connect • monitor • manage

CONNECT - Lab Guide

Deploy Apache Hadoop with Emulex OneConnect OCe14000 Ethernet Network Adapters

Hardware, software and configuration steps needed to deploy Apache Hadoop 2.4.1 with the Emulex family of OCe14000 10GbE Network Adapters

IMPLEMENTER'S LAB

hadoop

EMULEX®

Table of contents

Introduction

The rapid growth of social media, cellular advances and requirements for data analytics has challenged the traditional methods of data storage and data processing for many large business and government entities. To solve the data storing and processing challenges, organizations are starting to deploy large clusters of Apache Hadoop – a solution that helps manage the vast amount of what is commonly referred to as big data. The Emulex OneConnect® family of OCe14000 10Gb Ethernet (10GbE) Network Adapters play an important role in the Hadoop cluster to move the data efficiently across the cluster.

This lab guide describes the necessary hardware, software and configuration steps needed to deploy Apache Hadoop 2.4.1 with the Emulex family of OCe14000 10GbE Network Adapters. A brief introduction to centralized management of OCe14000 Adapters using Emulex OneCommand® Manager is also being reviewed.

Intended audience – System and network architects and administrators

Hardware requirements

For this lab guide, we implemented a five-node cluster, which is scalable as required, because adding a new DataNode to a Hadoop cluster is a very simple process. However, NameNode's RAM and disk space must be taken into consideration before adding additional DataNodes.

NameNode is the most important part of a Hadoop Distributed File System (HDFS). It keeps track of all the directory trees of all files in the file system, tracks where the file data is kept across the cluster, and is the single point of failure in a Hadoop cluster. With Hadoop 2.0, this issue has been addressed with the HDFS High Availability feature (refer to Apache's "HDFS High Availability Guide"). As a best practice, it is always recommended to implement high availability in the cluster.

DataNode stores data in the HDFS. A cluster will always have more than one DataNode, with data replicated across all them.

The required hardware for implementing and testing Hadoop with OCe14000 10GbE Adapters is listed below:

Hardware components	Quantity	Description	Comments
Server	4 or more	Any server with Intel/AMD processors, which support Linux	1 NameNode (MasterNode); 3 or more DataNode (SlaveNode, JobHistoryNode, ResourceManager)
Hard drives	2 or more per server	Any SAS or SATA drive	
RAID controller	4 or more	Any server, which supports Linux	
RAM	48GB+ per server		
Emulex OCe14000 Network Adapters	4 or more	10GbE network adapters	The OCe14102-UM adapter was used in this configuration
Switch	1	10Gbps switch	
Cables	4 or more	10Gbps optical SFP+ cables	

Figure 1. List of hardware required.

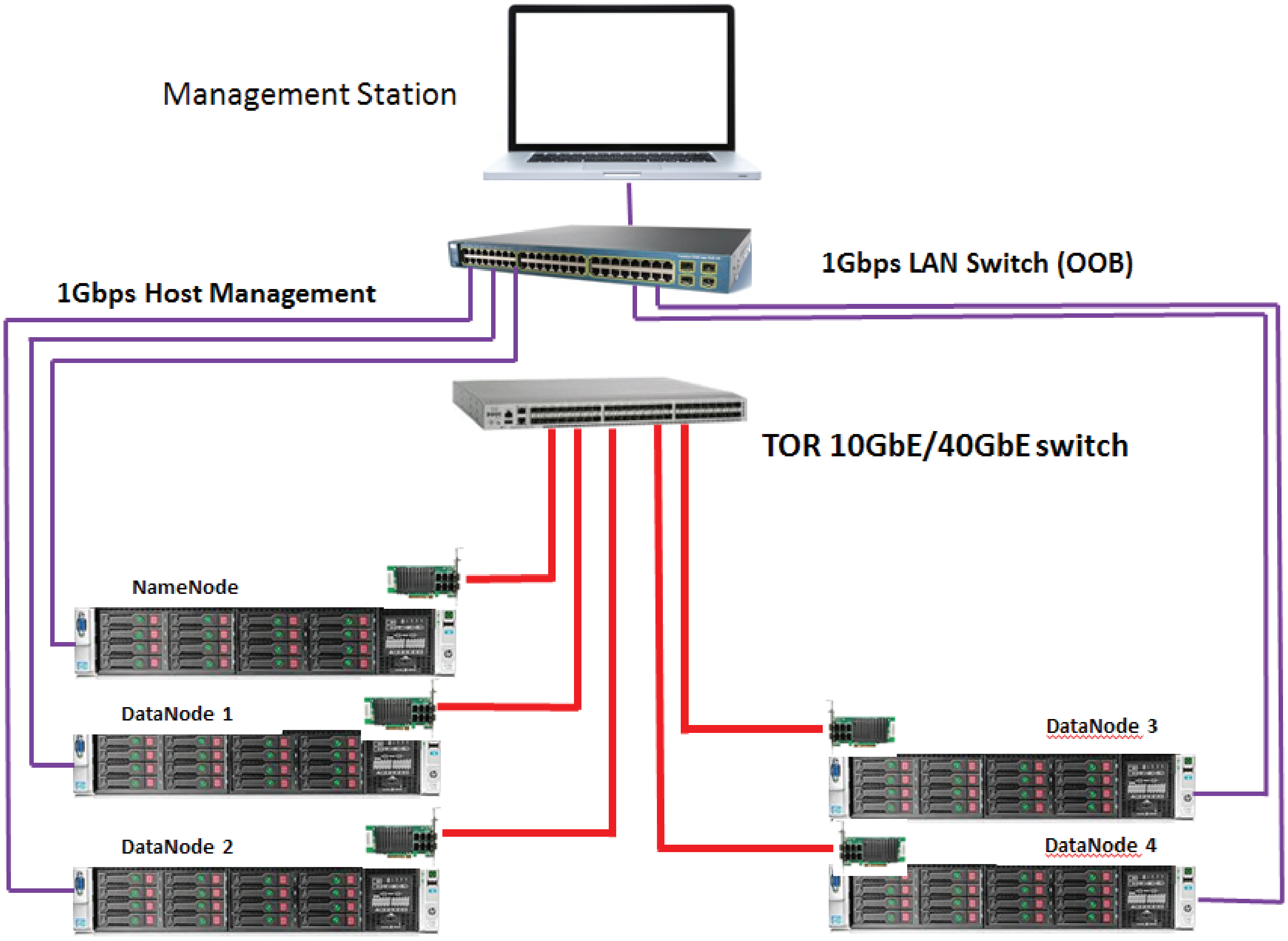


Figure 2. Sample configuration.

