|  |
| --- |
| Command List for Hadoop components and Hadoop Admin |
|  |

|  |
| --- |
|  |



© Brain4ce Education Solutions Pvt. Ltd.

# Command List for Hadoop Components on Hadoop Admin

Hive Commands:

Download hive tar file

https://hive.apache.org/downloads.html

Click on download release now, click on nearest mirror. Choose the parents directory for a version and download

apache-hive-0.13.1-bin.tar.gz

Once done, untar it to your preferred location and create a link for it, make your admin user as owner of hive link and directory and update your .bash rc with hive path

$cd /usr/local/hive

( As I have untarred hive in /usr/local and created a link hive-->hive directory)

Before we bring up hive shell..

We need to update hive-default..... config file.

-It should tell where namenode is running (same property tag as in core-site...file)

-It should tell where data of hive will be stored on hdfs.(search for warehouse related parameter and give a path where hive will store data on hdfs

In our case it is /user/hive/warehouse. Thus this path should exist on hdfs or created by us with right permissions)

Note\*\* cluster should be up and running.

$cd /usr/local/hadoop

$bin/hadoop fs -mkdir /user

$bin/hadoop fs -mkdir /user/hive

$bin/hadoop fs -mkdir /user/hive/warehouse

$bin/hadoop fs -chmod g+w /user/hive/warehouse

$bin/hive

Setting parameter using hive shell

hive>SET mapred.job.tracker=xxxx:50030;

Creating db, tb in hive and loading data into from local FS

hive>

create database xxxdb;

create table xxxtb(column1 datatype,column2 datatype);

create table xxxtb(id int,name string);

load data local inpath './examples/files/kv1.txt' overwrite into table xxxtb;

create table xxxtb2(column1 datatype,column2 datatype) partioned by (yy datatype);

create table xxxtb2(id int,name string) partioned by (ds string);

load data local inpath './examples/files/kv2.txt' overwrite into table xxxtb2 partition(ds='2008-08-15');

load data local inpath './examples/files/kv3.txt' overwrite into table xxxtb2 partition(ds='2008-08-08');

Loading data from hdfs into local FS

load data inpath '/user/hduser/kv2.txt' overwrite into table xxxtb2 partition(ds='2008-08-15');

If you run a query from hive shell with where clause, it will show you mr running to extract data.

Refer to HIVE PROGRAMING PDF FOR MORE PRACTICE.

====================================================

Sqoop Command :

Setups to setup Sqoop

Download sqoop tar file

http://sqoop.apache.org/ and click on Nearby mirror and then on the link on top whichis suggested as per your location and then from parent directory download

sqoop-1.4.5.bin\_\_hadoop-1.0.0.tar.gz for hadoop version 1.

Download appropriate MySQL connector

http://dev.mysql.com/downloads/connector/j by choosing your OS i.e. independent platform option and download

Platform Independent (Architecture Independent), Compressed TAR Archive 5.1.34 3.6M

Download (mysql-connector-java-5.1.34.tar.gz)

Once done, untar sqoop and MySQL connector to your preferred location

Create a link to these directories and update your .bashrc with path of sqoop

In my case, I untarred in /usr/local and created a link

$cd /usr/local

$sudo ln -s sqoop-1.4.5.bin\_\_hadoop-1.0.0 sqoop

$sudo ln -s mysql-connector-java-5.1.34 mysql

make hduser(my admin userid for hadoop) as owner of these directories and links

$sudo chown -R hduser:hadoop sqoop-1.4.5.bin\_\_hadoop-1.0.0

$sudo chown -R hduser:hadoop sqoop

$sudo chown -R hduser:hadoop mysql-connector-java-5.1.34

$sudo chown -R hduser:hadoop mysql

Now copy the mysql-xxxxx.jar file from mysql-connector-java-5.1.34 dir to sqoop/lib

Also make sure MySQL is setup on your machine.

You can install MySQL by giving

$sudo apt-get install mysql-server

$mysql-----

should bring you to

mysql>

if done

mysql>exit;

We can browse through MySQL databases and tables to choose which table's data needs to be brought to hdfs.

Note\*\* cluster should be up and running

and now we can copy contents of table from MySQL to hdfs using sqoop

Command is :

$cd /usr/local/sqoop

bin/sqoop import --connect jdbc:mysql://localhost/dbname --username xxxx --password yyyy --table tablename

--m 1

====================================================

Hbase Command :

bin/hadoop job -list all to check if multiple jobs are running.

