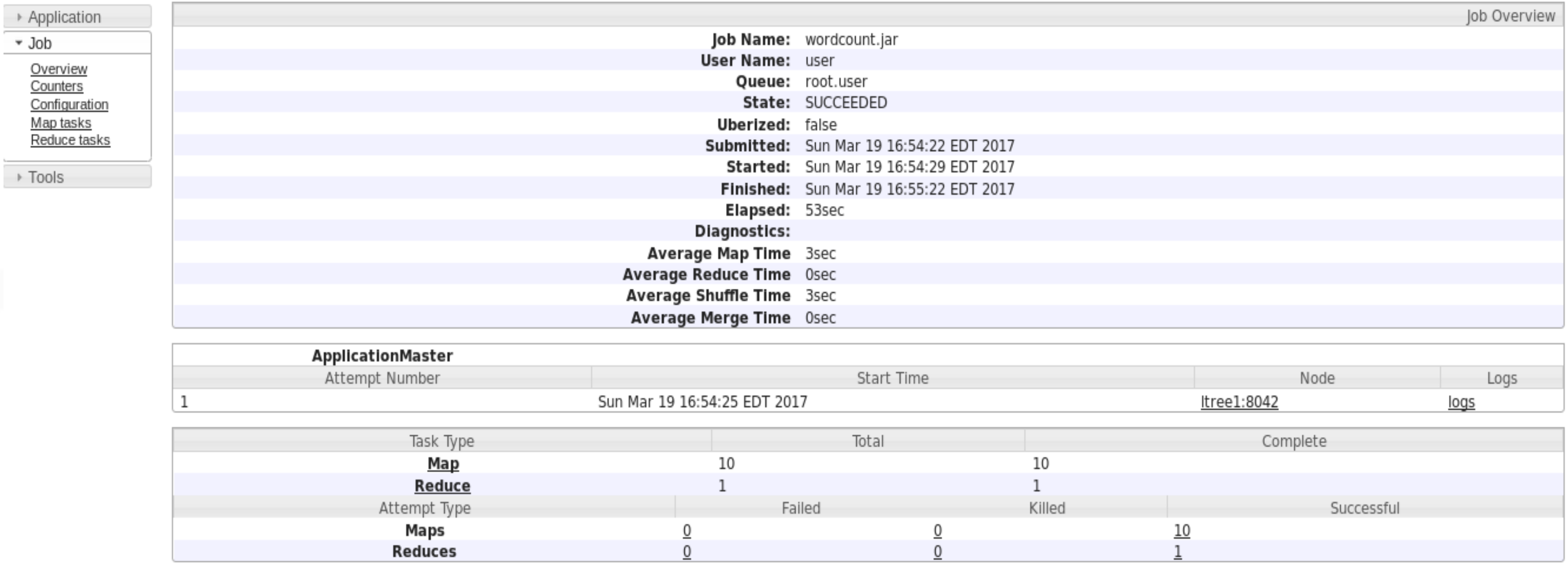
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved
0	0	1	1	0	0	0	0 B	0 B	0 B

Show 20 entries

Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1489956530316_0002	user	wordcount.jar	MAPREDUCE	root.user	Sun, 19 Mar 2017 21:02:42 GMT	N/A	RUNNING	UNDEFINED		ApplicationMaster
application_1489956530316_0001	user	wordcount.jar	MAPREDUCE	root.user	Sun, 19 Mar 2017 20:54:23 GMT	Sun, 19 Mar 2017 20:55:22 GMT	FINISHED	SUCCEEDED		History

Showing 1 to 2 of 2 entries



Job Overview

Job Name:	wordcount.jar
User Name:	user
Queue:	root.user
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Mar 19 16:54:22 EDT 2017
Started:	Sun Mar 19 16:54:29 EDT 2017
Finished:	Sun Mar 19 16:55:22 EDT 2017
Elapsed:	53sec
Diagnostics:	
Average Map Time	3sec
Average Reduce Time	0sec
Average Shuffle Time	3sec
Average Merge Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Mar 19 16:54:25 EDT 2017	ltree1:8042	logs

Task Type	Total	Complete
Map	10	10
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	10
Reduces	0	0	1

Continued….

```
$ ~/stop-hadoop.sh
```

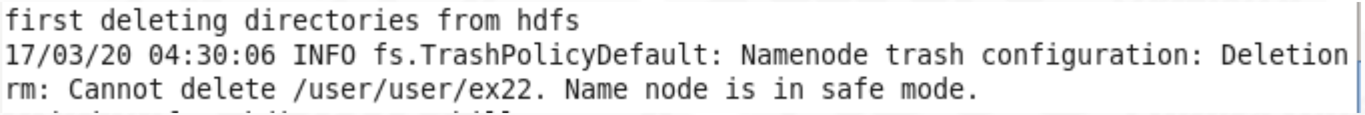
Stops HDFS.

```
$ exit
```

Exits the terminal.

Problems Faced and resolved

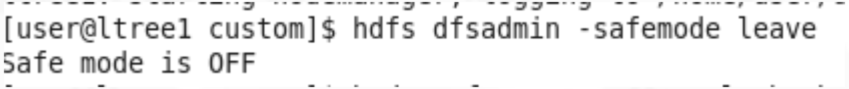
- Problem – Name node went to Safe node, Hence we couldn’t delete files from hdfs.



```
first deleting directories from hdfs
17/03/20 04:30:06 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion
rm: Cannot delete /user/user/ex22. Name node is in safe mode.
```

- Reason for the problem : During start up, Namenode loads the filesystem state from fsimage and edits log file. It then waits for data nodes to report their blocks so that it does not prematurely start replicating the blocks though enough replicas already exist in the cluster. During this time, Namenode stays in safe mode. If data nodes fail to report then Name node continues to be safe mode.

- Solution:



```
[user@ltree1 custom]$ hdfs dfsadmin -safemode leave
Safe mode is OFF
```

Conclusion

- Understood How to {Download, Install, Configure, Run examples} in Host, CDH VM, RedHat VM.
- Understood advantages of using a VM.
- Understood the perks of using Cloudera VM because of all pre-built utilities provided within.
- Understood different modes of Hadoop in real-time

References

- https://en.wikipedia.org/wiki/Cloudera
- https://www.cloudera.com/documentation/other/tutorial/CDH5/topics/ht_usage.html
- RedHat Adapta Learn
- https://hadoop.apache.org/docs/r2.5.2/hadoop-project-dist/hadoop-common/SingleCluster.html
- http://www.tutorialspoint.com/hadoop/
- http://askubuntu.com/questions/673597/ssh-connect-to-host-127-0-0-1-port-2222-connection-refused

Thank You!

(*for coming and being a phenomenal audience*)