Recommended Hadoop cluster hardware specification

There are a lot of factors affecting the choice of hardware for a new Hadoop cluster. Hadoop runs on industry-standard hardware. However, selecting the right mix of CPU, hard drives or RAM for any workload can help you to make the Hadoop cluster more efficient. The recommendations below are formulated from Cloudera. HortonWorks also has a best practices section for hardware selection. Please consider your organization's workload before selecting the hardware components.

Node	Hardware Components	Specifications
NameNode	CPU	2 quad/hex/oct core CPUs, running at least 2GHz
	Hard drive	2 or more 1TB hard disks in a RAID configuration
	RAM	48 - 128GB
DataNode	CPU	2 quad/hex/oct core CPUs, running at least 2GHz
	Hard drive	2 or more 1TB hard disks in a JBOD configuration
	RAM	64 - 512GB

Figure 3. Recommended server specifications for a NameNode and DataNode.

Software requirements

Software components	Quantity	Description	Application note components
OS	4 or more	Any supported Linux or Windows OS	CentOS 6.4
Java	4 or more	Java 1.6.x or higher	Java 1.7.0
Apache Hadoop	4 or more	Stable 2.x version	Hadoop 2.4.1
OCe14102 Firmware	4 or more	Download the latest firmware from Emulex website	10.2.370.19
OCe14102 Driver	4 or more	Download the latest driver from the Emulex website	10.2.363.0
Emulex OneCommand Manager	4 or more	Download the latest version of OneCommand Manager from the Emulex website	10.2.370.16

Figure 4. Software requirements.

Installation and configuration of servers

Install CentOS 6.4 on five servers. For this lab guide, five different names were assigned to the servers. Essential services like ResourceManager and JobHistory Server were split across the SlaveNode. The names along with the roles are listed below:

1. Elephant : NameNode Server (MasterNode)
2. Monkey : JobHistory Server, DataNode (SlaveNode)
3. Horse : ResourceManager, DataNode (SlaveNode)
4. Tiger : DataNode (SlaveNode)
5. Lion : DataNode (SlaveNode)

Connect the OCe14102 Adapter to the PCI Express (PCIe) slots. Upgrade the adapter with the latest firmware, driver and version of OneCommand Manager.

Connect port 0 of every server to the top-of-rack (TOR) switch and ensure that the link is up.

Note – There will be a system reboot required for upgrading the firmware.

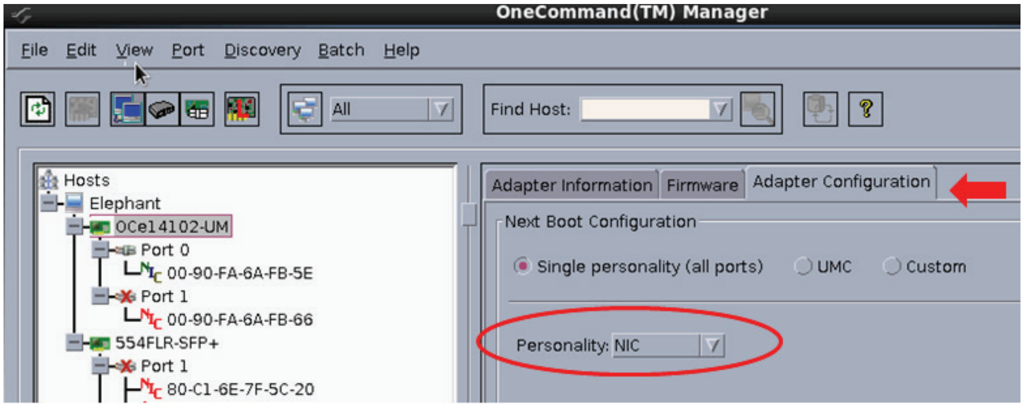
1. Verification of NIC profile, firmware and driver versions for all servers

Verify the Network Interface Card (NIC) profile in the following steps:

a) Start OneCommand Manger.

b) Select the OCe14102-UM Adapter and go to the Adapter Configuration tab.

c) Ensure that the personality is set to NIC.



Next Boot Configuration

● Single personality (all ports) ○ UMC ○ Custom

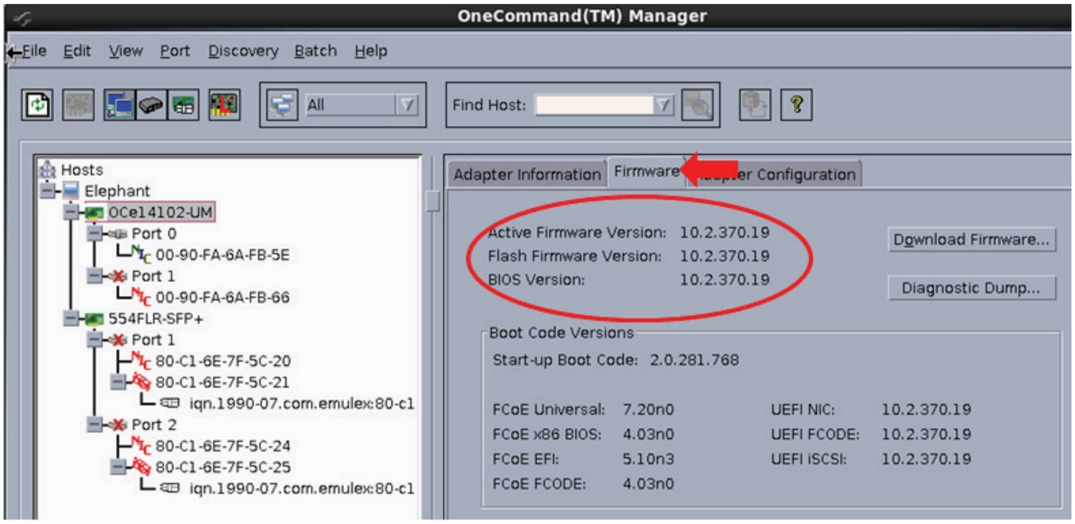
Personality: NIC

Figure 5. NIC personality verification.

Verify the firmware version in the following steps:

a) Select the OCe14102-UM Adapter and go to the firmware tab.

b) Ensure that the firmware version is the same as the one you have upgraded.



