# Setting up Hadoop, Zookeeper, HBase and Tomcat

1. Lunch Amazon Instance – Use Firefox ElasticFox.

ec2-public-images/fedora-8-x86\_64-base-v1.08.manifest.xml - ami-2547a34c

1. Setup /etc/hosts file
   1. 10.76.187.71 master
   2. 10.76.187.71 slave1
   3. ~~10.96.231.99 slave2~~
   4. ~~10.210.79.199 slave3~~
2. Setup iptables
   1. $ iptables -A INPUT -s 0.0.0.0 -j ACCEPT
   2. $ iptables -A INPUT -s master -j ACCEPT
   3. $ iptables -A INPUT -s slave1 -j ACCEPT
   4. ~~$ iptables -A INPUT -s slave2 -j ACCEPT~~
   5. ~~$ iptables -A INPUT -s slave3 -j ACCEPT~~
3. Setup the working shell

export PS1="[master \$PWD]# "

set -o vi

scp root@domU-12-31-39-0A-99-31.compute-1.internal:/mnt/sync/\* .

1. Setup auto login
   1. $ ssh-keygen -t rsa
   2. Generating public/private rsa key pair.
   3. Enter file in which to save the key (/home/jurn/.ssh/id\_rsa):
   4. Enter passphrase (empty for no passphrase):
   5. Enter same passphrase again:
   6. Your identification has been saved in /home/jurn/.ssh/id\_rsa
   7. Your public key has been saved in /home/jurn/.ssh/id\_rsa.pub
   8. $ cd ~/.ssh
   9. $ cat id\_rsa.pub >> authorized\_keys
   10. $ chmod 600 authorized\_keys
   11. $ cd ~
   12. $ ssh master
   13. $ exit
   14. $ ssh slave1
   15. $ exit
2. Setup JRE
   1. $ wget <http://www.bizosys.com/hsearch/jre.bin>
   2. $ cd /mnt
   3. $ chmod 700 ~/jre.bin
   4. $ ~/jre.bin
   5. $ mv jre1.6.0\_23 jdk1.6.0\_23
   6. $ mkdir /usr/java
   7. $ ln -s /mnt/jdk1.6.0\_23/ /usr/java
   8. $ ln -s /mnt/jdk1.6.0\_23/ /usr/lib/jdk
   9. $ ln -s /mnt/jdk1.6.0\_23/ /usr/java/jdk1.6.0
3. Setup Hadoop
   1. $ wget <http://www.poolsaboveground.com/apache//hadoop/common/hadoop-0.20.2/hadoop-0.20.2.tar.gz>
   2. $ gzip -d hadoop-0.20.2.tar.gz
   3. $ tar -xf hadoop-0.20.2.tar
   4. $ mv hadoop-0.20.2.tar hadoop-0.20.2
   5. $ mv hadoop-0.20.2 hadoop
   6. $ cd hadoop
   7. $ rm -rf hadoop-0.20.2-examples.jar hadoop-0.20.2-test.jar hadoop-0.20.2-ant.jar c++ contrib docs ivy ivy.xml src
   8. $ ls
   9. $ cd conf
   10. **Attach core-site.xml**
   11. $ echo "" > excludes
   12. $ mkdir -p /mnt/data/namenode /mnt/data/datanode /mnt/logs /mnt/data/namenode/dfsname
   13. Export variables to the end of the hadoop configuration file.

$ echo "export JAVA\_HOME=/usr/lib/jdk" >> hadoop-env.sh

$ echo "export HADOOP\_HEAPSIZE=2048" >> hadoop-env.sh

$ echo "export HADOOP\_OPTS=\"-server -XX:+UseParallelGC -XX:ParallelGCThreads=4 -XX:+AggressiveHeap -XX:+HeapDumpOnOutOfMemoryError\"" >> hadoop-env.sh

$ echo "export HADOOP\_LOG\_DIR=/mnt/logs" >> hadoop-env.sh

* 1. **Attach hdfs.site file here.**
  2. **Attach log4j.properties file here.**
  3. $ echo "master" > masters; cat masters
  4. $ echo "slave1" > slaves; cat slaves
  5. $ cd /mnt/hadoop/bin/
  6. **$ ./hadoop --config /mnt/hadoop/conf namenode -format**

1. Start Hadoop
   1. $ cd /mnt/hadoop/bin/
   2. $ ulimit -n 16384
   3. $ ./hadoop-daemon.sh --config /mnt/hadoop/conf/ start namenode; tail -f /mnt/logs/hadoop-root-namenode-\*.log

Sucessful means it should read at the end line *IPC Server handler 39 on 54310: starting*

