

# Hadoop 学习理解

----hadoop 集群搭建

笔者: 电杆

2018 年 10 月 6 日

---

## 目录

环境配置 .....	3
1.1基础环境 .....	3
2.hadoop集群配置 .....	6
2.1 集群配置 .....	6

## 环境配置

### 1.1 基础环境

#### 1.1.1 虚拟机环境

CentOS7 虚拟机三台

Master	192.168.16.129
Slave2	192.168.16.142
Slave3	192.168.16.143

设置对象虚拟机中主机名与 IP

#### 1.1.2 新增 hadoop 用户

三台虚拟机均执行操作

新增 hadoop 用户，并赋予权限

```
[root@localhost ~]# useradd -m hadoop -s /bin/bash
```

```
[root@localhost ~]# passwd Hadoop
```

新增用户，设置密码

更改用户权限：

```
[root@slave2 current]# visudo
```

添加：

```
hadoop ALL=(ALL)    ALL
```

```
##
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
hadoop  ALL=(ALL)    ALL
## Allows members of the 'sys' group to run
## service management apps and more.
```

#### 1.1.3 在三台虚拟机中配置免密登录

在 master 主机进入.ssh 文件夹，

```
[hadoop@master ~]$ cd ~/.ssh
```

然后执行：

如果之前有密钥，先进行删除

```
[hadoop@master .ssh]$ rm ./id_rsa*
```

然后执行

```
[hadoop@master .ssh]$ exit
```

```
[hadoop@master .ssh]$ ssh-keygen -t rsa
```

```
[hadoop@localhost ~]$ exit
登出
Connection to localhost closed.
[hadoop@localhost root]$ cd ~/.ssh/
[hadoop@localhost .ssh]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:JBwi9Ctekkq0gv8yqlr2HhfPDLHoACqehk0isqGs5jQ hadoop@localhost.localdomain
The key's randomart image is:
+---[RSA 2048]-----+
| .o . . |
| o o . |
|.. . + . |
|+... o = |
|Xo* + + S |
|@X.* * |
|=E* o . + |
|+++o o |
|Xo ++ |
+----[SHA256]-----+
[hadoop@localhost .ssh]$
```

生成密钥

然后执行本机免密

```
[hadoop@master .ssh]$ cat ./id_rsa.pub >> ./authorized_keys
```

```
[hadoop@master .ssh]$ chmod 600 ./authorized_keys
```

让 Master 节点需能无密码 SSH 本机，在 Master 节点上执行：

完成后可执行 `ssh Master` 验证一下（可能需要输入 `yes`，成功后执行 `exit` 返回原来的终端）。接着在 Master 节点将上公匙传输到 Slave1、2 节点：

```
[hadoop@master .ssh]$ scp ~/.ssh/id_rsa.pub hadoop@Slave2:/home/hadoop/
```

```
[hadoop@master .ssh]$ scp ~/.ssh/id_rsa.pub hadoop@Slave3:/home/hadoop/
```

然后在 slave2、slave3 中执行：（写入密钥）

```
cat ./id_rsa.pub >> ./authorized_keys
```

完成后再 master 中测试：

```
[hadoop@master .ssh]$ ssh slave2
```

```
Last login: Thu Sep 27 16:02:16 2018 from ::1
```

```
[hadoop@master .ssh]$ ssh slave2
Last login: Thu Sep 27 16:02:16 2018 from ::1
[hadoop@slave2 ~]$ exit
logout
Connection to slave2 closed.
```

若能登录。则表明配置正确，

---

#### 1.1.4 配置 java 环境

在之前的伪分布式中已有配置，这里参考之前的文档，配置 slave2 与 slave3 中的 java 环境，此处不在做详细说明

---

#### 1.1.5 关闭防火墙

操作参考 hadoop 分布式部署中方法

## 2.hadoop 集群配置

### 2.1 集群配置

#### 2.2.1 hadoop 集群配置文件

集群配置需要更改 5 个文件：

- slaves、core-site.xml、
- hdfs-site.xml、
- mapred-site.xml、
- yarn-site.xml
- slaves(新增)

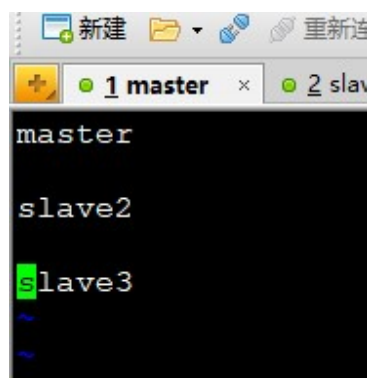
进入配置文件文件夹

```
[hadoop@Master ~]$ cd /usr/local/hadoop/etc/hadoop
```

1、新增文件 slaves

```
[hadoop@Master hadoop]$ vim slaves
```

```
master
slave2
slave3
```



2、更改 core-site.xml

之前配置伪分布式已经添加，若是新的在添加一遍即可

```
[hadoop@master hadoop]$ vi core-site.xml
```

添加：

```
<property>
```

```
<name>hadoop.tmp.dir</name>
```

```
<value>file:/usr/local/hadoop/tmp</value>
```

```
<description>Abase for other temporary directories.</description>
```

```
</property>
```

```
<property>
```

```
<name>fs.defaultFS</name>
```

```
<value>hdfs://master:9000</value>
```

```
</property>
```

```

<configuration>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>file:/usr/local/hadoop/tmp</value>
    <description>Abase for other temporary directories.</description>
  </property>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:9000</value>
  </property>
</configuration>
[hadoop@master hadoop]$