Active Firmware Version: 10.2.370.19
Flash Firmware Version: 10.2.370.19
BIOS Version: 10.2.370.19

Boot Code Versions

Start-up Boot Code: 2.0.281.768

FCoE Universal: 7.20n0
FCoE x86 BIOS: 4.03n0
FCoE EFI: 5.10n3
FCoE FCODE: 4.03n0
UEFI NIC: 10.2.370.19
UEFI FCODE: 10.2.370.19
UEFI iSCSI: 10.2.370.19

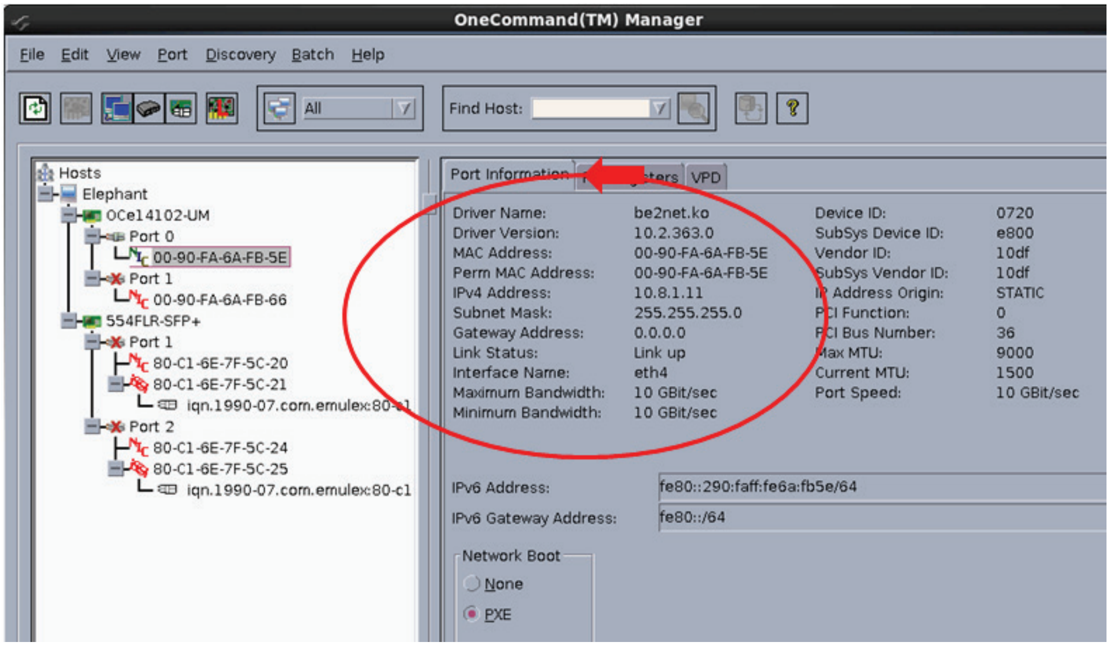
Figure 6. Firmware verification.

Verify the driver version in the following steps:

a) Start OneCommand Manager.

b) Select NIC from port 0 or port 1 of the OCe14102-UM Adapter and go to the Port Information tab.

c) Ensure that the driver version is same as the one you have upgraded.



Driver Name: be2net.ko
Driver Version: 10.2.363.0
MAC Address: 00-90-FA-6A-FB-5E
Perm MAC Address: 00-90-FA-6A-FB-5E
IPv4 Address: 10.8.1.11
Subnet Mask: 255.255.255.0
Gateway Address: 0.0.0.0
Link Status: Link up
Interface Name: eth4
Maximum Bandwidth: 10 GBit/sec
Minimum Bandwidth: 10 GBit/sec

Device ID: 0720
SubSys Device ID: e800
Vendor ID: 10df
SubSys Vendor ID: 10df
IP Address Origin: STATIC
PCI Function: 0
PCI Bus Number: 36
Max MTU: 9000
Current MTU: 1500
Port Speed: 10 GBit/sec

IPv6 Address: fe80::290:faff:fe6a:fb5e/64
IPv6 Gateway Address: fe80::/64

Network Boot
○ None
● PXE

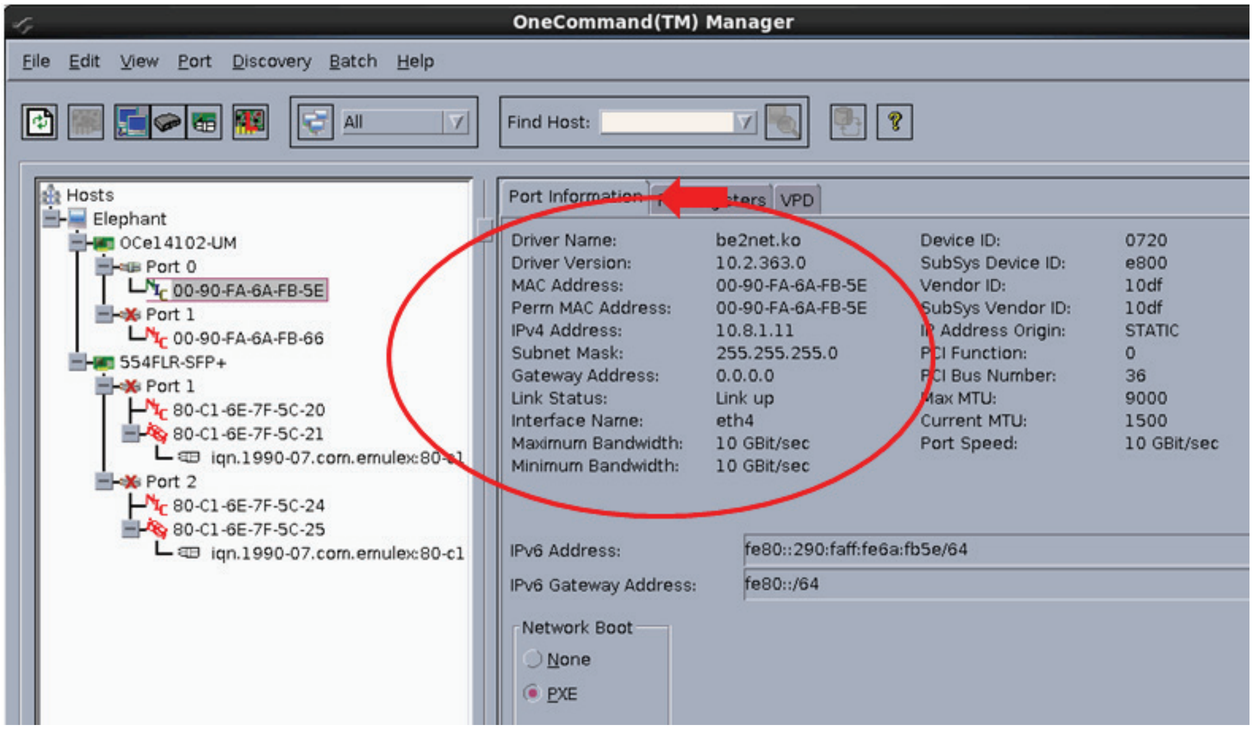
Figure 7. Driver verification.

