Module #: HADOOP ADMINISTRATION PROJECT

Project/Assignment Solution



Module#: HADOOP ADMINISTRATION PROJECT

Assignment Solution– title

Table of Contents

[1. Introduction 3](#_Toc425777070)

[2. Problem Solution 3](#_Toc425777071)

[3. Output **Error! Bookmark not defined.**](#_Toc425777072)

# Introduction

You need to use VMWARE player or any virtualization tool to set up the environment

CentOS 6.5 is preferred

You need to create a regular user in CentOS

You may either configure a DNS server or have the appropriate entries in hosts file

# Problem Solution

Create a 6 Node Hadoop cluster with the following specifications.

1. There should be One separate Machine for Namenode, Job Tracker and Secondary Namenode

1ST MACHINE = Namenode, Datanode, Tasktracker

2nd MACHINE = Jobtracker, Secondary Namenode, Datanode, Tasktracker

---------

1ST MACHINE = Namenode, Datanode, Tasktracker

2nd MACHINE = Jobtracker, Datanode, Tasktracker

3rd MACHINE = Secondary Namenode, Datanode, Tasktracker

---------

1ST MACHINE = Namenode

2nd MACHINE = Jobtracker, Datanode, Tasktracker

3rd MACHINE = Secondary Namenode, Datanode, Tasktracker

4th MACHINE = Datanode, Tasktracker

1. 3 Machines should be running as slave nodes
2. The default HDFS block size should be 128 MB
3. The default Secondary Namenode snapshot interval should be 3 hours

core-site.xml 🡪 fs.checkpoint.period = <seconds>

1. All the Hadoop log files should be stored in /log folder in user’s home folder

hadoop-env.sh 🡪 set log directory location =/home/<user directory>/log

1. Hadoop version should be 1.2.1 and Passwordless SSH should be configured
2. Once created, the cluster must have Rack Awareness configured with each Datanode falling in to separate racks.
3. All the files, which get deleted from HDFS, should be recycled in to Trash Directory with a time interval of 160 minutes. – TRASH SERVER.
4. The user should not be able to upload more than 500 files to the cluster, at any given point of time. - QUOTA MGT.
5. Layout a plan to expand this cluster to accommodate 50 TB of data. The plan should include options to add nodes seamlessly, without formatting. Consider a MOM growth of 20 TB for the cluster. List out the sample configurations for Datanodes as well.

HDFS replication factor =3 Storage=3 (RF) \* 20 =>60 TB

Monthly Growth = 5% Monthly Volume =60TB+5% =63TB AKA -> 64512GB

After one year =64512\*(1+0.05)^12 = 115855 GB

Intermediate MapR Data=25% Dedicated Space=4\*(1-(0.25+0.30))=1.8TB

NON HDFS reserved space/disk =30% Size of a HDD =4TB

Number of datanode to process

For 1st month data=64.512/1800 =35 nodes

12th month data=115.855/1800 = 64 nodes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | NAMENODE | JOBTRACKER | SECONDARY NAMENODE | DATANODE | TASKTRACKER |
|  |  |  |  |  |  |
| CPU | 2+ | 4+ | 2+ | 4+ | 4+ |
| HDD - TOTAL NO. OF DISKS | 1 TB |  | 1 TB |  |  |
| MEMORY | 2 - 4 GB |  | 2 - 4 GB | 4 - 8 GB |  |
| RAID OR NO RAID | RAID | NO RAID | RAID | NO RAID | NO RAID |
| POWER SUPPLIES | Redundant PSUs | Redundant PSUs | Redundant PSUs | Single PSU | Single PSU |
| NETWORK CONFIGURATION | 48-64 GB RAM |  | 32-48 GB RAM | 16-48 Gb RAM |  |

1. Once completed, pen down the step by step process to upgrade this cluster from Hadoop 1.2.1 to Hadoop 2.6.0

Hadoop Upgrade:

**Pre-Upgrade Action:**

1. Capture hadoop version info

cmd: hadoop version

cmd: cat data/\*\*\*/VERSION

2. Keep latest version of hadoop in respective folder - Linux

3. Stop mapred daemons (cmd:stop-mapred.sh)

4. Take backup

config files (cmd: hadoop fs -put hadoop/conf/\* backup/ConfigInput)

Metadata (cmd: hadoop fsck /-files -blocks -locations -racks > backup/Mdata.log)

All folders included namenode dir(cmd: hadoop fs -lsr / > backup/Folders.log)

DFS Admin Report (cmd: hadoop dfsadmin -report > backup/DFSAdmin.log)

Optional: "

If required , we need to move data from one cluster to other cluster using command "distcp"

