链滴

大数据搭建总结

作者：tadechen
原文链接：https://ld246.com/article/1570810988056
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

1. 安装虚拟机
2. VMware创建虚拟机,一路next就好, **可以在安装时配置好ip,不用修改ifcfg-end33文件**
3. 先创建一个模板虚拟机出来,后面的其他机器可以在这基础上克隆,模板机需要修改的:
    1. 配置好ip(ping locahost/网关/外网都能ping通即可)
    2. 关闭防火墙

```
systemctl stop firewalld
systemctl disable firewalld
```

4. 关闭模板机,克隆三个机子出来,启动,修改ip后重启网卡(之后的操作就可以在XShell做了)

```
vi /etc/sysconfig/network-script/ifcfg-ens33 # 将ip修改,BOOTPROTO修改成static
# 重启网卡
systemctl restart network
```

5. 修改三台机子的hostname和hosts映射文件

```
hostnamectl set-hostname XXX(要取的主机名) # 修改主机名
vi /etc/hosts
```

6. 设置ssh免密登录

```
# 先创建dsa码
ssh-keygen -t dsa -P '' -f /root/.ssh/id_dsa
# 将各自的id_dsa.pub发送给要免密登录的机器,添加到.ssh文件夹中的authorized_keys文件中
# shmily01机器发送
scp id_dsa.pub shmily02:`pwd`/shmily01_id_dsa.pub
# shmily02机器添加
cat shmily01_id_dsa.pub > authorized_keys
01即可免密登录到02
```

7. 到此,基本配置完成啦

2.安装jdk

1. 将需要的压缩包通过XFtp上传到虚拟机中

2. 将jdk解压,配置环境变量

```
tar -zxvf jdk-8u181-linux-x64.tar.gz -C /opt/module
vi /etc/profile
export JAVA_HOME=/opt/module/jdk
export PATH=$PATH:$JAVA_HOME/bin
. /etc/profile
```

3. 安装zookeeper

1. 解压zookeeper压缩包,配置环境变量

2. 复制conf中的zoo_sample.cfg为zoo.cfg并修改

```
dataDir=/var/hadoop/zk
# 在最后添加
server.1=192.168.47.11:2888:3888
server.2=192.168.47.12:2888:3888
server.3=192.168.47.13:2888:3888
```

3. 在/var/hadoop/zk中创建myid, 192.168.47.11写1 192.168.47.12写2 192.168.47.13写3,和serve后的数字一致

```
zkServer.sh start  # 启动
zkServer.sh status # 查看状态
```

4.安装hadoop

1. 解压压缩包,配置环境变量

2. 修改配置文件中的JAVA_HOME(hadoop-env.sh,yarn-env.sh,mapred-env.sh)

3. 直接来搭建完全分布式吧!

4. 修改core-site.xml

```
<configuration>
    <!--用来指定hdfs的路径，mycluster为固定属性名和hdfs-site中mycluster-->
    <property>
        <name>fs.defaultFS</name>
       <value>hdfs://mycluster</value>
    </property>
    <!--用来指定hadoop运行时产生文件的存放目录-->
    <property>
        <name>hadoop.tmp.dir</name>
        <value>/var/hadoop/ha</value>
    </property>
    <!-- 配置zookeeper队列 -->
    <property>
        <name>ha.zookeeper.quorum</name>
        <value>shmily01:2181,shmily02:2181,shmily03:2181</value>
    </property>
```

```
</configuration>
```