2. Assign IPv4 address to port 0 on every server

a) Assign the following IP addresses to port 0 of the OCe14102 Adapter using Network Manager or ipconfig.

Server	Port 0 IP
Elephant	10.8.1.11
Monkey	10.8.1.12
Tiger	10.8.1.13
Lion	10.8.1.14
Horse	10.8.1.15

b) Verify the IP address assignment using OneCommand Manager.



Driver Name: be2net.ko
Driver Version: 10.2.363.0
MAC Address: 00-90-FA-6A-FB-5E
Perm MAC Address: 00-90-FA-6A-FB-5E
IPv4 Address: 10.8.1.11
Subnet Mask: 255.255.255.0
Gateway Address: 0.0.0.0
Link Status: Link up
Interface Name: eth4
Maximum Bandwidth: 10 GBit/sec
Minimum Bandwidth: 10 GBit/sec

Device ID: 0720
SubSys Device ID: e800
Vendor ID: 10df
SubSys Vendor ID: 10df
IP Address Origin: STATIC
PCI Function: 0
PCI Bus Number: 36
Max MTU: 9000
Current MTU: 1500
Port Speed: 10 GBit/sec

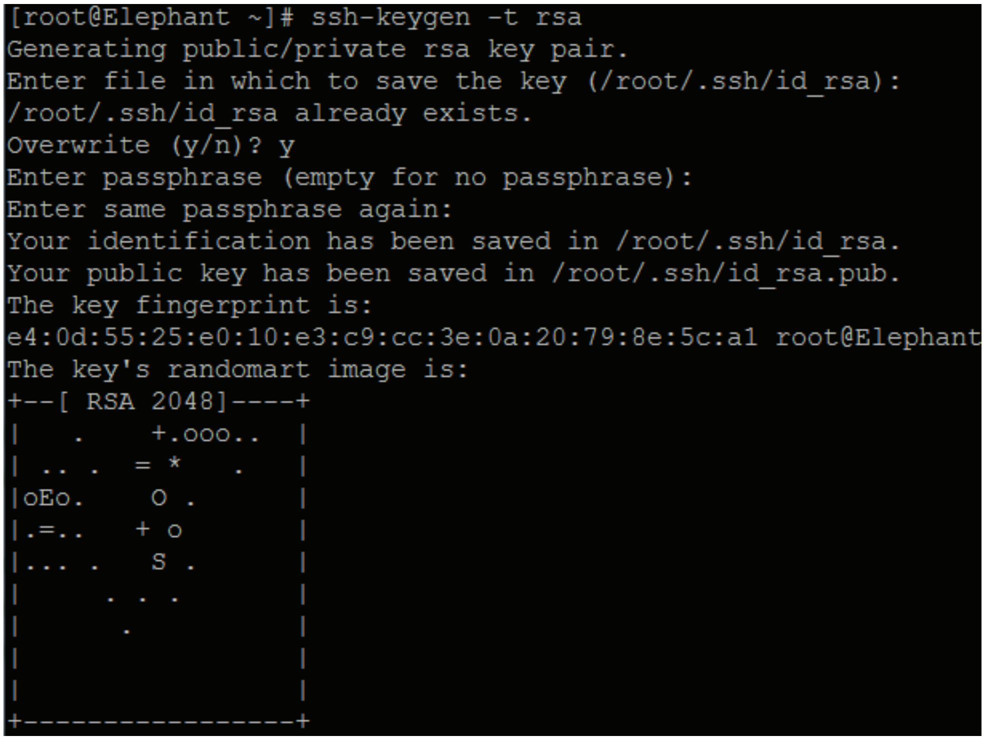
IPv6 Address: fe80::290:faff:fe6a:fb5e/64
IPv6 Gateway Address: fe80::/64

Network Boot
○ None
● PXE

Figure 8. IP address verification.

3. Setup passwordless Secure Shell (SSH) on all servers

a) Generate authentication SSH-keygen keys on Elephant using **ssh-keygen -t rsa**.

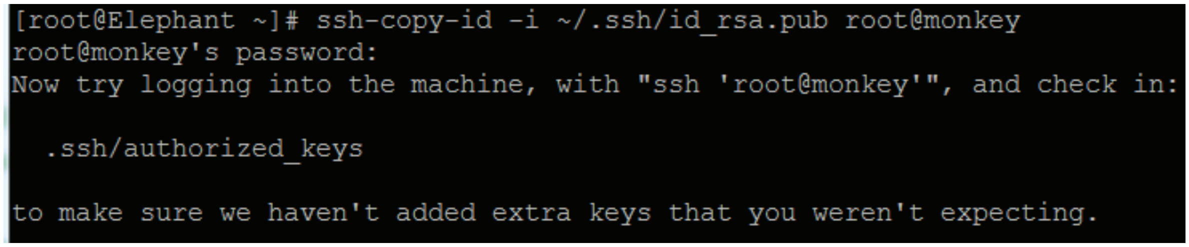


```
[root@Elephant ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
e4:0d:55:25:e0:10:e3:c9:cc:3e:0a:20:79:8e:5c:a1 root@Elephant
The key's randomart image is:
+--[ RSA 2048]----+
|   .    +.ooo..  |
| .. .  = *   .   |
|oEo.    O .      |
|.=..   + o       |
|... .   S .      |
|     . . .       |
|      .          |
|                 |
|                 |
+-----------------+
```

Figure 9. ssh-keygen output.

Note – logged in user should have privileges to run the ssh-keygen command or run the sudo command before running the ssh-kegen command.

b) Run **ssh-copy-id -i ~/.ssh/id_rsa.pub root@HOSTNAME** for Monkey, Lion, Tiger and Horse.

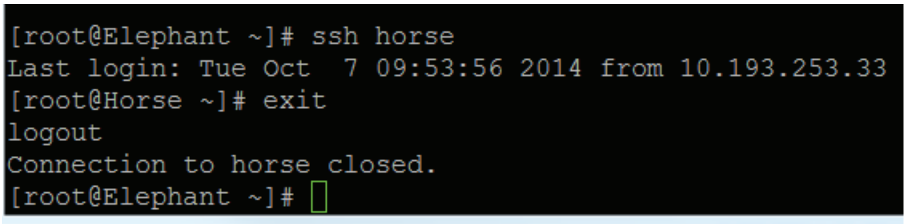


```
[root@Elephant ~]# ssh-copy-id -i ~/.ssh/id_rsa.pub root@monkey
root@monkey's password:
Now try logging into the machine, with "ssh 'root@monkey'", and check in:

  .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
```

Figure 10. ssh-copy sample output.

