Pre-condition

1. SSH 無密碼登入 hadoop 利用ssh 去執行其他slave
2. modify hostname, hadoop 利用hostname連線 用ip 會有問題

sudo vim /etc/hostname

sudo vim /etc/hosts

1. Install Java **JDK, not JRE**

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

sudo vim /etc/profile

export JAVA\_HOME=/usr/lib/java/$JAVA\_DIR

export PATH=$JHAVA\_HOME/bin:$PATH

1. Install Zookeeper

<http://apache.stu.edu.tw/zookeeper/>

sudo vim /etc/profile

export ZOOKEEPER\_HOME=/home/hadoopuser/zookeeper/zookeeper-3.4.6

export PATH=$PATH:$ZOOKEEPER\_HOME/bin:$ZOOKEEPER\_HOME/conf

then cd zookeeper-3.4.6/conf

cp zoo\_sample.cfg zoo.cfg

**In zoo.cfg**

**dataDir**=/home/hadoopuser/zookeeper/zookeeper-3.4.6/data

# the port at which the clients will connect

clientPort=2181

server.1=master:2888:3888

server.2=slave1:2888:3888

server.3=slave2:2888:3888

In dataDir, create a myid text，文件中的内容只有一行，为本主机对应的id值，也就是上图中server.id中的id。例如：在服务器1中的myid的内容应该写入1

* SSH 無密碼登入

ssh-keygen -t rsa

# 一直按回车就可以，生成的密钥保存为.ssh/id\_rsa

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

test: ssh hadoopuser@master

copy public key to each slave

scp ~/.ssh/id\_rsa.pub hadoopuser@slave1:/home/hadoopuser/.ssh/aauthorize\_keys

or cat ~/.ssh/id\_rsa.pub | ssh hadoopuser@slave1 "cat >> /home/hadoopuser/.ssh/authorized\_keys"

test: ssh hadoopuser@slave1

scp ~/.ssh/id\_rsa.pub hadoopuser@slave2:/home/hadoopuser/.ssh/authorized\_keys

test: ssh hadoopuser@slave2

Next copy zookeeper to slaves

scp -r zookeeper hadoopuser@slave1:~/zookeeper

scp -r zookeeper hadoopuser@slave2:~/zookeeper

modify myid。例如修改slave1中的myid如下：

hadoopuser@slave1:~/zookeeper/zookeeper-3.4.6$ echo "2" > data/myid

hadoopuser@slave2:~/zookeeper/zookeeper-3.4.6$ echo "3" > data/myid

安装daemontools

su - root

mkdir /package

chmod 755 /package

cd /package

wget http://cr.yp.to/daemontools/daemontools-0.76.tar.gz

tar zxf daemontools-0.76.tar.gz

cd admin/daemontools-0.76

vim src/error.h 找到：extern int errno; 改成：#include <errno.h>

package/install

## 监控Zookeeper

在/service下，新建文件夹zookeeper，新建run文件：

cd /service

mkdir zookeeper

cd zookeeper

vim run

chmod 755 run

run内容：

#!/bin/bash

exec 2>&1

exec /zk/zookeeper-3.4.4/bin/zkServer.sh start

运行：

supervise /service/zookeeper

//或者可以用nohup以后台方式运行。如下：

**sudo nohup supervise /service/zookeeper &**

如果出现

supervise: fatal: unable to acquire /service/zookeeper/supervise/lock: temporary failure

快速rm /service/zookeeper/supervise 重新执行 supervise /service/zookeeper 即可

验证监控zookeeper是否成功：

kill zookeeper进程，查看zookeeper的进程是否自动重启

ps -aux|grep Dzookeeper //查看zookeeper的进程

svc command

<http://cr.yp.to/daemontools/svc.html>

* -u: Up. If the service is not running, start it. If the service stops, restart it.
* -d: Down. If the service is running, send it a TERM signal and then a CONT signal. After it stops, do not restart it.
* -o: Once. If the service is not running, start it. Do not restart it if it stops.
* -p: Pause. Send the service a STOP signal.
* -c: Continue. Send the service a CONT signal.
* -h: Hangup. Send the service a HUP signal.
* -a: Alarm. Send the service an ALRM signal.
* -i: Interrupt. Send the service an INT signal.
* -t: Terminate. Send the service a TERM signal.
* -k: Kill. Send the service a KILL signal.
* -x: Exit. supervise will exit as soon as the service is down. If you use this option on a stable system, you're doing something wrong; supervise is designed to run forever.

