



helper_scripts_and_commands_for_troubleshooting

1. Establish self equivalence within cluster. By these all nodes shall be able to do passphraseless ssh to each other and itself.

1.1 On every node run below to generate public/private key of self

```
#As hadoop user
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
```

1.2 Run the script to collect public key of every node into a common file and dump this as authorized keys for all nodes.

1.2.1 Create a validating script named testallue.sh - user equivalence

```
#testallue.sh

flag=0;
for dip in `cat ip.lst`
do
    ssh hadoop@$dip exit
    status=`echo $?`
    if test $status -ne 0
    then
        echo "Status: SSH is not successful to $dip"
        flag=1
    fi
done;

if test $flag -eq 0
then
    echo "Status: All SSH is successful."
fi;
```

1.2.2 Create a ip.lst file containing ip of every node per line in current directory

1.2.3 Retrieve all public keys locally and generate a combined file name allpub.keys

```
# each remote file locally copied with corresponding ip as filename

for sip in `cat ip.lst`
do
    scp hadoop@$sip:/home/hadoop/.ssh/id_rsa.pub $lf
done

#combine into one
for filename in `ls *.pub`
do
    cat $filename >> allpub.keys
done

chmod 600 allpub.keys

for ip in `cat ip.lst`; do scp allpub.keys hadoop@$ip:/home/hadoop/.ssh/authorized_keys; done;
```

Now onward when you do ssh/scp to and from any participating nodes as hadoop user it must not ask for a password. Type 'yes' it may ask for known_hosts entry.

1.2.4 Run below to copy over and confirm that it shows passphraseless SSH is working within cluster.

```
for ip in `cat ip.lst`
do
    scp ip.lst hadoop@$ip:/home/hadoop/ip.lst;
    scp testallue.sh hadoop@$ip:/home/hadoop/testallue.sh;
    scp /home/hadoop/.ssh/known_hosts hadoop@$ip:/home/hadoop/.ssh/known_hosts;
done;
```

1.2.5 Run testallue.sh on every node from home directory manually.

Open multiple tab for every nodes using below script

```
cl="gnome-terminal"
for ip in `cat ip.lst`
do
    cl="$cl --tab -e \"ssh hadoop@$ip\""
done
eval "$cl"
```

```
chmod +x ~/testallue.sh ./testallue.sh
```

2. Prepare content of /etc/hosts file locally using a script to fix hostnames of participating nodes.

```
echo -n '' > hadoopEtcHosts
for ip in `cat ip.lst`
do
    number=`echo $ip | cut -d . -f 4`
    echo "$ip hadoop-$number" >> hadoopEtcHosts
done
```

3.

Copy etchosts, jdk, hadoop setup and installation scripts on every participating nodes.

Download jdk and hadoop-2.8 from web.

```
for ip in `cat ip.lst`;
do
    scp hadoopEtcHosts hadoop@$ip:/home/hadoop/hadoopEtcHosts;
    scp jdk-8u121-linux-x64.tar.gz hadoop@$ip:/home/hadoop/jdk-8u121-linux-x64.tar.gz;
    scp hadoop-2.8.0.tar.gz hadoop@$ip:/home/hadoop/hadoop-2.8.0.tar.gz;
```

done;

5. Set the hostnames of every system using root login as appropriate: i.e. vi /etc/sysconfig/network HOSTNAME=hadoop-master

6. Make entries into every /etc/hosts for self and others on every node

```
#As root
cat /home/hadoop/hadoopEtcHosts >> /etc/hosts
```

7. Compile and run below java program to make sure that the hostname and cononical names are as what we are going to use in hadoop configuration.

```
import java.net.InetAddress;
import java.net.UnknownHostException;

class CheckDNS
{
public static void main(String args[]) throws UnknownHostException
{
    System.out.println("Hello Java");
    InetAddress addr = InetAddress.getLocalHost();
    System.out.println(
        String.format(
            "IP:%s hostname:%s canonicalName:%s",
            addr.getHostAddress(),
            addr.getHostName(),
            addr.getCanonicalHostName()
        )
    );
}
}
```

Sample Output: [hduser@hadoop-slave16 ~]\$ java CheckDNS Hello Java IP:192.168.26.16 hostname:hadoop-slave16 canonicalName:hadoop-slave16 [hduser@hadoop-slave16 ~]\$

Read more about this requirement from book/web copy [https://books.google.co.in/books?](https://books.google.co.in/books?id=W5VWrrCOuQ8C&pg=PA59&lpg=PA59&dq=dns+found+localhost+hadoop+issue&source=bl&ots=PDhlJxfQ_3&sig=zxie6T1DA7phElSCaSmO7qxz8P4&hl=en&sa=X&ved=0ahUKEwjB65zO7o3TAhXDMY)

[id=W5VWrrCOuQ8C&pg=PA59&lpg=PA59&dq=dns+found+localhost+hadoop+issue&source=bl&ots=PDhlJxfQ_3&sig=zxie6T1DA7phElSCaSmO7qxz8P4&hl=en&sa=X&ved=0ahUKEwjB65zO7o3TAhXDMY](https://books.google.co.in/books?id=W5VWrrCOuQ8C&pg=PA59&lpg=PA59&dq=dns+found+localhost+hadoop+issue&source=bl&ots=PDhlJxfQ_3&sig=zxie6T1DA7phElSCaSmO7qxz8P4&hl=en&sa=X&ved=0ahUKEwjB65zO7o3TAhXDMY)

Hadoop Operations - Page 59 - Google Books Result Hadoop Operations: A Guide for Developers and Administrators

Eric Sammer “O'Reilly Media, Inc.”

Other program to check on DNS <https://github.com/sujee/hadoop-dns-checker> [https://github.com/sujee/hadoop-dns-checker]

999. Helper Command to view ports for http daemon, IPC, etc

```
netstat -nptl | grep `jps | grep NameNode | cut -d ' ' -f 1`
netstat -nptl | grep `jps | grep DataNode | cut -d ' ' -f 1`
netstat -nptl | grep `jps | grep ResourceManager | cut -d ' ' -f 1`
netstat -nptl | grep `jps | grep NodeManager | cut -d ' ' -f 1`
netstat -nptl | grep `jps | grep JobHistoryServer | cut -d ' ' -f 1`
```

1000. XML Format/Parsing Error can be validated upto some extent with following command: xmllint –format filename.xml