If you're initializing a cluster for the first time, it's likely that you'll want to start small with just one or two slaves (DataNodes).

Over time, as you operate your cluster, gaining more experience and confidence, you'll likely want to add more slaves.

I started with a simple NameNode and dual-DataNode cluster configuration.  This article details the steps I took to add an extra node to my cluster.

**Outline**

1. The first step I took was to clone an existing VM.
2. Update the /etc/hosts file
   1. Add the new slave and IP address
   2. Copy this file to ea
3. Start the DataNode

**Clone an existing DataNode**

If you're cloning a DataNode that's already been used within a cluster, you'll want to clean out the $HADOOP\_DATA\_DIR. An easy way to do this is simply delete the existing directory, and re-create it (reference " Configuring the HDFS Base Node.doc from drive" for more information on this step):

**sudo rm -rf $HADOOP\_DATA\_DIR  
 mkdir -p $HADOOP\_DATA\_DIR/data   
 mkdir -p $HADOOP\_DATA\_DIR/name   
 mkdir -p $HADOOP\_DATA\_DIR/local   
 sudo chmod 755 $HADOOP\_DATA\_DIR**

If you plan to clone more than one node, I recommend cloning the additional nodes from this step onward.

Once my data directory is cleared, I use VirtualBox to create my clone:

**VBoxManage clonevm "**%~1**" --name "**%~2**" --register --basefolder %vm%**

Substitute the node names for the parameters above. Reference "VirtualBox for Virtualization doc from drive" for automated cloning recipes.

**Updating the Hosts File**

On your NameNode, update the hosts file:

**sudo gedit /etc/hosts**

and add the node name and IP address for each newly created slave node.

My /etc/hosts file now looks like this:

127.0.0.1 localhost   
 127.0.1.1 CVB   
 192.168.1.10 master   
 192.168.1.11 slave1   
 192.168.1.12 slave2   
 192.168.1.15 slave3   
 **192.168.1.17 slave4   
 192.168.1.18 slave5**   
 192.168.1.14 dev   
 # The following lines are desirable for IPv6 capable hosts   
 ::1 ip6-localhost ip6-loopback   
 fe00::0 ip6-localnet   
 ff00::0 ip6-mcastprefix   
 ff02::1 ip6-allnodes   
 ff02::2 ip6-allrouters

The new lines in my file are in blue bold.  The information in your file will not be identical.

**Copying the Host File**

Do I need to copy the hosts file on my NameNode to each DataNode in the cluster?

Yes, you should do this.  There are cases where DataNodes will talk to each other.  DataNode do this when they are replicating data.  Also, when adding a new DataNode to an existing cluster, data re-balancing (addressed below) will occur.  This requires DataNodes to address one another.

It is important that each DataNode be able to address each other in a consistent fashion.  I recommend maintaining (and editing) the master copy of the hosts file on the NameNode.

On my NameNode, I've created a script in my home directory that will copy the hosts file to each DataNode in my cluster:

**cd ~   
 gedit copy-hosts.sh**

Copy this script with the appropriate modifications:

#!/bin/bash   
 for i in {1..**5**}   
 do   
 scp /etc/hosts **craigtrim**@slave$i:/etc/hosts   
 done

Don't forget to make your shell script executable:

**chmod +x copy-hosts.sh**

**Operational Output**

When I execute this script, it simply reports back to me that all the files were copied successfully:

craigtrim@CVB:~$ **./copy-hosts.sh**   
 hosts 100% 355 0.4KB/s 00:00   
 hosts 100% 355 0.4KB/s 00:00   
 hosts 100% 355 0.4KB/s 00:00   
 hosts 100% 355 0.4KB/s 00:00   
 hosts 100% 355 0.4KB/s 00:00

It doesn't hurt to logon to at least one of the slaves and verify that the file was copied correctly:

craigtrim@CVB:~$ **ssh slave3**   
 Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic x86\_64)   
 \* Documentation: <https://help.ubuntu.com/>   
 229 packages can be updated.   
 87 updates are security updates.   
 Last login: Wed Nov 26 14:26:43 2014 from master   
 craigtrim@CVB:~$ **cat /etc/hosts**   
 127.0.0.1 localhost   
 127.0.1.1 CVB   
 192.168.1.10 master   
 192.168.1.11 slave1   
 192.168.1.12 slave2   
 192.168.1.15 slave3   
 **192.168.1.17 slave4   
 192.168.1.18 slave5** 192.168.1.14 dev   
 # The following lines are desirable for IPv6 capable hosts   
 ::1 ip6-localhost ip6-loopback   
 fe00::0 ip6-localnet   
 ff00::0 ip6-mcastprefix   
 ff02::1 ip6-allnodes   
 ff02::2 ip6-allrouters

Once you gain confidence that the script is doing what it's supposed to, you can likely skip this verification step in the future.