Test Hadoop Cluster??? fr filesize

To test write performance, write 10 files each of 1000 megabytes. It will write an output in a dir /benchmarks where all metrics are written.

bin/hadoop jar hadoop-test-0.20.205.0.jar TestDFSIO -write -nrFiles 10 -fileSize 1000

To test read performance,read 10 files each of 1000 megabytes. It will write an output in a dir /benchmarks where all metrics are written.

bin/hadoop jar hadoop-test-0.20.205.0.jar TestDFSIO -read -nrFiles 10 -fileSize 1000

bin/hadoop jar hadoop-test-0.20.205.0.jar TestDFSIO -clean

Generate Tera Data:

To generate dummy data within hdfs instead of downloading dataset from somewhere and uploading on hdfs.

bin/hadoop jar hadoop/hadoop-examples...jar teragen 1000 /user/hduser/terasort-input

Using your dummy data you can check sorting performance

bin/hadoop jar hadoop/hadoop-examples...jar terasort /user/hduser/terasort-input /user/hduser/terasort-output

bin/hadoop job -history all /user/hduser/terasort-input

Hadoop Admin setup :

We need to make one machine as NFS so that active namenode can write edits to a shared location and standby namenode can read from shared location.

For Ubuntu:

On machine which you want make as your NFS.

sudo apt-get install nfs-kernel-server

To make sure there is no problem in mounting an NFS share, make sure nfs-common package is installed on your clients(machines from where you will mount data on nfs)

To install nfs-common

sudo apt-get install nfs-common

sudo vi /etc/exports

path \*(rw,sync,no\_root\_sqash)

path needs to be created and ur admin user can be made owner of path.

sudo service nfs-kernel-server start

Now going to each machine which will act as namenode give the following command.

showmount -e ipaddress of ur nfs machine

This should show you path on nfs which was shared. Now create a mount point where you want edits to be written and which will be shared

say /mnt/filer and make hduser as owner of it.

Now,

mount -t nfs ipaddress of ur nfs machine:/shared path /mnt/filer

same needs to be done on other nn machines also.

Setting up on Centos:

How To Set Up an NFS Mount on CentOS 6

About NFS (Network File System) Mounts

NFS mounts work to share a directory between several servers.

This has the advantage of saving disk space, as the home directory is only kept on one server,

and others can connect to it over the network. When setting up mounts, NFS is most effective for permanent fixtures that should always be accessible

Setup

An NFS mount is set up between at least two servers. The machine hosting the shared network is called the server, while the ones that connect to it are called ‘clients’.

This tutorial requires 2 servers: one acting as the server and one as the client.

We will set up the server machine first, followed by the client.

The following IP addresses will refer to each one:

Master: 12.34.56.789

Client: 12.33.44.555

The system should be set up as root. You can access the root user by typing

sudo su

Setting Up the NFS Server

Step 1:

Download the Required Software. Start off by using apt-get to install the nfs programs.

yum install nfs-utils nfs-utils-lib

Subsequently, run several startup scripts for the NFS server:

chkconfig nfs on

service rpcbind start

service nfs start

Step 2: Export the Shared Directory

The next step is to decide which directory we want to share with the client server. The chosen directory should then be added to the /etc/exports file, which specifies both the directory to be shared and the details of how it is shared.

Suppose we wanted to share the directory, /home. We need to export the directory:

vi /etc/exports

Add the following lines to the bottom of the file, sharing the directory with the client:

/home 12.33.44.555(rw,sync,no\_root\_squash,no\_subtree\_check)

These settings accomplish several tasks:

rw: This option allows the client server to both read and write within the shared directory

sync: Sync confirms requests to the shared directory only once the changes have been committed.

no\_subtree\_check: This option prevents the subtree checking. When a shared directory is the subdirectory of a larger filesystem, nfs performs scans of every directory above it, in order to verify its permissions and details. Disabling the subtree check may increase the reliability of NFS, but reduce security.

no\_root\_squash: This phrase allows root to connect to the designated directory

Once you have entered in the settings for each directory, run the following command to export them:

exportfs -a

Setting Up the NFS Client

Step 1: Download the Required Software

Start off by using apt-get to install the nfs programs.

yum install nfs-utils nfs-utils-lib

Step 2: Mount the Directories

Once the programs have been downloaded to the the client server, create the directory that will

contain the NFS shared files

mkdir -p /mnt/nfs/home

Then go ahead and mount it

mount 12.34.56.789:/home /mnt/nfs/home

You can use the df -h command to check that the directory has been mounted. You will see it last on the list.

df -h

Filesystem Size Used Avail Use% Mounted on

/dev/sda 20G 783M 18G 5% /

12.34.56.789:/home 20G 785M 18G 5% /mnt/nfs/home

Additionally, use the mount command to see the entire list of mounted file systems.