* 1. $ ./hadoop-daemon.sh --config /mnt/hadoop/conf/ start datanode; tail -f /mnt/logs/hadoop-root-datanode-\*.log

1. Test Hadoop
   1. $ cd /mnt/hadoop/bin/
   2. $ ./hadoop dfs -du /
   3. Upload and Download a File

$ cat > /tmp/a.txt

* 1. $ ./hadoop dfs -copyFromLocal /tmp/a.txt /
  2. $ ./hadoop dfs -ls /
  3. $ ./hadoop dfs -cat /a.txt
  4. $ ./hadoop dfs -rm /a.txt
  5. $ ./hadoop dfs -ls /

1. Stop Hadoop
   1. $ cd /mnt/hadoop/bin/
   2. $ ./hadoop-daemon.sh --config /mnt/hadoop/conf/ stop datanode; tail -f /mnt/logs/hadoop-root-datanode-\*.log
   3. $ ./hadoop-daemon.sh --config /mnt/hadoop/conf/ stop namenode; tail -f /mnt/logs/hadoop-root-namenode-\*.log
2. Setup Zookeeper
   1. $ cd /mnt
   2. $ wget http://www.poolsaboveground.com/apache//zookeeper/zookeeper-3.3.3/zookeeper-3.3.3.tar.gz
   3. $ gzip -d zookeeper-3.3.3.tar.gz
   4. $ tar -xf zookeeper-3.3.3.tar
   5. $ mv zookeeper-3.3.3.tar zookeeper-3.3.3
   6. $ mv zookeeper-3.3.3 zookeeper
   7. $ cd zookeeper
   8. $ rm -rf docs ivy.xml ivysettings.xml src contrib dist-maven; ls
   9. $ cd conf
   10. $ cp zoo\_sample.cfg zoo.cfg
   11. $ vi zoo.cfg
   12. $ modify the line “dataDir=/mnt/data/zoodata”
   13. Zoo Cluster setup can be done by adding following lines where zookeeper is running in master, slave1 and slave2

server.1=master:2888:3888

server.2=slave1:2888:3888

server.3=slave2:2888:3888

* 1. $ echo "export ZOO\_LOG\_DIR=/mnt/logs" >> /mnt/zookeeper/bin/zkEnv.sh
  2. $ mkdir /mnt/data/zoodata

1. Start Zookeeper
   1. $ cd /mnt/zookeeper/bin
   2. $ export JAVA\_HOME=/usr/lib/jdk
   3. $ export PATH=$PATH:$JAVA\_HOME/bin
   4. $ /mnt/zookeeper/bin/zkServer.sh start > /mnt/logs/zoo.cfg ; tail -f /mnt/logs/zoo.cfg
2. Test Zookeeper
   1. $ ps -ef | grep zoo
3. Stop Zookeeper
   1. $ /mnt/zookeeper/bin/zkServer.sh stop
   2. $ ps -ef | grep zoo
4. Setup HBase
   1. $ wget http://www.takeyellow.com/apachemirror//hbase/hbase-0.90.4/hbase-0.90.4.tar.gz
   2. $ gzip -d hbase-0.90.4.tar.gz
   3. $ tar -xf hbase-0.90.4.tar
   4. $ mv hbase-0.90.4 hbase
   5. $ mv hbase-0.90.4.tar hbase
   6. $ cd hbase
   7. $ rm -rf docs hbase-0.90.4-tests.jar pom.xml src
   8. $ cd conf
   9. $ echo "slave1" > regionservers; cat regionservers
   10. log4j.properties file edit.

hbase.log.dir=/mnt/logs

* 1. Export variables to the end of the hbase configuration file.

$ echo "export JAVA\_HOME=/usr/lib/jdk" >> hbase-env.sh

$ echo "export HBASE\_HEAPSIZE=2048" >> hbase-env.sh

$ echo "export HBASE\_OPTS=\"-server -XX:+UseParallelGC -XX:ParallelGCThreads=4 -XX:+AggressiveHeap -XX:+HeapDumpOnOutOfMemoryError\"" >> hbase-env.sh

$ echo "export HBASE\_LOG\_DIR=/mnt/logs" >> hbase-env.sh

* 1. Setup following properties in hbase-site.xml file

*<property>*

*<name>hbase.cluster.distributed</name>*

*<value>true</value>*

*</property>*

*<property>*

*<name>hbase.rootdir</name>*

*<value>hdfs://master:54310/hbase</value>*

*</property>*

*<property>*

*<name>hbase.zookeeper.quorum</name>*

*<value>master~~,slave1,slave2~~</value>*

*</property>*

* 1. $ cd /mnt/hadoop; mv hadoop-0.20.2-core.jar hadoop-0.20.2-core.jar.orig
  2. $ cp /mnt/hbase/lib/hadoop-core-0.20-append-r1056497.jar hadoop-0.20.2-core.jar