sudo jps, see if zookeeper exist in each node

QuorumPeerMain

zookeeper status

command: zkServer status

Hadoop Env

hadoop-env.sh

export JAVA\_HOME=/home/hadoopuser/jdk1.8.0\_31

* Install lzo compression

-----------------------------------------------------------------------

In lzo-2.08 build code

wget http://www.oberhumer.com/opensource/lzo/download/lzo-2.08.tar.gz

export CFLAGS=-m64

./configure -enable-shared -prefix=/home/hadoopuser/hadoop/hadoop-2.6.0/lzo/

make

sudo make install

cp /home/hadoopuser/hadoop/hadoop-2.6.0/lzo/lib/\* /usr/lib/

cp /home/hadoopuser/hadoop/hadoop-2.6.0/lzo/lib/\* /usr/lib64/

cp -r /home/hadoopuser/hadoop/hadoop-2.6.0/lzo/include/\* /usr/include/

----------------------------------------------------------------------

In lzop

wget http://www.lzop.org/download/lzop-1.03.tar.gz

tar -zxvf lzop-1.03.tar.gz

cd lzop-1.03

./configure -enable-shared -prefix=/home/hadoopuser/hadoop/hadoop-2.6.0/lzop/

make && make install

cd /usr/bin

ln -s -f /home/hadoopuser/hadoop/hadoop-2.6.0/lzop/bin/lzop

Test bin, lzop should can compress

lzop <file>

--------------------------------------------------------------------------

In twitter hadoop

git clone https://github.com/twitter/hadoop-lzo.git

可以修改pom.xml来调整自己的hadoop版本, 找到hadoop.current.version配置项进行修改

export CFLAGS=-m64

export CXXFLAGS=-m64

export C\_INCLUDE\_PATH=/home/hadoopuser/hadoop/hadoop-2.6.0/lzo/include

export LIBRARY\_PATH=/home/hadoopuser/hadoop/hadoop-2.6.0/lzo/lib

mvn clean package -Dmaven.test.skip=true

cp -r target/native/Linux-amd64-64 /home/hadoopuser/hadoop/hadoop-2.6.0/lib/native/

cp target/hadoop-lzo-0.4.20-SNAPSHOT.jar /home/hadoopuser/hadoop/hadoop-2.6.0/share/hadoop/common/

* PS If use SPARK

SPARK\_LIBRARY\_PATH=$SPARK\_LIBRARY\_PATH:/path/to/your/hadoop-lzo/libs/native

SPARK\_CLASSPATH=$SPARK\_CLASSPATH:/path/to/your/hadoop-lzo/java/libs

hadoop-env.sh

|  |  |
| --- | --- |
| 1  2  3 | # 添加如下配置项  export JAVA\_LIBRARY\_PATH=$HADOOP\_HOME/lib/native/Linux-amd64-64  export LD\_LIBRARY\_PATH=$HADOOP\_HOME/lzo/lib |

core-site.xml

<property>

<name>io.compression.codecs</name>

<value>org.apache.hadoop.io.compress.GzipCodec,org.apache.hadoop.io.compress.DefaultCodec,com.hadoop.compression.lzo.LzoCodec,com.hadoop.compression.lzo.LzopCodec,org.apache.hadoop.io.compress.BZip2Codec</value>

</property>

<property>

<name>io.compression.codec.lzo.class</name>

<value>com.hadoop.compression.lzo.LzoCodec</value>

</property>

mapred-site.xml

<property>

<name>mapred.compress.map.output</name>

<value>true</value>

</property>

<property>

<name>mapred.map.output.compression.codec</name>

<value>com.hadoop.compression.lzo.LzoCodec</value>

</property>

<property>

<name>mapred.child.env</name>

<value>LD\_LIBRARY\_PATH=/usr/local/cloud/hadoop/lzo/lib</value>

</property>

* first launch

In each node : launch journal node in below command

hadoop-daemon.sh start journalnode

hdfs namenode –format => active node => first use namenode need format

hdfs namenode –bootstrapStandby = > standby node => need to point standby node

for zookeeper = > hdfs zkfc –formatZK

then start-dfs.sh

then you can see two namenode in cluster, stand for active and standby

then you can see two ZKController in cluster

then slave can see data node

You can also type bellow commands to launch each node

hadoop-daemon.sh start namenode

hadoop-daemon.sh start datanode

* check namenode HA status

**hdfs haadmin -getServiceState nn1**

**hdfs haadmin -getServiceState nn2**

start-yarn.sh for two nodes, active node, standby node

* check RM HA status

**yarn rmadmin -getServiceState rm1**

**yarn rmadmin -getServiceState rm2**

hadoop command

http://debugo.com/hdfs-fscmd/

hdfs dfs –df –h

hadoop fs -count -h -q /user/hadoopuser/warehouse

QUOTA REMAINING\_QUOTA SPACE\_QUOTA REMAINING\_SPACE\_QUOTA

none inf 500 G 496.0 G

DIR\_COUNT FILE\_COUNT CONTENT\_SIZE FILE\_NAME

2 3 1.3 G hdfs://master:54310/path/

Q＆A

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