**Hadoop集群搭建详细笔记**

# 准备工作

**总：环境配置**

**1. All nodes are disabled SELinux and firewalld**

**2. All nodes can ping with each other**

**3. All nodes have same hadoop directory structure and a same user account**

**4. Create a hadoop user, home directory is /home/hadoop, add into root group**

**5. hadoop directory is /usr/local/hadoop, directory owner is hadoop**

**6. Master node and slave node can SSH with no password publick key authentication**

**7. All nodes have same /etc/hosts, add master node and slave node record line**

## （1）修改主机名

使用如下命令：

#hostnamectl set-hostname master 【在master主机上执行】  
#hostnamectl set-hostname slave01 【在slave01主机上执行】  
#hostnamectl set-hostname slave02 【在slave02主机上执行】

## （2）修改hosts文件，三台机器依次执行

#sudo vim /etc/hosts

192.168.42.134 master

192.168.42.135 slave01

192.168.42.136 slave02

## （3）关闭防火墙（root权限）

# Centos7 中使用 systemctl 命令来管理服务，命令格式如下

# systemctl [start 开启]|[stop 停止]|[restart 重启]|[status 状态][enable 开机启动]| [disable 禁止开机启动] 服务名称

[root@localhost ~]# systemctl start firewalld # 开启防火墙

[root@localhost ~]# systemctl status firewalld # 查看防火墙状态

[root@localhost ~]# systemctl disable firewalld # 永久关闭防火墙（重启生效）

[root@localhost ~]# systemctl stop firewalld # 临时关闭防火墙（立即生效）

[root@localhost ~]# systemctl status firewalld # 检查修改

## （4）关闭SELinux（root权限）

因为CentOS的所有访问权限都是有SELinux来管理的，为了避免我们安装中由于权限关系而导致的失败，需要先将其关闭，以后根据需要再进行重新管理。

在CentOS中关闭SELinux使用如下操作命令：

[root@localhost ~]# getenforce # 查看当前的 SELinux 状态Enforcing

# setenforce 1 可以设置 SELinux 为 enforcing 模式

[root@localhost ~]# setenforce 0 # SELinux 的状态临时设置为Permissive 模式（立即生效）

[root@localhost ~]# getenforce # 检查修改Permissive

[root@localhost ~]# vi /etc/selinux/config

# 编辑 config 文件将 SELINUX=enforcing 修改为 SELINUX=disabled（重启生效）

[root@localhost ~]# cat /etc/selinux/config # 检查修改

注意:使用 getenforce 命令获取当前 SELinux 的运行状态为 permissive 或者 disabled 时均表示关闭。

## （5）网络配置

采用NAT联网，设置成静态IP，保证内外、外网都能通。

## (6)创建接下来hadoop新用户

1.#su root #以root账号登录并创建接下来hadoop新用户

2.#useradd -m hadoop -s /bin/bash #创建新用户hadoop

3.#passwd hadoop 给hadoop新用户设置登录密码

4.visudo 增加管理员权限（root权限角色）

5. 找到root ALL=（ALL）ALL这一行，一般在98行附近，然后在这一行下面 增加下面指令： hadoop ALL=（ALL）ALL （当中的间隔为tab）

6. 添加完之后，保存退出

7. #su hadoop,切换到hadoop用户上来

## (７)配置ssh免密登录（每台服务器同样的操作）

#生成ssh免密登录密钥

#ssh-keygen –t rsa (四个回车)

执行完这个命令后，会生成id\_rsa(私钥)、id\_rsa.pub(公钥)

