

yangc-blog www.sysopen.cn

qq 群 : 521621407

Hadoop、hbase、zookeeper、chukwa 集群配置

一、 环境配置 :

1、 Hadoop、hbase、zookeeper、chukwa 安装指定地址下载的软件 :

Hadoop2.7.1 : [http://192.168.5.30:8080/job/build-omms\(win\)/16/artifact/Hadoop/hadoop/](http://192.168.5.30:8080/job/build-omms(win)/16/artifact/Hadoop/hadoop/)

Hbase1.1.2 : [http://192.168.5.30:8080/job/build-omms\(win\)/16/artifact/Hadoop/hbase/](http://192.168.5.30:8080/job/build-omms(win)/16/artifact/Hadoop/hbase/)

Zookeeper3.4.6 : [http://192.168.5.30:8080/job/build-omms\(win\)/16/artifact/Hadoop/zookeeper/](http://192.168.5.30:8080/job/build-omms(win)/16/artifact/Hadoop/zookeeper/)

Chukwa0.6.0 : [http://192.168.5.30:8080/job/build-omms\(win\)/16/artifact/Hadoop/chukwa/](http://192.168.5.30:8080/job/build-omms(win)/16/artifact/Hadoop/chukwa/)

2、 部署 PATH :

运行环境需要 Java 环境 , 先安装 jdk-7u60-linux-x64.tar.gz , 解压之后重命名为 java ,

放在/usr/local/目录下 ;

解压 Hadoop (版本 2.7) 、 hbase、zookeeper、chukwa 压缩文件 , 放在/usr/local/目录下 ;

自定义 PATH 环境 : vim /etc/profile

```
export JAVA_HOME=/usr/local/java
```

```
export CLASSPATH=$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
```

```
export HADOOP_DEV_HOME=/usr/local/hadoop
```

```
export HADOOP_MAPRED_HOME=${HADOOP_DEV_HOME}
```

```
export HADOOP_COMMON_HOME=${HADOOP_DEV_HOME}
```

```
export HADOOP_HDFS_HOME=${HADOOP_DEV_HOME}
```

```

export YARN_HOME=${HADOOP_DEV_HOME}
export HADOOP_CONF_DIR=${HADOOP_DEV_HOME}/etc/hadoop
export HDFS_CONF_DIR=${HADOOP_DEV_HOME}/etc/hadoop
export YARN_CONF_DIR=${HADOOP_DEV_HOME}/etc/hadoop
export ZOOKEEPER_HOME=/usr/local/zookeeper
export HBASE_HOME=/usr/local/hbase
export HBASE_CONF_DIR=${HBASE_HOME}/conf
export CHUKWA_HOME=/usr/local/chukwa
export CHUKWA_CONF_DIR=${CHUKWA_HOME}/etc/chukwa
export
PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$HADOOP_DEV_HOME/bin:
$HADOOP_DEV_HOME/sbin:$ZOOKEEPER_HOME/bin:$ZOOKEEPER_HOME/conf:
$HBASE_HOME/bin:$CHUKWA_HOME/bin:$CHUKWA_HOME/sbin:$PATH

```

source /etc/profile , 是自定义的环境生效

3、修改/etc/hosts , 本地无 DNS , 在 hosts 中相互绑定 , 最好 /etc/sysconfig/network 中

的 hostname 也保持一致 ;

192.168.5.201 master

192.168.5.202 node1

192.168.5.203 node2

4、实现 master 到 node 节点 ssh 无密码登录 :

在 master 上使用 ssh-keygen -t rsa 生产相应的公钥和私钥 , 进入 ~/.ssh 目录 , cp id_rsa.pub authorized_keys , 然后将 authorized_keys 通过 scp 发送给节点服务器 , 存放在相同的目录下。

二、 Hadoop 部署 :

1、配置文件 : 修改 hadoop 的 [配置文件](#) , 即各种 site 文件 , 文件存放

在 /usr/local/hadoop/etc/hadoop/ 下 , 主要配置 core-site.xml 、 hdfs-site.xml 、 mapred-site.xml 、 yarn-site.xml , 环境变量设置 hadoop-env.sh 、 mapred-env.sh 、 yarn-env.sh 这几个文件中的 JAVA_HOME 改为 /usr/local/java/。

core-site.xml 配置 :

```

<configuration>
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://master:9000</value>
</property>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/hadoop/tmp</value>
</property>
<property>
  <name>fs.file.impl</name>
  <value>org.apache.hadoop.fs.LocalFileSystem</value>
  <description>The FileSystem for file: uris.</description>
</property>

<property>
  <name>fs.hdfs.impl</name>
  <value>org.apache.hadoop.hdfs.DistributedFileSystem</value>
  <description>The FileSystem for hdfs: uris.</description>
</property>

</configuration>