1. Start HBase
   1. $ cd /mnt/hbase/bin/
   2. $ ./hbase-daemon.sh --config /mnt/hbase/conf start master; tail -f /mnt/logs/hbase-root-master\*.log
   3. $ ./hbase-daemons.sh --config /mnt/hbase/conf start regionserver; tail -f /mnt/logs/hbase-root-regionserver\*.log
2. Test HBase
   1. $ ./hbase shell

list

1. Stop HBase
   1. $ cd /mnt/hbase/bin/
   2. $ ./hbase-daemons.sh --config /mnt/hbase/conf stop regionserver
   3. $ ./hbase-daemon.sh --config /mnt/hbase/conf stop master
2. Setup Tomcat
   1. $ cd /mnt
   2. $ wget http://mirrors.ibiblio.org/apache/tomcat/tomcat-7/v7.0.22/bin/apache-tomcat-7.0.22.tar.gz
   3. $ gzip -d apache-tomcat-7.0.22.tar.gz
   4. $ tar -xf apache-tomcat-7.0.22.tar
   5. $ mv apache-tomcat-7.0.22.tar apache-tomcat-7.0.22
   6. $ mv apache-tomcat-7.0.22 tomcat
   7. $ cd tomcat
   8. $ cd conf
   9. $ mv server.xml server.xml.orig; vi server.xml
   10. $ mv logging.properties logging.properties.orig
   11. $ sed 's/\${catalina.base}/\/mnt/g' logging.properties.orig > logging.properties
   12. Replace tomcat-users with

<?xml version='1.0' encoding='utf-8'?>

<tomcat-users>

</tomcat-users>

* 1. $ rm –rf /mnt/tomcat/webapps/\*
  2. $ cd /mnt/tomcat/webapps
  3. $ mkdir ROOT
  4. $ cd ROOT
  5. $ cat > index.html
  6. Deploy the hsearch code
     1. $ cd /mnt/hbase/lib; wget <http://www.bizosys.com//download/hsearch-0.94-termfilter.jar>
     2. Restart the hbase
     3. $ cd /mnt/tomcat/webapps
     4. $ wget http://www.bizosys.com/download/hsearch-0.94.war

1. Start Tomcat
   1. $ export JAVA\_HOME=/usr/lib/jdk
   2. $ export PATH=$PATH:$JAVA\_HOME/bin
   3. $ cd /mnt; /mnt/tomcat/bin/startup.sh ; tail -f /mnt/logs/localhost\*.log
2. Test Tomcat
   1. Browse http://ec2-107-22-40-37.compute-1.amazonaws.com/index.html
3. Stop Tomcat
   1. $ /mnt/tomcat/bin/shutdown.sh; sleep 5; ps -ef | grep catalina

# Maintenance Setup

1. $ export JAVA\_HOME=/usr/lib/jdk
2. $ export PATH=$PATH:$JAVA\_HOME/bin
3. $

for file in `ls /mnt/tomcat/webapps/ROOT/WEB-INF/lib/`;

do

export CLASSPATH=$CLASSPATH:/mnt/tomcat/webapps/ROOT/WEB-INF/lib/$file;

done

export CLASSPATH=$CLASSPATH:hmaintenance-0.1-core.jar;/mnt/hadoop/conf

java -cp $CLASSPATH com.red5.AdminConsole table=content mode=download

du --max-depth 1 hbase

# java –cp $CLASSPATH com.red5.HadoopBackup /hbase . 0 1024

# HBase Backup

$ java com.bizosys.oneline.maintenance.HBaseBackup mode=backup.incremental backup.folder=s3://hsearchbackup/ tables=0,1,2,3,4,5,6,7,8,9,a,b,c,config,content,d,dictionary,e,f,g,h,i,idmap,j,k,l,m,n,o,p,preview,q,,r,s,t,z,\_ duration=30

$ java com.bizosys.oneline.maintenance.HBaseBackup mode=backup.full backup.folder=s3://hsearchbackup/ tables=0,1,2,3,4,5,6,7,8,9,a,b,c,config,content,d,dictionary,e,f,g,h,i,idmap,j,k,l,m,n,o,p,preview,q,,r,s,t,z,\_