将公钥拷贝到要免密登录的目标机器上

#ssh-copy-id localhost

#ssh-copy-id master

#ssh-copy-id slave01

#ssh-copy-id slave02

测试：

#ssh master 登录成功

#ssh slave01 登录成功

#ssh slave02 登录成功

## （８）创建相关目录

在/usr/local/ 目录下创建目录：java、hadoop、zookeeper、hbase四个目录并分配权限给hadoop用户：

#cd /usr/local

#sudo mkdir java

#sudo mkdir hadoop

#sudo mkdir zookeeper

#sudo mkdir hbase

[root@localhost local]# sudo chown -R hadoop:hadoop /usr/local/java

[root@localhost local]# sudo chown -R hadoop:hadoop /usr/local/hadoop

[root@localhost local]# sudo chown -R hadoop:hadoop /usr/local/zookeeper

[root@localhost local]#sudo chown -R hadoop:hadoop /usr/local/hbase

## （９）安装JDK

1、新建一个文件夹：java

2、上传jdk的压缩包到java文件夹中

3、解压：tar -zxvf jdk-13.0.2\_linux-x64\_bin.tar.gz

4、配置环境变量：#vim /etc/profile

#添加如下内容：

#JAVA\_HOME

/usr/local/java/jdk1.8.0\_172

export JAVA\_HOME=/usr/local/java/jdk-13.0.2

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

5、#source /etc/profile #刷新

6、jdk测试

#java -version

#javac -version

## （１０）同步时间（三台服务器）

#sudo yum install -y ntp

#同步

#sudo ntpdate cn.pool.ntp.org

1. **HADOOP分布式部署**

**总：修改如下文件：**

**1.修改core-site.xml**

**2.修改Hadoop-env.sh**

**3.修改hdfs-site.xml**

**4.修改mapred-site.xml**

**5.修改yarn-site.xml**

**6.修改yarn.env.sh**

**7.修改workers文件**

## （1）core-site.xml修改为如下:

<configuration>  
 <property>  
 <name>fs.default.name</name>  
 <value>hdfs://master:8020/hbase</value>  
 </property>  
 <property>  
 <name>hdfs://master:8020/hbase</name>  
 <value></value>  
 </property>  
 <property>  
 <name>io.file.buffer.size</name>  
 <value>131072</value>  
 </property>  
 <property>  
 <name>hadoop.tmp.dir</name>  
 <value>file:/usr/local/hadoop/hadoop-2.9.2/data</value>  
 </property>  
 <property>  
 <name>hadoop.proxyuser.hduser.hosts</name>  
 <value>\*</value>  
 </property>  
 <property>  
 <name>hadoop.proxyuser.hduser.groups</name>  
 <value>\*</value>  
 </property>  
  
 <property>  
 <name>ha.zookeeper.quorum</name>  
 <value>master:2181,slave01:2181,slave02:2181</value>  
 </property>  
  
</configuration>

## (2) Hadoop-env.sh修改为如下:

*export JAVA\_HOME=/usr/local/java/jdk1.8.0\_172*

*export PATH=$PATH:/usr/local/hadoop/hadoop-3.2.1/bin*

**#刷新**

*# source hadoop-env.sh*

*# hadoop version*

## （3）hdfs-site.xml修改为如下:

<configuration>  
 <name>dfs.replication</name>  
 <value>3</value>  
 </property>  
  
 <property>  
 <name>dfs.namenode.http-address</name>  
 <value>master:8084</value>  
 </property>  
  
 <property>  
 <name>dfs.datanode.data.dir</name>  
 <value>/usr/local/hadoop/hadoop-2.9.2/data</value>  
 </property>  
  
 <!-- hdfs保存namenode当前数据的路径，默认值需要配环境变量，建议使  
用自己创建的路径，方便管理-->  
 <property>  
 <name>dfs.namenode.name.dir</name>  
 <value>/usr/local/hadoop/hadoop-2.9.2/data</value>  
 </property>  
</configuration>

## （4）mapred-site.xml修改为如下:

<configuration>  
<property>  
 <name>mapreduce.framework.name</name>  
 <value>yarn</value>  
</property>  
  
 <property>  
 <name>yarn.app.mapreduce.am.env</name>  
 <value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop/hadoop-2.9.2/</value>  
 </property>  
 <property>  
 <name>mapreduce.map.env</name>  
 <value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop/hadoop-2.9.2/</value>  
 </property>  
 <property>  
 <name>mapreduce.reduce.env</name>  
 <value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop/hadoop-2.9.2/</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>