Note – ssh-copy-id command must be run for all the servers in the cluster to enable the login to SSH without a password (passwordless login).

c) Verify that the password SSH is working.



```
[root@Elephant ~]# ssh horse
Last login: Tue Oct  7 09:53:56 2014 from 10.193.253.33
[root@Horse ~]# exit
logout
Connection to horse closed.
[root@Elephant ~]#
```

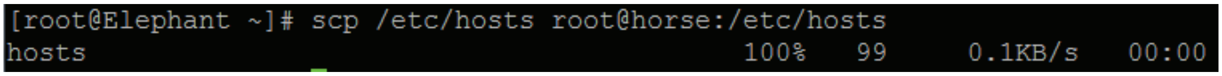
Figure 11. Verification of passwordless SSH.

4. Configure /etc/hosts file
   The host names of the servers in the cluster along with the corresponding IP address need to be added to the /etc/hosts file. This is used for the operating system to map host names to IP addresses.

   a) The port 0 IP assigned to the OCe14000 series adapter corresponding to each host should be added to the /etc/hosts file. Add the lines listed below to the /etc/hosts file on Elephant.

```
10.8.1.11   Elephant
10.8.1.15   Horse
10.8.1.12   Monkey
10.8.1.13   Tiger
10.8.1.14   Lion
```

   b) Copy the /etc/hosts file from elephant to Horse, Monkey, Tiger and Lion using the scp command



```
[root@Elephant ~]# scp /etc/hosts root@horse:/etc/hosts
hosts                                          100%   99     0.1KB/s   00:00
```

Figure 12. scp /etc/hosts.

Note – /etc/hosts should be copied to all the servers in the Hadoop cluster

5. Install Java
   Download and install Java on all of the hosts in the Hadoop cluster.

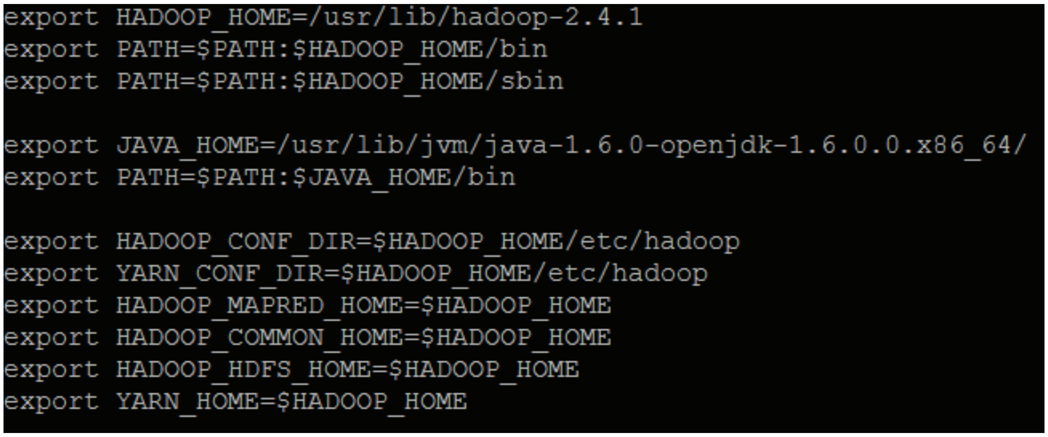
6. Disable Firewall
   For this lab guide, the firewall was disabled. Please consult your network administrator for allowing the necessary services by the firewall to make the Hadoop cluster work.

Installing and configuring Hadoop

Notes:

- All config file changes will be made on Elephant (the NameNode) and then should be copied to Horse, Lion, Tiger and Monkey.
- The user was "root" in this lab guide's installations and configuration. If you are creating a local user, please make sure appropriate privileges are assigned to the user to run all of the commands.
- Config files for Hadoop are present under /usr/lib/hadoop-2.4.1/etc/hadoop or $HADOOP_HOME/etc/Hadoop.
- Scripts are located in the Appendix.

1. Download Hadoop 2.x tarball to /usr/lib on all of the hosts in the Hadoop Cluster.
2. Extract the tarball.
3. Setup environment variables according to your environment needs. It's necessary to set up the variables correctly because the jobs will fail if the variables are set up incorrectly. Edit the ~/.bashrc file on Elephant (NameNode).

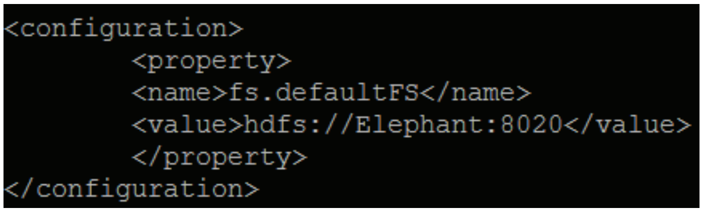


```
export HADOOP_HOME=/usr/lib/hadoop-2.4.1
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin

export JAVA_HOME=/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0.x86_64/
export PATH=$PATH:$JAVA_HOME/bin

export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
```

Figure 13. Environment variables in ~/.bashrc.

4. Set up the slaves, *.xml config files and hadoop_env.sh according to the needs of your environment. For this lab guide, the following files were edited with the parameters listed below:

a) core-site.xml



```
<configuration>
        <property>
        <name>fs.defaultFS</name>
        <value>hdfs://Elephant:8020</value>
        </property>
</configuration>
```

Figure 14. core-site.xml sample configuration.

b) mapred-site.xml

Name	Value
mapreduce.framework.name	yarn
mapreduce.jobhistory.address	monkey:10020
mapreduce.jobhistory.webapp.address	monkey:19888
yarn.app.mapreduce.am.staging-dir	/user
mapred.child.java.opts	-Xmx512m

Figure 15. mapred-site.xml sample configuration.

c) hdfs-site.xml

Name	Value
dfs.namenode.name.dir	file:///disk1/dfs/nn,file:///disk2/dfs/nn
dfs.datanode.data.dir	file:///root/Desktop/data1/dn

Figure 16. hdfs-site.xml sample configuration.

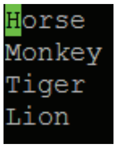
d) yarn-site.xml

Name	Value
yarn.resourcemanager.hostname	horse
yarn.application.classpath	Leave the value specified in the file
yarn.nodemanager.aux-services	mapreduce_shuffle
yarn.nodemanager.local-dirs	file:///root/Desktop/data1/nodemgr/local
yarn.nodemanager.log-dirs	/var/log/hadoop-yarn/containers
yarn.nodemanager.remote-app-log-dir	/var/log/hadoop-yarn/apps
yarn.log-aggregation-enable	true

Figure 17. yarn-site.xml.

e) slaves

Add the names of all the DataNode in this file.



```
Horse
Monkey
Tiger
Lion
```

Figure 18. Sample slaves configuration.