```

3、更改 hdfs-site.xml 将其配置设为 3

```
[hadoop@master hadoop]$ vi hdfs-site.xml
```

添加或 更改:

```
<property>
```

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

```
<value>3</value>
```

```
</property>
```

```
<property>
```

```
<name>dfs.namenode.name.dir</name>
```

```
<value>file:/usr/local/hadoop/tmp/dfs/name</value>
```

```
</property> <property>
```

```
<name>dfs.datanode.data.dir</name>
```

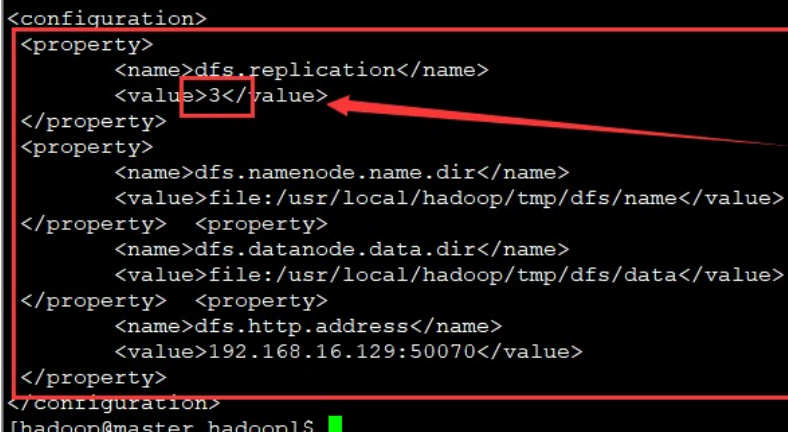
```
<value>file:/usr/local/hadoop/tmp/dfs/data</value>
```

```
</property> <property>
```

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

```
<value>192.168.16.129:50070</value>
```

```
</property>
```



```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/usr/local/hadoop/tmp/dfs/name</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/usr/local/hadoop/tmp/dfs/data</value>
  </property>
  <property>
    <name>dfs.http.address</name>
    <value>192.168.16.129:50070</value>
  </property>
</configuration>
```

#### 4、修改 mapred-site.xml

```
[hadoop@master hadoop]$ vi mapred-site.xml
```

配置如下:

```
<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>
```

(注意: 其中红色框内为自己主机的名称, 自行更改)



```

<!-- Put site-specific property overrides in this file. -->
<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>
[hadoop@master hadoop]$

```

##### 5、修改 yarn-site.xml:

```
[hadoop@master hadoop]$ vi yarn-site.xml
```

配置如下:

```

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

```

```

</configuration>
<!-- Site specific YARN configuration properties -->
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>master</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
</configuration>
[hadoop@master hadoop]$

```

### 2.2.2 配置 slave2、slave3

将在 master 上配置好的 hadoop 文件夹压缩。并拷贝到 slave2、slave3 虚拟机中相同位置并解压

压缩前先删除之前运行的临时文件：

```
[hadoop@Master local]$ sudo rm -r ./hadoop/tmp
```

删除日志文件

```
[hadoop@Master local]$ sudo rm -r ./hadoop/logs/*
```

先压缩再复制

```
[hadoop@Master local]$ tar -zcf ~/hadoop.master.tar.gz ./hadoop
```

```
[hadoop@Master ~]$ scp ./hadoop.master.tar.gz slave2:/home/hadoop
```

```
[hadoop@Master ~]$ scp ./hadoop.master.tar.gz slave3:/home/hadoop
```

拷贝成功或在 slave2 与 slave3 上操作，一 slave2 做演示，二者操作一致，

解压传过来的文件

```
[hadoop@Slave2 ~]$ sudo tar -zxf ~/hadoop.master.tar.gz -C /usr/local
```

修改所有者

```
[hadoop@Slave1 ~]$ sudo chown -R hadoop /usr/local/hadoop
```

在 slave2 与 slave3 上均解压完成后，在 master 上进行 NameNode 的格式化：

（仅在 master 上执行。仅执行一次）

```
[hadoop@Master ~]$ hdfs namenode -format
```

切换到 sbin 目录：

```
[hadoop@master sbin]$ cd /usr/local/hadoop/sbin
```

启动服务：

```
[hadoop@Master ~]$ start-dfs.sh
```

```
[hadoop@Master ~]$ start-yarn.sh
```

```
[hadoop@Master ~]$ mr-jobhistory-daemon.sh start historyserver
```

通过命令 `jps` 可以查看各个节点所启动的进程。正确的话，在 Master 节点上可以看到 NameNode、ResourceManager、SecondaryNameNode、JobHistoryServer 进程，如下图所示：

```

[adoop@master sbin]$ jps
35621 JobHistoryServer
34630 DataNode
34520 NameNode
35210 NodeManager
34798 SecondaryNameNode
35102 ResourceManager
38239 Jps
[adoop@master sbin]$

```

已连接 192.168.16.129:22。

在 Slave 节点可以看到 DataNode 和 NodeManager 进程，如下图所示：

```

[adoop@Slave2 root]$ jps
1515 NodeManager
1646 Jps
1407 DataNode

```

也可以通过 Web 页面看到查看 DataNode 和 NameNode 的状态：

<http://master:50070/>。

缺少任一进程都表示出错。另外还需要在 Master 节点上通过命令 `hdfs dfsadmin`

`-report` 查看 DataNode 是否正常启动，如果 Live datanodes 为 3，则说明集群启动成功。