$ java com.bizosys.oneline.maintenance. HBaseBackup mode=restore backup.folder=s3://hsearchbackup/Sun\_Dec\_04\_19\_54\_01\_IST\_2011.incr tables=0,1,2,3,4,5,6,7,8,9,a,b,c,config,content,d,dictionary,e,f,g,h,i,idmap,j,k,l,m,n,o,p,preview,q,,r,s,t,z,\_

java com.bizosys.oneline.maintenance. HBackup mode=backup.history backup.folder=s3://hsearchbackup

java com.bizosys.oneline.maintenance. HBackup mode=restore backup.folder=s3://hsearchbackup/Sun\_Dec\_04\_16\_15\_33\_IST\_2011 tables=content

# Slaves Setup

**Maintain all slaves and master at one place**

|  |  |  |  |
| --- | --- | --- | --- |
| Server | Internal IP | Internal Name | External Name |
| Master | 10.211.154.191 | domU-12-31-39-0A-99-31.compute-1.internal | ec2-184-73-65-245.compute-1.amazonaws.com |
| slave1 |  |  |  |
| slave2 |  |  |  |
| Slave3 |  |  |  |
| Slave4 |  |  |  |
| Slave5 |  |  |  |
| Slave6 |  |  |  |
| Slave7 |  |  |  |

**Check the Disk Usage across slaves**

for server in slave1 slave2 slave3 slave4 slave5 slave6 slave7 slave8 slave9 slave10

do

ssh $server du --max-depth 1 /mnt/data/

done

**Execute Command across slaves**

for server in master slave1

do

ssh $server echo $server ; grep exception /mnt/logs/\*.log ;

done

**Nutch**

1. $ cd /mnt/nutch/runtime/local
2. $ export JAVA\_HOME=/usr/lib/jdk
3. $ export PATH=$PATH:$JAVA\_HOME/bin

bin/nutch crawl urls -dir crawl -depth 3 -topN 100

bin/nutch readdb crawl/crawldb -dump /tmp/entiredump

bin/nutch invertlinks crawl/linkdb/ crawl/segments/20111216025534 crawl/segments/20111216030000 crawl/segments/20111216030354

bin/nutch readlinkdb crawl/linkdb -dump /tmp/linksdir

**Configuration Files**

# export JAVA\_HOME=/usr/lib/jdk

# export PATH=$PATH:$JAVA\_HOME/bin

# export HBASE\_HOME=/mnt/hbase/

#

for file in `ls $HBASE\_HOME/lib`;

do export CLASSPATH=$CLASSPATH:$HBASE\_HOME/lib/$file;

done

export CLASSPATH=$CLASSPATH:$HBASE\_HOME/conf;

for file in `ls $HBASE\_HOME/\*.jar`;

do export CLASSPATH=$CLASSPATH:$file;

done

export CLASSPATH=$CLASSPATH:./hmaintenance-0.1-core.jar

export CLASSPATH=$CLASSPATH:./hmaintenance-0.1-core.jar

export CLASSPATH=$CLASSPATH:/mnt/tomcat/webapps/ROOT/WEB-INF/lib/xstream.jar:/mnt/tomcat/webapps/ROOT/WEB-INF/lib/xpp.jar

**Kid Search Engine**

## 5D74C6768EFBEC9C2CCD52607F47E6DD

1. $ export JAVA\_HOME=/usr/lib/jdk
2. $ export PATH=$PATH:$JAVA\_HOME/bin
3. $

for file in `ls /mnt/tomcat/webapps/ROOT/WEB-INF/lib/`;

do

export CLASSPATH=$CLASSPATH:/mnt/tomcat/webapps/ROOT/WEB-INF/lib/$file;

done

export CLASSPATH=$CLASSPATH:/mnt/download/data:/mnt/tomcat/webapps/ROOT/WEB-INF/classes

export CLASSPATH=$CLASSPATH:/mnt/tomcat/webapps/ROOT/WEB-INF/classes/services/service-search/service-search.jar

java -cp $CLASSPATH HSearchLoad apikey=5D74C6768EFBEC9C2CCD52607F47E6DD document.url="file:///mnt/download/data/videos.tsf" document.type=video id.column=0 columns.separator=tab linebreak=[-] columns.format=0,0,0,0,0,0,0,0,0,0,0,0,0 columns.nonempty=0 weight.column=13 columns.title=4 url.column=5 columns.desc=7 keyword.column=6 index.start=1 columns.indexable=1,2,8,11 index.batch.size=150 index.runplan=FilterDuplicateId,TokenizeStandard,FilterStopwords,FilterTermLength,FilterLowercase,FilterStem,RemoveNonAscii,NormalizeAccents,ComputeTokens,SaveToDictionary,SaveToIndex,SaveToPreview,SaveToContent & > /mnt/logs/fileload.log 2>&1