## (5)yarn-site.xml修改为如下：

<configuration>  
  
<!-- 必须配置 指定YARN的老大（ResourceManager）在哪一台主机 -->  
<property>  
 <name>yarn.resourcemanager.hostname</name>  
 <value>master</value>  
</property>  
  
<!-- 必须配置 提供mapreduce程序获取数据的方式 默认为空 -->  
<property>  
 <name>yarn.nodemanager.aux-services</name>  
 <value>mapreduce\_shuffle</value>  
</property>  
  
  
<property>  
 <name>yarn.resourcemanager.address</name>  
 <value>master:8032</value>  
</property>  
  
<property>  
 <name>yarn.resourcemanager.scheduler.address</name>  
 <value>master:8030</value>  
 </property>  
  
<property>  
 <name>yarn.resourcemanager.resource-tracker.address</name>  
 <value>master:8031</value>  
</property>  
  
<property>  
 <name>yarn.resourcemanager.admin.address</name>  
 <value>master:8033</value>  
</property>  
  
<property>  
 <name>yarn.resourcemanager.webapp.address</name>  
 <value>master:3206</value>  
</property>  
  
 <name>ha.zookeeper.quorum</name>  
 <value>master:2181,slave01:2181,slave02:2181</value>  
</property>  
  
<!--配置与zookeeper的连接地址-->  
 <property>  
 <name>yarn.resourcemanager.zk-state-store.address</name>  
 <value>master:2181,slave01:2181,slave02:2181</value>  
 </property>  
 <property>  
 <name>yarn.resourcemanager.store.class</name>  
 <value>org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateStore</value>  
 </property>  
 <property>  
 <name>yarn.resourcemanager.zk-address</name>  
 <value>master:2181,slave01:2181,slave02:2181</value>  
 </property>  
  
</configuration>

## （6）yarn-env.sh修改为如下：

*export JAVA\_HOME=/usr/local/java/jdk1.8.0\_172*

*#刷新*

*#source yarn-env.sh*

## （7）修改workers文件

master

slave01

slave02

## (8)配置Hadoop环境变量

**#vim /etc/profile #添加如下内容：**

#HADOOP\_HOME

export HADOOP\_HOME=/usr/local/hadoop/hadoop-2.9.2

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

export PATH=$PATH:$HADOOP\_HOME/sbin

## （9）# source /etc/profile #刷新

**(10)启动服务之前先格式化**，进入/usr/local/hadoop/hadoop-3-0-1/bin文件夹下：

#cd /usr/local/hadoop/hadoop-3-0-1/bin/

#sudo ./hadooop namenode -format

**(11)启动服务，**进入 /usr/local/hadoop/hadoop-3-0-1/sbin目录下

#cd /usr/local/hadoop/hadoop-3-0-1/sbin

#sh start-all.sh

# Zookeeper分布式部署

在zoo.cfg添加如下内容：

dataDir=/usr/local/zookeeper/zookeeper-3.4.14/zkData/data

# the port at which the clients will connect

clientPort=2181

server.1=192.168.42.134:2888:3888

server.2=192.168.42.135:2888:3888

server.3=192.168.42.136:2888:3888

# Hbase分布式部署

修改文件：

<configuration>  
<property>  
  
<name>hbase.rootdir</name>  
  
<value>hdfs://master:8020/hbase</value>  #hdfs前面不能留空格  
  
</property>  
  
<property>  
  
<name>hbase.cluster.distributed</name>  
  
<value>true</value>  
  
</property>  
  
<property>  
  
<name>hbase.zookeeper.quorum</name>  
  
<value>master,slave01,slave02</value>  
  
</property>  
  
<property>  
  
<name>hbase.zookeeper.property.dataDir</name>  
  
<value>/usr/local/zookeeper/zookeeper-3.4.14/zkData/data</value>  
  
</property>  
  
<property>   
  
<name>hbase.tmp.dir</name>   
  
<value>/usr/local/hbase/hbase-2.2.3/data</value>   
  
</property>  
  
<property>  
 <name>hbase.master.info.port</name>  
 <value>9084</value>  
</property>  
  
<property>  
 <name>zookeeper.znode.parent</name>  
 <value>/hbase</value>  
</property>  
  
</configuration>