mount

Your list should look something like this:

/dev/sda on / type ext4 (rw,errors=remount-ro)

none on /proc/sys/fs/binfmt\_misc type binfmt\_misc (rw)

sunrpc on /var/lib/nfs/rpc\_pipefs type rpc\_pipefs (rw)

nfsd on /proc/fs/nfsd type nfsd (rw)

12.34.56.789:/home on /mnt/nfs/home type nfs (rw,noatime,nolock,bg,nfsvers=2,intr,tcp,actimeo=1800,addr=12.34.56.789)

Testing the NFS Mount

Once you have successfully mounted your NFS directory, you can test that it works by creating a file on the Client and checking its availability on the Server.

Create a file in the directory to try it out:

touch /mnt/nfs/home/example

You should then be able to find the files on the Server in the /home.

ls /home

You can ensure that the mount is always active by adding the directory to the fstab file on the client. This will ensure that the mount starts up after the server reboots.

vi /etc/fstab

12.34.56.789:/home /mnt/nfs/home nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0

You can learn more about the fstab options by typing in:

man nfs

After any subsequent server reboots, you can use a single command to mount directories specified in the fstab file:

mount -a

You can check the mounted directories with the two earlier commands:

df -h

mount

Removing the NFS Mount

Should you decide to remove a directory, you can unmount it using the umount command:

cd

sudo umount /directory name

You can see that the mounts were removed by then looking at the filesystem again.

df -h

You should find your selected mounted directory gone.

Setup these in two machines which will act as active and standby namenode.

In my case aj1-active

aj4-standby

in core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://mycluster</value>

<description>The name of default file system</description>

</property>

</configuration>

In hdfs-site.xml

<configuration>

<name>dfs.nameservices</name>

<value>mycluster</value>

<property>

<name>dfs.replication</name>

<value>2</value>

<description>to specifiy replication</description>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/wkdaydec1ha/name</value>

<final>true</final>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/wkdaydec1ha/data1</value>

<final>true</final>

</property>

<property>

<name>dfs.ha.namenodes.mycluster</name>

<value>nn1,nn2</value>

</property>

<property>

<name>dfs.namenode.rpc-address.mycluster.nn1</name>

<value>aj1:9000</value>

</property>

<property>

<name>dfs.namenode.rpc-address.mycluster.nn2</name>

<value>aj4:9000</value>

</property>

<property>

<name>dfs.namenode.http-address.mycluster.nn1</name>

<value>aj1:50070</value>

</property>

<property>

<name>dfs.namenode.http-address.mycluster.nn2</name>

<value>aj4:50070</value>

</property>

<property>

<name>dfs.namenode.shared.edits.dir</name>

<value>file:///mnt/filer</value>

</property>

<property>

<name>dfs.client.failover.proxy.provider.mycluster</name>

<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>

</property>

<property>

<name>dfs.ha.fencing.methods</name>

<value>sshfence</value>

</property>

<property>

<name>dfs.ha.fencing.ssh.private-key-files</name>

<value>/home/hduser/.ssh/id\_rsa</value>

</property>

<property>

<name>dfs.ha.fencing.methods</name>

<value>sshfence

shell(/bin/true)

</value>

</property>

</configuration>

=====================================

Hadoop Admin commands

hdfs namenode -initializeSharedEdits -force

hdfs hdfs namenode -bootstrapStandby -force

hdfs haadmin -failover --forceactive nn1 nn2

hdfs haadmin -transitionToActive nn1

hdfs haadmin -getServiceState nn1/nn2

===================================

To setup hadoop 2

First way:

mapred-site.xml.template

<property>

<name>mapred.job.tracker</name>

<value>aj4:9001</value>

</property>

<property>

<name>mapred.framework.name</name>

<value>yarn</value>

</property>

and in yarn-site.xml

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

Note\*\*https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/PluggableShuffleAndPluggableSort.html

Second way:

mapred-site.xml.template

<property>

<name>mapred.framework.name</name>

<value>yarn</value>

</property>

and in yarn-site.xml

<property>

<name>yarn.resourcemanager.address</name>

<value>aj4:9001</value>

</property>

kerberos

/etc/init.d/krb5kdc restart

for this to run install kerberos package

yum install krb5kdc-server

cat /etc/krb5.conf

update your domain name/host name

yum install krb5-libs

krb5-server

klist

kinit

================

/etc/init.d/krb5kdc restart

sudo apt-get install krb5kdc-server

cat /etc/krb5.conf

Execute kinit