**Updating the Slaves File**

This section is almost identical to the one above. Just as we maintain a master /etc/hosts on the designated NameNode, we'll maintain our master slaves file there.

Let's go ahead and edit it, and add the two new slaves (DataNodes):

**cd $HADOOP\_CONF\_DIR**   
 **gedit slaves**

My /etc/hosts file now looks like this:

slave1   
 slave2   
 slave3   
 **slave4   
 slave5**

The new lines in my file are in blue bold. That's about as simple as it gets. Use of the hosts file prevents us from having to enter actual IP addresses in this file.

In like manner, we can also write a script to copy our slaves file to each DataNode in the cluster.  I call this script "copy-slaves.sh", make it executable in the same manner as the last script we wrote, and enter this:

#!/bin/bash   
 for i in {1..**5**}   
 do   
 scp $HADOOP\_CONF\_DIR/slaves **craigtrim**@slave$i:$HADOOP\_CONF\_DIR   
 done

**Operational Output**

The script output contains nothing surprising:

craigtrim@CVB:~$ **./copy-slaves.sh**    
 slaves 100% 35 0.0KB/s 00:00   
 slaves 100% 35 0.0KB/s 00:00   
 slaves 100% 35 0.0KB/s 00:00   
 slaves 100% 35 0.0KB/s 00:00   
 slaves 100% 35 0.0KB/s 00:00

**Starting Up**

If the entire cluster has been stopped, you should just start the cluster using the start-dfs and the start-yarn shell scripts, as described in this article "[Flipping the Switch](http://trimc-hdfs.blogspot.com/2014/11/starting-hdfs-cluster.html)".

If your cluster is already operational, and you want to "hot deploy" a new DataNode, then execute this command the DataNode:

**hadoop-daemon.sh --config $HADOOP\_CONF\_DIR --script hdfs start datanode**

this works then notices once I start up the datanode (it heartbeats to the namenode - diagram this)

2014-11-26 13:08:56,141 INFO [main] datanode.DataNode (StringUtils.java:startupShutdownMessage(619)) - STARTUP\_MSG:   
 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
 STARTUP\_MSG: Starting DataNode   
 STARTUP\_MSG: host = CVB/127.0.1.1   
 STARTUP\_MSG: args = []   
 STARTUP\_MSG: version = 2.5.2   
 STARTUP\_MSG: classpath = /usr/lib/apache/hadoop/2.5.2/conf:/usr/lib/apache/hadoop/2.5.2/share/hadoop/common/lib/commons-net-3.1.jar:/usr/lib/apache/hadoop/2.5.2/share/hadoop/common/lib/mockito-all-1.8.5.jar:/usr/lib/apache/hadoop/2.5.2/share/hadoop/common/lib/jersey-core-1.9.jar:/usr/lib/apache/hadoop/2.5.2/share/hadoop/common/lib/commons-cli-1.2.jar:/usr/lib/apache/hadoop/2.5.2/share/hadoop/common/lib/commons-lang-  
 ... etc ...

Notice how the auto -rebalancing occurs It looks like this

2014-11-26 13:09:07,488 INFO [DataXceiver for client at /192.168.1.12:47654 [Receiving block BP-1847084755-127.0.1.1-1416961177933:blk\_1073835163\_94339]] datanode.DataNode (DataXceiver.java:writeBlock(766)) -

Received BP-1847084755-127.0.1.1-1416961177933:blk\_1073835163\_94339

src: /192.168.1.12:47654

dest: /192.168.1.15:50010

of size 6221

LIkewise, the summary in the web interface should now show the extra node:

achine generated alternative text:
Summary 
Security is off. 
Safemode is off

For more information on how DataNodes join the cluster, read up on the heartbeat mechanism in the Hadoop Architecture. Refer the “Understanding Hadoop cluster and the network doc from drive”

**Troubleshooting**

**Node is expected to serve this storage**

If you create a clone VM, and don't clean out the $HADOOP\_DATA\_DIR, you're likely to get this error.