Note – Do not copy slaves

f) hadoop-env.sh

Edit the JAVA_HOME, HADOOP_HOME and HADOOP_CONF_DIR according to your environment. For this lab guide, the following values were used:

Name	Value
HADOOP_HOME	/usr/lib/hadoop-2.4.1
HADOOP_CONF	$HADOOP_HOME/etc/hadoop
JAVA_HOME	/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0.x86_64/

Figure 19. hadoop-env.sh sample configuration.

g) Copy all config files (except for slaves) and environment variables to Horse, Lion, Tiger and Monkey using the script copy_config.sh.

5. Create the Namenode and Datanode (hdfs) directories using the following commands:

```
mkdir -p /disk1/dfs/nn
mkdir -p /disk2/dfs/nn
mkdir -p /root/Desktop/data1/dn
mkdir -p /root/Desktop/data1/nodemgr/local
```

Note – These directories are based on the available storage on the setup. Please update the directories according to the directory structure in your environment.

6. Format and start the Namenode on Elephant using one of the following commands:

```
/usr/lib/hadoop-2.4.1/bin/hadoop namenode -format
or
$HADOOP_HOME/bin/hadoop namenode  -format
```

7. Start the Namenode on Elephant using the following script located in the appendix:

```
NameNode_startup.sh
```

8. Start the Datanode on Horse, Lion, Tiger and Monkey using the following command:

```
cd $HADOOP_HOME
sbin/hadoop-daemon.sh start datanode
```

9. Create the yarn directories using the following commands:

```
hadoop fs -mkdir -p /var/log/hadoop-yarn/containers
hadoop fs -mkdir -p /var/log/hadoop-yarn/apps
hadoop fs -mkdir -p /user
hadoop fs -mkdir -p /user/history
```

10. Stop the Datanode on Horse, Lion, Tiger and Monkey using the following command:

```
cd $HADOOP_HOME
sbin/hadoop-daemon.sh stop datanode
```

11. Start the services on the servers according to the roles assigned to them. The scripts are located in the Appendix. Use the following scripts to start the services:

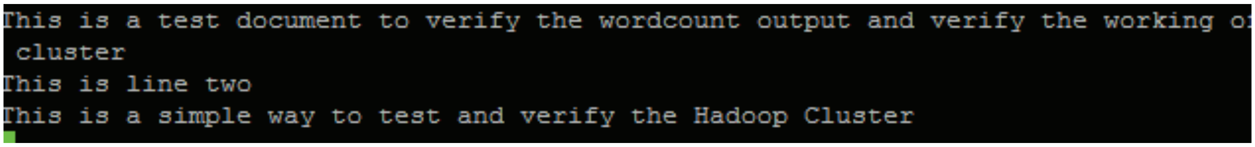
- Horse: **ResourceManager_startup.sh**
- Monkey: **JobHistory_startup.sh**
- Lion and Tiger: **DataNode_startup.sh**

Note – Steps 5-10 should be performed for first time installations only.

Verification of Hadoop cluster

To verify the Hadoop cluster, a simple word count example can be run.

1. Create a sample file.
   Create a sample file with a few test sentences using the VI editor on Elephant: **vi input_word_count**



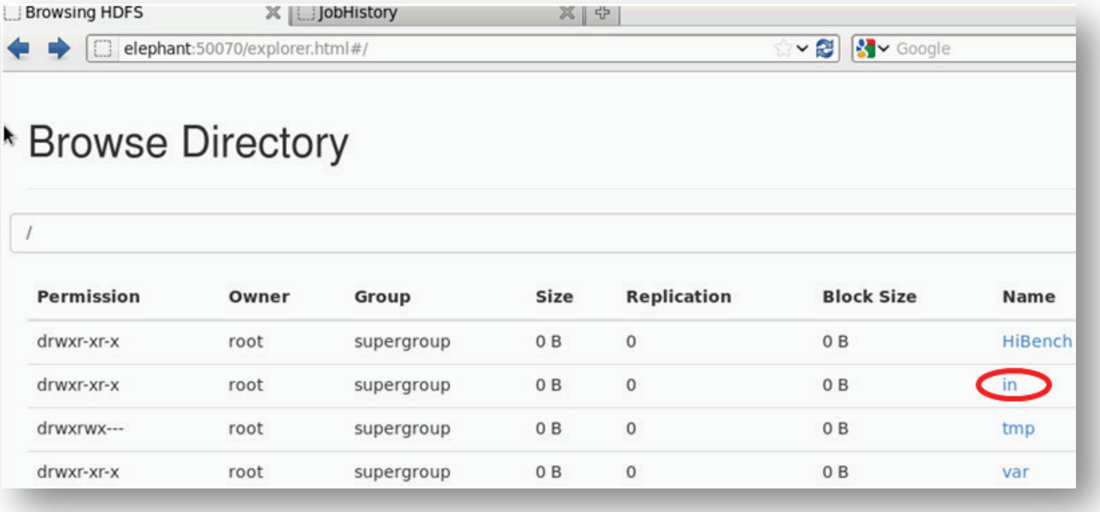
```
This is a test document to verify the wordcount output and verify the working o
 cluster
This is line two
This is a simple way to test and verify the Hadoop Cluster
```

Figure 20. Sample file.

2. Copy the file to HFDS.
   From Elephant, create a directory named "in" in HDFS and copy the sample file **input_word_count** to HDFS under the "in" directory:

```
hadoop fs -mkdir /in
hadoop fs -put input_word_count /in
hadoop fs -ls /in
```

The directory, file and the actual location of the file can be viewed using a web interface on the NameNode. The web address is "elephant:50070". This will also show that the replication factor of each block is 3 and it is saved at three different DataNodes (Figures 21 and 22).

