centos7 +hadoop2.7.5+hbase1.4.3 安装 (一)

作者：centrexzj

hadoop2.7.2完全分布式环境搭建

环境准备

我们准备3台服务器

192.168.31.216 namenode 主机名：hbase1 =注意namenode对应hbase1
192.168.31.217 datanode1 主机名：hbase2
192.168.31.218 datanode2 主机名：hbase3

1. 创建hadoop用户(3台操作)

先使用groupadd hadoop 建立hadoop用户组

新建用户，useradd -d /usr/hadoop -g hadoop -m hadoop （新建用户hadoop指定用户主目录/usr/hadoop 及所属组hadoop）

passwd hadoop 设置hadoop密码（这里设置密码为hadoop）

2. 配置hadoop集群中免密码登录(namenode->datanode1,datanode2)

su hadoop
/usr/hadoop/.ssh
ssh-keygen -t rsa
ssh-copy-id hbase1
ssh-copy-id hbase2
ssh-copy-id hbase3
hbase1,hbase2,hbase3是我的主机名，主机名自己设置的

尝试下是否实现了无密码切换服务器。

3. 安装好jdk1.8版本

centos7.4 安装卸载jdk

● 查看jdk安装包

su root
[hadoop@hbase1 hadoop]$ rpm -qa | grep jdk
copy-jdk-configs-2.2-5.el7_4.noarch
java-1.8.0-openjdk-headless-1.8.0.161-0.b14.el7_4.x86_64
java-1.8.0-openjdk-accessibility-1.8.0.161-0.b14.el7_4.x86_64
java-1.8.0-openjdk-javadoc-zip-1.8.0.161-0.b14.el7_4.noarch
java-1.8.0-openjdk-javadoc-zip-debug-1.8.0.161-0.b14.el7_4.noarch
java-1.8.0-openjdk-headless-debug-1.8.0.161-0.b14.el7_4.x86_64
java-1.8.0-openjdk-debug-1.8.0.161-0.b14.el7_4.x86_64
java-1.8.0-openjdk-1.8.0.161-0.b14.el7_4.x86_64
java-1.8.0-openjdk-accessibility-debug-1.8.0.161-0.b14.el7_4.x86_64
java-1.8.0-openjdk-demo-1.8.0.161-0.b14.el7_4.x86_64
java-1.8.0-openjdk-javadoc-1.8.0.161-0.b14.el7_4.noarch
java-1.8.0-openjdk-src-1.8.0.161-0.b14.el7_4.x86_64
java-1.8.0-openjdk-javadoc-debug-1.8.0.161-0.b14.el7_4.noarch

java-1.8.0-openjdk-src-debug-1.8.0.161-0.b14.el7_4.x86_64
java-1.8.0-openjdk-devel-debug-1.8.0.161-0.b14.el7_4.x86_64
java-1.8.0-openjdk-devel-1.8.0.161-0.b14.el7_4.x86_64
java-1.8.0-openjdk-demo-debug-1.8.0.161-0.b14.el7_4.x86_64

● 卸载jdk

对于如上的每一行输出，都调用rpm -e --nodeps，如：

rpm -e --nodeps java-1.8.0-openjdk-headless-1.8.0.161-0.b14.el7_4.x86_64

-e 表示卸载

--nodeps 表示不考虑是否有依赖问题，强制卸载

-安装openjdk

yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel #安装openjdk

vim /usr/hadoop/.bash_profile

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
PATH=$PATH:$JAVA_HOME/bin

source /usr/hadoop/.bash_profile

4. 修改主机名 vim /etc/hostname 分别改为hbase1,hbase2,hbase3

5. 修改hosts vim /etc/hosts

192.168.31.216 hbase1
192.168.31.217 hbase2
192.168.31.218 hbase3

mkdir -p /usr/local/hadoop
chown -R hadoop:hadoop /usr/local/hadoop

6. 切换hadoop用户

cd /usr/local/hadoop

7.上传hadoop-2.7.5.tar.gz 到/usr/local/hadoop并解压

tar zxvf hadoop-2.7.5.tar.gz

8. 配置hadoop 环境变量,修改hadoop用户目录下的.bash_profile文件

vim /usr/hadoop/.bash_profile
export HADOOP_HOME=/usr/local/hadoop/hadoop-2.7.5
PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
source /usr/hadoop/.bash_profile

9. 修改/usr/local/hadoop/hadoop-2.7.5/etc/hadoop/hadoop-env.sh 文件，修改内容如下：

● 加入一行 export HADOOP_LOG_DIR=/usr/local/hadoop/logs ,如果不存在该日志路径，创建。

● 设置jdk 安装路径 export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk (把配置文件中jdk配置掉注释,并加上路径即可)

修改hadoop配置文件

● core-site.xml

```
<configuration>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>/usr/local/hadoop/hadoop-2.7.5/tmp/</value>
        <description> Abase for other temporary directories</description>

    </property>

    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://hbase1:9000</value>
    </property>
    <property>
        <name>io.file.buffer.size</name>
        <value>4096</value>
    </property>
</configuration>
```

● hdfs-site.xml

```
<configuration>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>/usr/local/hadoop/hadoop-2.7.5/dfs/name</value>
    </property>

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

    <property>
        <name>dfs.datanode.data.dir</name>
        <value>/usr/local/hadoop/hadoop-2.7.5/dfs/data</value>
    </property>

    <property>
        <name>dfs.chechpoint.dir</name>
        <value>/usr/local/hadoop/hadoop-2.7.5/checkpoint/dfs/cname</value>
    </property>

    <property>
        <name>dfs.http.address</name>
```

```
    <value>hbase1:50070</value>
  </property>

  <property>
    <name>dfs.secondary.http.address</name>
    <value>hbase1:50090</value>
  </property>

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

  <property>
    <name>dfs.permissions</name>
    <value>false</value>
  </property>
</configuration>
```

● mapred-site.xml

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
    <final>true</final>
  </property>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>hbase1:10020</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>hbase1:19888</value>
  </property>
</configuration>
```

● yarn-site.xml

```
<configuration>
<!-- Site specific YARN configuration properties -->
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>hbase1:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>hbase1:8030</value>
  </property>
```

```xml
<property>
  <name>yarn.resourcemanager.resource-tracker.address</name>
  <value>hbase1:8031</value>
</property>
<property>
  <name>yarn.resourcemanager.admin.address</name>
  <value>hbase1:8033</value>
</property>
<property>
  <name>yarn.resourcemanager.webapp.address</name>
  <value>hbase1:8088</value>
</property>
</configuration>
```

配置slaves

```
vim etc/hadoop/slaves
hbase1
hbase2
hbase3
```

在hbase2，hbase3上分别修改配置文件

启动之前需要格式化

启动之前，在namenode服务器上先格式化，只需格式化一次就好了

```
hadoop namenode -format
```

启动hadoop

```
start-dfs.sh
start-yarn.sh
```

这就大功告成了