```

hdfs-site.xml 配置 :

```

<configuration>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>/hadoop/dfs/name</value>
  <description> </description>
</property>

<property>
  <name>dfs.datanode.data.dir</name>
  <value>/hadoop/dfs/data</value>
  <description> </description>
</property>

<property>
  <name>dfs.replication</name>
  <value>2</value>
</property>

<property>

```

```
<name>dfs.namenode.secondary.http-address</name>
<value>master:50090</value>
</property>

<property>
<name>dfs.webhdfs.enabled</name>
<value>true</value>
</property>
</configuration>
```

mapred-site.xml 配置 :

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
<property>
<name>mapreduce.jobhistory.address</name>
<value>master:10020</value>
</property>

<property>
<name>mapreduce.jobhistory.webapp.address</name>
<value>master:19888</value>
</property>

</configuration>
```

yarn-site.xml 配置 :

```
<configuration>

<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>

<property>
<name>yarn.nodemanager.recovery.enabled</name>
<value>true</value>
</property>

<property>
```

```

        <description>The address of the applications manager interface in the
RM.</description>
        <name>yarn.resourcemanager.address</name>
        <value>master:9030</value>
    </property>

    <property>
        <description>The address of the scheduler interface.</description>
        <name>yarn.resourcemanager.scheduler.address</name>
        <value>master:9031</value>
    </property>

    <property>
        <description>The address of the RM web application.</description>
        <name>yarn.resourcemanager.webapp.address</name>
        <value>master:9033</value>
    </property>

    <property>
        <description>The address of the resource tracker interface.</description>
        <name>yarn.resourcemanager.resource-tracker.address</name>
        <value>master:9035</value>
    </property>

    <property>
        <name>yarn.nodemanager.resource.memory-mb</name>
        <value>1572</value>
    </property>

    <property>
        <description>address of node manager IPC.</description>
        <name>yarn.nodemanager.address</name>
        <value>0.0.0.0:3999</value>
    </property>

</configuration>

```

2、集群配置

slaves 文件中添加 node 节点主机名或者对应 IP

node1、 node2

3、启动 Hadoop :

用 `scp -r` 将 master 上的配置好的 Hadoop 同步到 node 节点上，并格式化 namenode，在 master 节点上执行如下命令：`/usr/local/hadoop/bin/hdfs namenode -format`，

```
15/07/17 19:17:45 INFO util.GSet: VM type = 64-bit
15/07/17 19:17:45 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
15/07/17 19:17:45 INFO util.GSet: capacity = 2^18 = 262144 entries
15/07/17 19:17:45 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
15/07/17 19:17:45 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
15/07/17 19:17:45 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension = 30000
15/07/17 19:17:45 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
15/07/17 19:17:45 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
15/07/17 19:17:45 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
15/07/17 19:17:45 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
15/07/17 19:17:45 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
15/07/17 19:17:45 INFO util.GSet: Computing capacity for map NameNodeRetryCache
15/07/17 19:17:45 INFO util.GSet: VM type = 64-bit
15/07/17 19:17:45 INFO util.GSet: 0.0299999999328447746% max memory 889 MB = 273.1 MB
15/07/17 19:17:45 INFO util.GSet: capacity = 2^16 = 32768 entries
Re-format filesystem in Storage Directory /tmp/hadoop-root/dfs/name ? (Y or N) Y
15/07/17 19:17:47 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1209387458-192.168.15.135-1437131267391
15/07/17 19:17:47 INFO common.Storage: Storage directory /tmp/hadoop-root/dfs/name has been successfully formatted.
15/07/17 19:17:47 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
15/07/17 19:17:47 INFO util.ExitUtil: Exiting with status 0
15/07/17 19:17:47 INFO namenode.NameNode: SHUTDOWN_MSG:
/*
SHUTDOWN_MSG: Shutting down NameNode at master/192.168.15.135
*/
[root@master hadoop-2.7.1]#
```

看到 successfully formatted，格式化成功。

使用 `/usr/local/hadoop/sbin/start-dfs.sh` 和 `/usr/local/hadoop/sbin/start-yarn.sh` 启动

Hadoop，在 master 上通过 `jps` 检查相关服务是否正常启动：

```
[root@master hadoop-2.7.1]# jps
12707 NameNode
13219 ResourceManager
12999 SecondaryNameNode
13503 Jps
[root@master hadoop-2.7.1]#
```

三、 zookeeper 部署：

- 1、进入 `/usr/local/zookeeper/config` 目录中，将 `zoo_sample.cfg` 拷贝一份命名为 `zoo.cfg`，

修改配置：

```
tickTime=2000
dataDir=/var/lib/zookeeper/
clientPort=2181
initLimit=5
syncLimit=2
server.1=master:2888:3888
server.2=node1:2888:3888
```

server.3=node2:2888:3888

- 2、创建 myid : 在 zoo.cfg 配置文件中的 dataDir 的目录下面创建 myid , 每个节点 myid 要求不一样 :

master : echo 1 > /var/lib/zookeeper/myid

node1: echo 2> /var/lib/zookeeper/myid

node2: echo 3 > /var/lib/zookeeper/myid

四、 hbase 部署 :

- 1、 配置文件 : hbase-env.sh、 hbase-site.xml、 regionservers 三个文件的配置

修改 hbase-env.sh 的 JAVA_HOME=/usr/local/java

hbase-site.xml 配置 :