2014-11-26 12:56:16,660 WARN [DataNode: [[[DISK]file:/home/craigtrim/HADOOP\_DATA\_DIR/data/]] heartbeating to master/192.168.1.10:9000] datanode.DataNode (BPServiceActor.java:offerService(731)) - Block pool BP-1847084755-127.0.1.1-1416961177933 (Datanode Uuid 9b460762-eba0-45f2-b0b4-f00e11572ed6) service to master/192.168.1.10:9000 is shutting down   
 org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.hdfs.protocol.UnregisteredNodeException): Data node DatanodeRegistration(192.168.1.15, datanodeUuid=9b460762-eba0-45f2-b0b4-f00e11572ed6, infoPort=50075, ipcPort=50020, storageInfo=lv=-55;cid=CID-822421ef-e4d6-49d7-9b25-e0b81ae32b7d;nsid=1137176454;c=0) is attempting to report storage ID 9b460762-eba0-45f2-b0b4-f00e11572ed6. Node 192.168.1.12:50010 is expected to serve this storage.   
 at org.apache.hadoop.hdfs.server.blockmanagement.DatanodeManager.getDatanode(DatanodeManager.java:475)   
 at org.apache.hadoop.hdfs.server.blockmanagement.BlockManager.processReport(BlockManager.java:1702)   
 at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.blockReport(NameNodeRpcServer.java:1049)   
 at org.apache.hadoop.hdfs.protocolPB.DatanodeProtocolServerSideTranslatorPB.blockReport(DatanodeProtocolServerSideTranslatorPB.java:152)   
 at org.apache.hadoop.hdfs.protocol.proto.DatanodeProtocolProtos$DatanodeProtocolService$2.callBlockingMethod(DatanodeProtocolProtos.java:28061)   
 at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:585)   
 at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:928)   
 at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2013)   
 at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2009)   
 at java.security.AccessController.doPrivileged(Native Method)   
 at javax.security.auth.Subject.doAs(Subject.java:422)   
 at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1614)   
 at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2007)   
 at org.apache.hadoop.ipc.Client.call(Client.java:1411)   
 at org.apache.hadoop.ipc.Client.call(Client.java:1364)   
 at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:206)   
 at com.sun.proxy.$Proxy12.blockReport(Unknown Source)   
 at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)   
 at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)   
 at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)   
 at java.lang.reflect.Method.invoke(Method.java:483)   
 at org.apache.hadoop.io.retry.RetryInvocationHandler.invokeMethod(RetryInvocationHandler.java:187)   
 at org.apache.hadoop.io.retry.RetryInvocationHandler.invoke(RetryInvocationHandler.java:102)   
 at com.sun.proxy.$Proxy12.blockReport(Unknown Source)   
 at org.apache.hadoop.hdfs.protocolPB.DatanodeProtocolClientSideTranslatorPB.blockReport(DatanodeProtocolClientSideTranslatorPB.java:214)   
 at org.apache.hadoop.hdfs.server.datanode.BPServiceActor.blockReport(BPServiceActor.java:476)   
 at org.apache.hadoop.hdfs.server.datanode.BPServiceActor.offerService(BPServiceActor.java:699)   
 at org.apache.hadoop.hdfs.server.datanode.BPServiceActor.(BPServiceActor.java:834)   
 at java.lang.Thread.run(Thread.java:745)   
 2014-11-26 12:56:16,664 WARN [DataNode: [[[DISK]file:/home/craigtrim/HADOOP\_DATA\_DIR/data/]] heartbeating to master/192.168.1.10:9000] datanode.DataNode (BPServiceActor.java:run(845)) - Ending block pool service for: Block pool BP-1847084755-127.0.1.1-1416961177933 (Datanode Uuid 9b460762-eba0-45f2-b0b4-f00e11572ed6) service to master/192.168.1.10:9000   
 2014-11-26 12:56:16,766 INFO [DataNode: [[[DISK]file:/home/craigtrim/HADOOP\_DATA\_DIR/data/]] heartbeating to master/192.168.1.10:9000] datanode.DataNode (BlockPoolManager.java:remove(103)) - Removed Block pool BP-1847084755-127.0.1.1-1416961177933 (Datanode Uuid 9b460762-eba0-45f2-b0b4-f00e11572ed6)   
 2014-11-26 12:56:16,766 INFO [DataNode: [[[DISK]file:/home/craigtrim/HADOOP\_DATA\_DIR/data/]] heartbeating to master/192.168.1.10:9000] datanode.DataBlockScanner (DataBlockScanner.java:removeBlockPool(273)) - Removed bpid=BP-1847084755-127.0.1.1-1416961177933 from blockPoolScannerMap   
 2014-11-26 12:56:16,766 INFO [DataNode: [[[DISK]file:/home/craigtrim/HADOOP\_DATA\_DIR/data/]] heartbeating to master/192.168.1.10:9000] impl.FsDatasetImpl (FsDatasetImpl.java:shutdownBlockPool(1755)) - Removing block pool BP-1847084755-127.0.1.1-1416961177933   
 2014-11-26 12:56:18,768 WARN [main] datanode.DataNode (DataNode.java:secureMain(2019)) - Exiting Datanode   
 2014-11-26 12:56:18,770 INFO [main] util.ExitUtil (ExitUtil.java:terminate(124)) - Exiting with status 0   
 2014-11-26 12:56:18,772 INFO [Thread-1] datanode.DataNode (StringUtils.java:run(645)) - SHUTDOWN\_MSG:   
 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
 SHUTDOWN\_MSG: Shutting down DataNode at CVB/127.0.1.1   
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

The solution is to clear out the directory as mentioned above, and restart the DataNode.