5. Stop all dfs daemons (cmd: Stop dfs.sh)

**Upgrade Action:**

6. Extract hadoop new version (cmd: tar zxvf hadoop-2.6.0.tar.gz)

7. Set .bash\_profile (cmd : vim .bash\_profile ) for hadoop new version path and ensure $HADOOP\_HOME/**sbin**

8. Source .bash\_profile

9. Configure all required xml files under hadoop/etc/hadoop/...ensure name properties value

10. Configure hadoop.env.sh for java home and hadoo home path

11. Ensure the latest version details reflected (Cmd : hadoop version)

12. Upgrade Namenode (cmd: hadoop-daemon.sh start namemode **-upgrade**)

13. Start Datanode (cmd: hadoop-daemon.sh start datanode)

14. Start SecondaryNamenode (cmd: hadoop-daemon.sh start secondarynamenode)

15. Verify layout version (cmd: cat data/nn/current/VERSION )

16. Confirm the status (" Not Yet not finalised " ) through web interface using namenode IP

**Post-Upgrade Action:**

17. If any discrepancy found by comparing backup (folder/report/input/file count etc), we can rollback

18. Finalize the upgrade process if everything is fine

(cmd: hdfs dfsadmin -finalizeUpgrade)

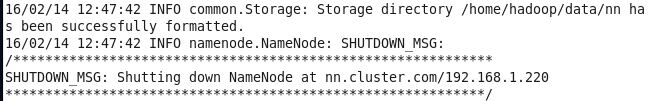
RESULT

2 Machines Configuration:

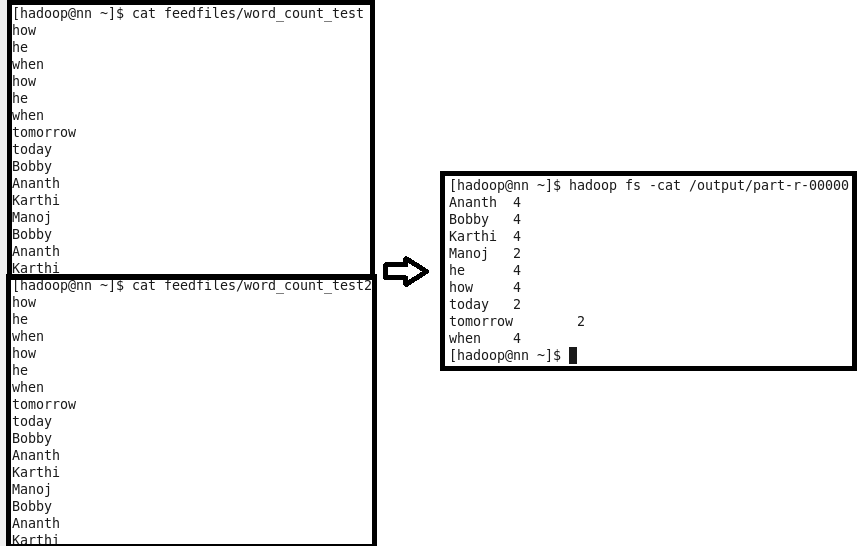
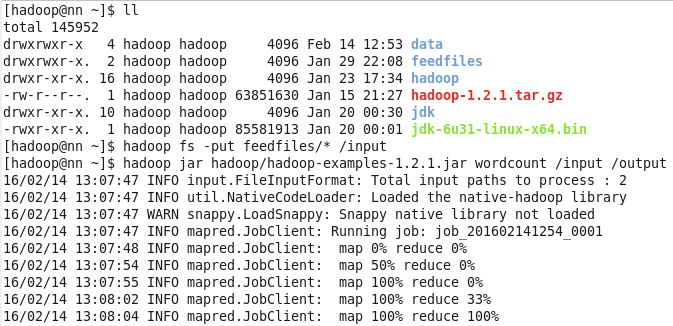
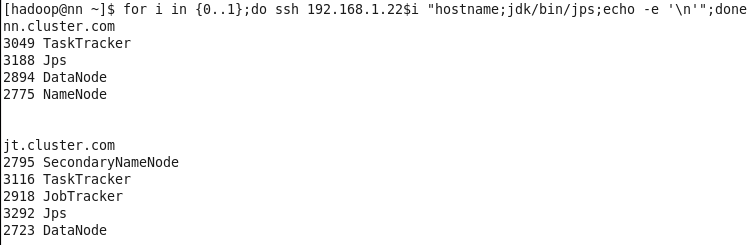
1ST MACHINE = Namenode, Datanode, Tasktracker

2nd MACHINE = Jobtracker, Secondary Namenode, Datanode, Tasktracker

Namenode Format Result:



Active hadoop Daemons at 2 machines:

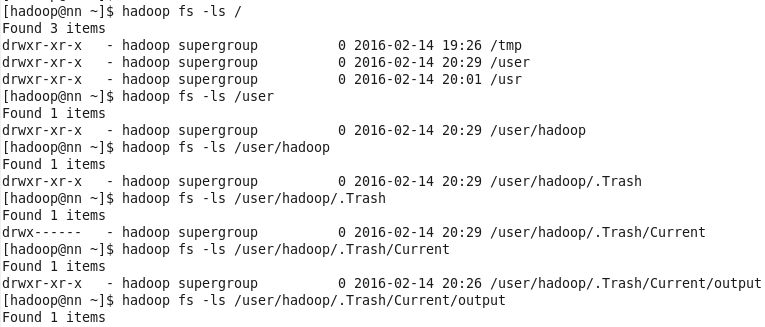
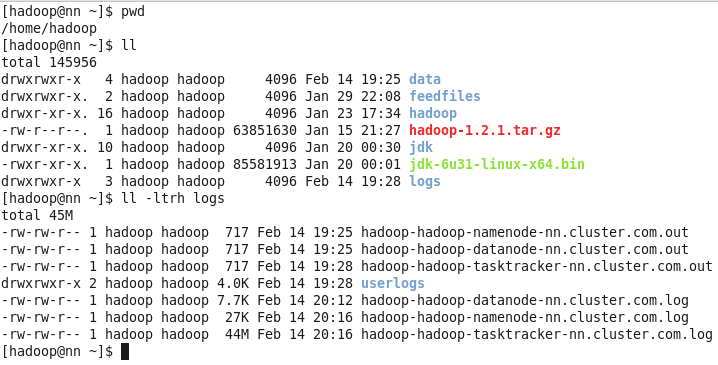
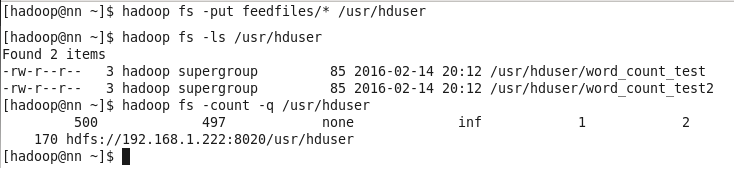
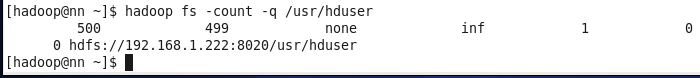
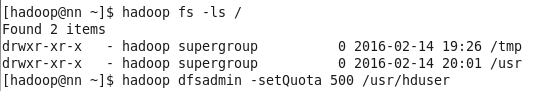
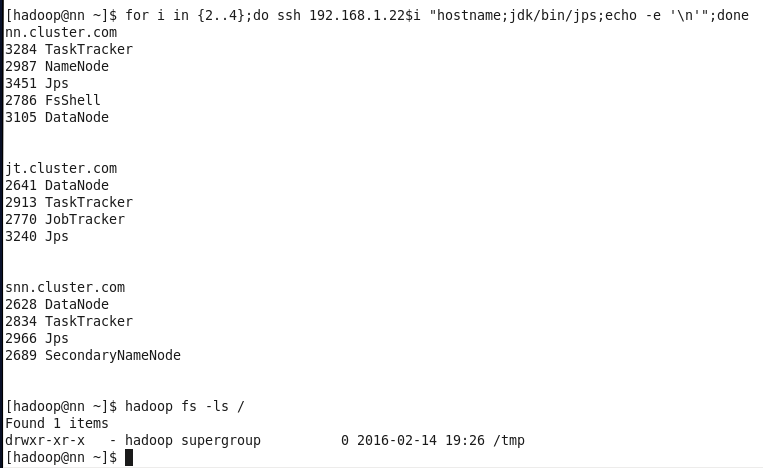
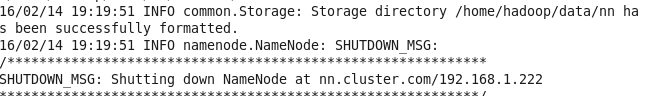


3 Machines Configuration:

1ST MACHINE = Namenode, Datanode, Tasktracker

2nd MACHINE = Jobtracker, Datanode, Tasktracker

3rd MACHINE = Secondary Namenode, Datanode, Tasktracker



4 Machines Configuration

1ST MACHINE = Namenode

2nd MACHINE = Jobtracker, Datanode, Tasktracker

3rd MACHINE = Secondary Namenode, Datanode, Tasktracker

4th MACHINE = Datanode, Tasktracker