5. 修改hdfs-site.xml

```
<configuration>
    <!--配置block副本数量-->
    <property>
        <name>dfs.replication</name>
        <value>2</value>
    </property>
    <property>
        <name>dfs.nameservices</name>
        <value>mycluster</value>
    </property>
    <property>
        <name>dfs.ha.namenodes.mycluster</name>
        <value>nn1,nn2</value>
    </property>
    <!--nn1的RPC通信地址-->
    <property>
        <name>dfs.namenode.rpc-address.mycluster.nn1</name>
        <value>shmily01:8020</value>
    </property>
    <!--nn2的RPC通信地址-->
    <property>
        <name>dfs.namenode.rpc-address.mycluster.nn2</name>
        <value>shmily02:8020</value>
    </property>
    <!--nn1的http通信地址-->
    <property>
        <name>dfs.namenode.http-address.mycluster.nn1</name>
        <value>shmily01:50070</value>
    </property>
    <!--nn2的http通信地址-->
    <property>
        <name>dfs.namenode.http-address.mycluster.nn2</name>
        <value>shmily02:50070</value>
    </property>
    <!--指定namenode的元数据在JournalNode上的存放位置, 这样，namenode2可以从jn集群里
取最新的namenode的信息，达到热备的效果-->
    <property>
        <name>dfs.namenode.shared.edits.dir</name>
        <value>qjournal://shmily01:8485;shmily02:8485;shmily03:8485/mycluster</value>
    </property>
    <!--指定JournalNode存放数据的位置-->
    <property>
        <name>dfs.journalnode.edits.dir</name>
        <value>/var/hadoop/ha/jn</value>
    </property>
    <!--配置切换的实现方式-->
    <property>
        <name>dfs.client.failover.proxy.provider.mycluster</name>
        <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider<
value>
    </property>
```

```
    <!--配置隔离机制-->
    <property>
      <name>dfs.ha.fencing.methods</name>
      <value>sshfence</value>
    </property>
    <!--配置隔离机制的ssh登录秘钥所在的位置-->
    <property>
      <name>dfs.ha.fencing.ssh.private-key-files</name>
      <value>/root/.ssh/id_dsa</value>
    </property>
    <!--开启namenode故障时自动切换-->
    <property>
      <name>dfs.ha.automatic-failover.enabled</name>
      <value>true</value>
    </property>
     <!--设置hdfs的操作权限，false表示任何用户都可以在hdfs上操作文件，生产环境不配置此项，
认为true-->
    <property>
      <name>dfs.permissions</name>
      <value>false</value>
    </property>
</configuration>
```

6. 修改slaves,添加datanode

```
shmily01
shmily02
shmily03
```

7. 添加yarn了,修改mapred-site.xml,让mp支持yarn

```
    <!--指定mapreduce运行在yarn上-->
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
```

8. 修改yarn-site.xml

```
    <!--NodeManager获取数据的方式-->
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <!-- 开启YARN HA -->
    <property>
        <name>yarn.resourcemanager.ha.enabled</name>
        <value>true</value>
    </property>
    <property>
        <name>yarn.resourcemanager.cluster-id</name>
        <value>cluster1</value>
    </property>
    <!-- 指定两个resourcemanager的名称 -->
    <property>
        <name>yarn.resourcemanager.ha.rm-ids</name>
```

```
        <value>rm1,rm2</value>
    </property>
    <property>
        <name>yarn.resourcemanager.hostname.rm1</name>
        <value>shmily02</value>
    </property>
    <property>
        <name>yarn.resourcemanager.hostname.rm2</name>
        <value>shmily03</value>
    </property>
    <property><name>yarn.resourcemanager.webapp.address.rm1</name>
        <value>shmily02:8088</value>
    </property>
    <property>
        <name>yarn.resourcemanager.webapp.address.rm2</name>
        <value>shmily03:8088</value>
    </property>
    <!-- 配置zookeeper的地址 -->
    <property>
      <name>yarn.resourcemanager.zk-address</name>
     <value>shmily01:2181,shmily02:2181,shmily03:2181</value>
    </property>
    <!-- 是否启动一个线程检查每个任务正在使用的物理内存量,如果任务超出分配制,则直接将其杀掉,
认是true -->
    <property>
      <name>yarn.nodemanager.pmem-check-enabled</name>
      <value>false</value>
    </property>
    <property>
      <name>yarn.nodemanager.vmem-check-enabled</name>
      <value>false</value>
    </property>
```

9. **记得把这个hadoop分发给其他俩台机器嗷**

10. 分别启动三台机器的journalnode

```
hadoop-daemon.sh start journalnode
```

11. 在一台namenode中格式化第一台namenode

```
hadoop namenode -format
```

启动第一台namenode

```
hadoop-daemon.sh start namenode
```

12. 在第二台namenode中同步第二台namenode

```
hdfs namenode -bootstrapStandby
```

13. 在第一台shmily01中的格式化zookeeper

```
hdfs zkfc -formatZK
```

14. 好了好了,可以启动了

```
start-all.sh # 开启
```

stop-all.sh # 关闭所有

15. 然后使用jps查看一下进程,resourcemanager应该是没起来,需要去shmily02,shmily03中手动启动

yarn-daemon.sh start resourcemanager

16. jps是这样就ok了