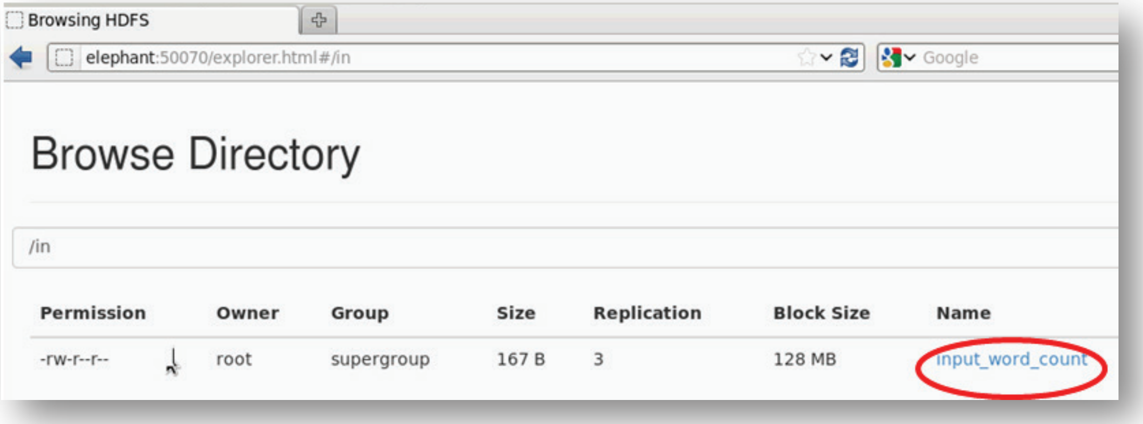


Browse Directory

/

Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	root	supergroup	0 B	0	0 B	HiBench
drwxr-xr-x	root	supergroup	0 B	0	0 B	in
drwxrwx---	root	supergroup	0 B	0	0 B	tmp
drwxr-xr-x	root	supergroup	0 B	0	0 B	var

Figure 21. Verification of HFDS directory listing using web interface.

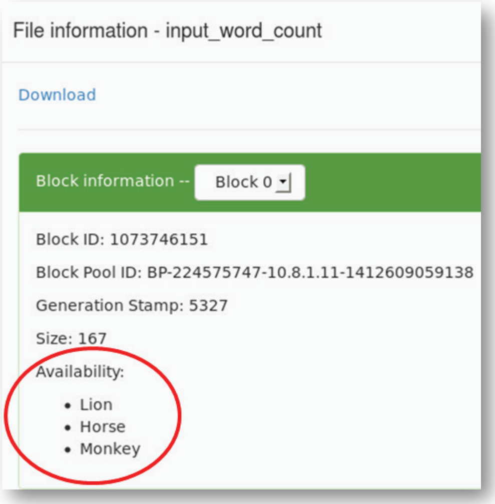


Browse Directory

/in

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	root	supergroup	167 B	3	128 MB	input_word_count

Figure 22. File size in HFDS.


File information - input_word_count

Block ID: 1073746151

Block Pool ID: BP-224575747-10.8.1.11-1412609059138

Generation Stamp: 5327

Size: 167

Availability:

- Lion
- Horse
- Monkey

Figure 23. Replication factor and file location information on the HFDS.

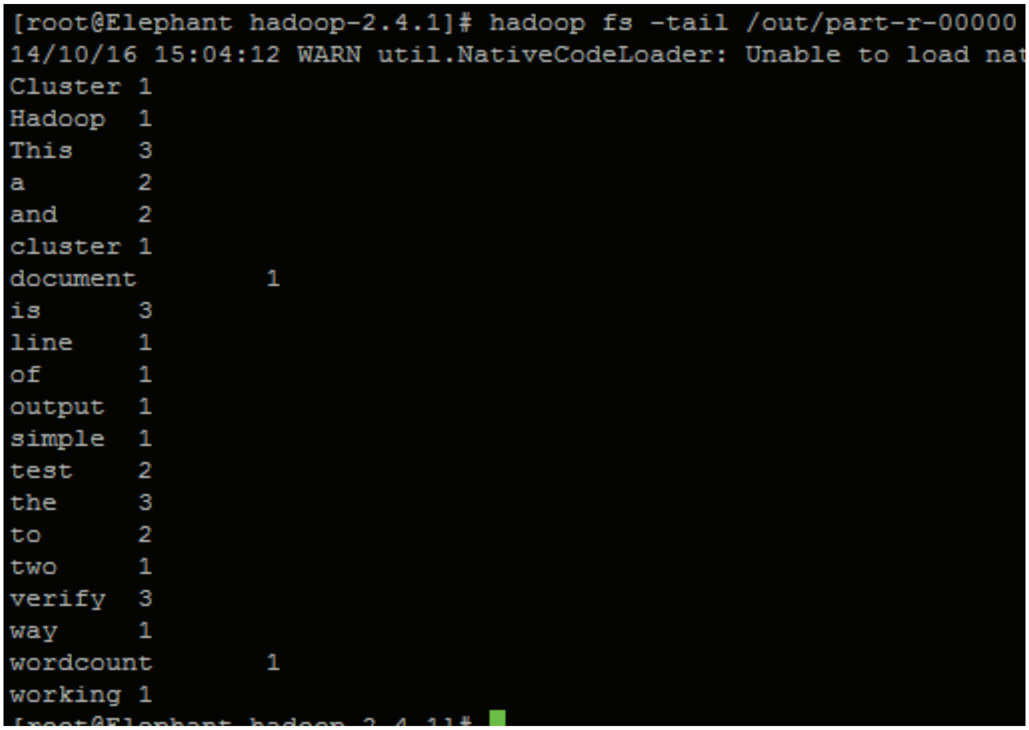
3. Run the inbuilt *.jar application.

   Run the command listed below to start the word count application.

```
/usr/lib/hadoop-2.4.1/bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.4.1.jar wordcount /in/input_word_count /out
```

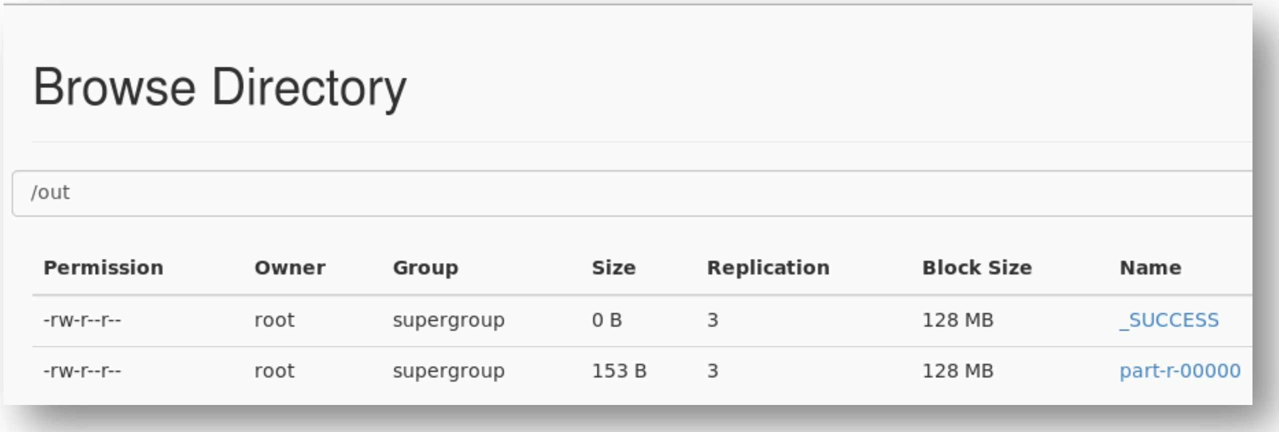
Note – Depending on the Apache Hadoop version, the *.jar name will change.