```
<configuration>
```

```
<property>
```

```
<name>hbase.rootdir</name>
```

```
<value>hdfs://192.168.5.201:9000/hbase</value>
```

```
<description>hbase data directory</description>
```

```
</property>
```

```
<property>
```

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

```
<value>true</value>
```

```
</property>
```

```
<property>
```

```
<name>hbase.master</name>
```

```
<value>192.168.5.201:6000</value>
```

```
</property>
```

```
<property>
```

```
<name>dfs.replication</name>
```

```
<value>2</value>
```

```
</property>
```

```
<property>
```

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

```
<value>master,node1,node2</value>
```

```
</property>
```

```

    <property>
    <name>hbase.zookeeper.property.clientPort</name>
    <value>2181</value>
    <description>Property from ZooKeeper's config zoo.cfg. The port at which the clients
will connect.</description>
    </property>

    <property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>/usr/local/zookeeper/data</value>
    <description>Property from ZooKeeper's config zoo.cfg. The directory where the
snapshot is stored.</description>
    </property>

</configuration>

```

regionservers 配置：

node1

node2

regionservers 里面是节点服务器。

五、chukwa 部署：

- 1、准备工作：首先把 hadoop，停用，把配置目录中的 log4j.properties 和 hadoop-metrics2.properties 文件改名备份，然后把 chukwa 配置目录中的 log4j.properties 和 hadoop-metrics2.properties 文件复制到 hadoop 配置目录中；把 chukwa 中的 chukwa-0.6.0-client.jar 和 json-simple-1.1.jar 两个 jar 文件复制到 hadoop 中 lib 目录下；修改 chukwa-config.sh 的 JAVA_HOME=/usr/local/java；
- 2、将下载的 output 目录下的文件拷贝至 Hadoop\chukwa\share\chukwa 目录下，将原有的备份。
- 3、配置：

agents：默认为本机 localhost

chukwa-agent-conf.xml：已配置好，不用修改

chukwa-collector-conf.xml : 将 IP 地址修改为 master 的 IP

```
<property>
  <name>writer.hdfs.filesystem</name>
  <value>hdfs://192.168.5.201:9000</value>
  <description>HDFS to dump to</description>
</property>
```

六、 启动集群：

在启动之前，使用 `scp -r` 将 zookeeper、hbase、chkwa 发送到 node1、node2 的相同目录

启动顺序依次：Hadoop、zookeeper、hbase、chukwa

Hadoop 的启动与停止：`/usr/local/hadoop/sbin/{start-all.sh(start-dfs.sh,start-yarn.sh),stop-all.sh(stop-dfs.sh,stop-yarn.sh)}`

启动 thrift2 服务：`/usr/local/hbase/bin/hbase-daemon.sh start thrift2`

Zookeeper 的启动与停止：`/usr/local/zookeeper/bin/{zkServer.sh start、stop}`

Hbase 启动与停止 `/usr/local/hbase/bin/{start-hbase.sh,stop-hbase.sh}`，在启动 hbase 后

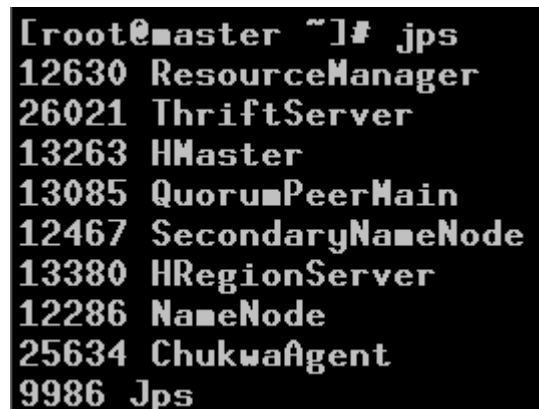
使用 hbase shell 创建两个数据库：

`create 't_FMServer_info','FMServer_info_CF'`

Chukwa 启动与停止：`/usr/local/chukwa/bin/chukwa`

`agent , /usr/local/chukwa/bin/chukwa agent stop`

所有服务启动之后再 master 上输入：`jps`



```
[root@master ~]# jps
12630 ResourceManager
26021 ThriftServer
13263 HMaster
13085 QuorumPeerMain
12467 SecondaryNameNode
13380 HRegionServer
12286 NameNode
25634 ChukwaAgent
9986 Jps
```

node2：`jps`

```
[root@node2 ~]# jps
12104 DataNode
24055 Jps
16702 ChukwaAgent
12191 NodeManager
12437 QuorumPeerMain
12496 HRegionServer
```