The output will be displayed on the terminal. The result location can also be viewed using the web interface.



```
[root@Elephant hadoop-2.4.1]# hadoop fs -tail /out/part-r-00000
14/10/16 15:04:12 WARN util.NativeCodeLoader: Unable to load nat
Cluster 1
Hadoop  1
This    3
a       2
and     2
cluster 1
document        1
is      3
line    1
of      1
output  1
simple  1
test    2
the     3
to      2
two     1
verify  3
way     1
wordcount       1
working 1
[root@Elephant hadoop-2.4.1]#
```

Figure 24. Sample output.



Browse Directory

/out

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	root	supergroup	0 B	3	128 MB	_SUCCESS
-rw-r--r--	root	supergroup	153 B	3	128 MB	part-r-00000

Figure 25. Sample web interface output in HFDS.

Conclusion

A complete overview of hardware and software components required for successfully deploying and evaluating Apache Hadoop with Emulex OCe14000 Network Adapters was presented in this lab guide. More details about the Hadoop architecture can be obtained from the official Apache Hadoop website: http://hadoop.apache.org/. The solution described in this deployment guide is scalable for a larger number of DataNodes and racks to suit the environment and needs of your organization. Emulex OCe14000 Network Adapters can be used in the cluster to move the data efficiently across the nodes.

Appendix

File: copy_config_sh

```
#!/bin/bash
#scp to tiger to copy core,hdfs,mapred,yarn-site.xml,hadoop-env.sh and bashrc
scp /usr/lib/hadoop-2.4.1/etc/hadoop/hdfs-site.xml root@tiger:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/yarn-site.xml root@tiger:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/mapred-site.xml root@tiger:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/core-site.xml root@tiger:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /etc/hosts root@tiger:/etc/hosts
scp /usr/lib/hadoop-2.4.1/etc/hadoop/hadoop-env.sh root@tiger:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp ~/.bashrc root@tiger:~/.bashrc
echo Copied files to Tiger

#scp to monkey to copy core,hdfs,mapred,yarn-site.xml,hadoop-env.sh and bashrc
scp /usr/lib/hadoop-2.4.1/etc/hadoop/hdfs-site.xml root@monkey:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/yarn-site.xml root@monkey:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/mapred-site.xml root@monkey:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/core-site.xml root@monkey:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/hadoop-env.sh root@monkey:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /etc/hosts root@monkey:/etc/hosts
scp ~/.bashrc root@monkey:~/.bashrc
echo Copied files to Monkey

#scp to horse to copy core,hdfs,mapred,yarn-site.xml,hadoop-env.sh and bashrc
scp /usr/lib/hadoop-2.4.1/etc/hadoop/hdfs-site.xml root@horse:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/yarn-site.xml root@horse:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/mapred-site.xml root@horse:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/core-site.xml root@horse:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/hadoop-env.sh root@horse:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /etc/hosts root@horse:/etc/hosts
scp ~/.bashrc root@horse:~/.bashrc
echo Copied files to Horse

#scp to Lion to copy core,hdfs,mapred,yarn-site.xml,hadoop-env.sh and bashrc
scp /usr/lib/hadoop-2.4.1/etc/hadoop/hdfs-site.xml root@lion:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/yarn-site.xml root@lion:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/mapred-site.xml root@lion:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/core-site.xml root@lion:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /usr/lib/hadoop-2.4.1/etc/hadoop/hadoop-env.sh root@lion:/usr/lib/hadoop-2.4.1/etc/hadoop/
scp /etc/hosts root@lion:/etc/hosts
scp ~/.bashrc root@lion:~/.bashrc
echo Copied files to Lion
```

File: NameNode_startup script

```
#!/bin/bash
#start namenode only!!
cd $HADOOP_HOME
sbin/hadoop-daemon.sh start namenode
jps
jps -v
```

File: DataNode_startup.sh script

```
#!/bin/bash
#start datanode,nodemanager on Tiger & Lion!!
cd $HADOOP_HOME
sbin/hadoop-daemon.sh start datanode
sbin/yarn-daemon.sh start nodemanager
jps
jps -v
```

File: ResourceManager_startup.sh script

```
#!/bin/bash
#start datanode,resourcemanager,nodemanager
cd $HADOOP_HOME
sbin/hadoop-daemon.sh start datanode
sbin/yarn-daemon.sh start resourcemanager
sbin/yarn-daemon.sh start nodemanager
jps
jps -v
```

File: JobHistory_startup.sh script

```
#!/bin/bash
#start datanode,jobhistory server,nodemanager on monkey!!!
cd $HADOOP_HOME
sbin/hadoop-daemon.sh start datanode
sbin/yarn-daemon.sh start nodemanager
sbin/mr-jobhistory-daemon.sh start historyserver
jps
jps -v
```

References

Solution Implementer's Lab
http://www.implementerslab.com/

Emulex Ethernet networking and storage connectivity products
http://www.emulex.com/products/ethernet-networking-storage-connectivity/

Apache Hadoop overview
http://hadoop.apache.org/

How to set up a multimode Hadoop cluster
http://wiki.apache.org/hadoop/#Setting_up_a_Hadoop_Cluster

Cloudera resources
http://www.cloudera.com/content/cloudera/en/resources.html

Hortonworks on Hadoop architecture
http://hortonworks.com/blog/

Wikipedia
http://en.wikipedia.org/wiki/Apache_Hadoop

DataNode
http://wiki.apache.org/hadoop/DataNode

NameNode
http://wiki.apache.org/hadoop/NameNode

EMULEX®

World Headquarters 3333 Susan Street, Costa Mesa, CA 92626 +1 714 662 5600
Bangalore, India +91 80 40156789 | Beijing, China +86 10 84400221
Dublin, Ireland +35 3 (0) 1 652 1700 | Munich, Germany +49 (0) 89 97007 177
Paris, France +33 (0) 158 580 022 | Tokyo, Japan +81 3 5325 3261 | Singapore +65 6866 3768
Wokingham, United Kingdom +44 (0) 118 977 2929 | Brazil +55 11 3443 7735

www.emulex.com

©2015 Emulex, Inc. All rights reserved.

This document refers to various companies and products by their trade names. In most cases, their respective companies claim these designations as trademarks or registered trademarks. This information is provided for reference only. Although this information is believed to be accurate and reliable at the time of publication, Emulex assumes no responsibility for errors or omissions. Emulex reserves the right to make changes or corrections without notice. This document is the property of Emulex and may not be duplicated without permission from the Company.