

在搭建 Hadoop 集群之前需要准备已经安装好的 3 台节点分别为: master、slave1 和 slave2, Linux 节点安装过程这里不在赘述。

## 1. 集群规划

### 1.1 主机规划

	Master	Slave1	Slave2
Namenode	是	是	
Datanode	是	是	是
ResourceManager	是	是	
NodeManager	是	是	是
Journalnode	是	是	是
Zookeeper	是	是	是

### 1.2 软件规划

软件	版本	位数
Jdk	1.8	64
Centos	6.5	64
Zookeeper	zookeeper-3.4.5-cdh5.13.0.tar.gz	
Hadoop	hadoop-2.6.0-cdh5.13.0.tar.gz	

### 1.3 用户规划

节点名称	用户组	用户
Master	Cdh	Cdh
Slave1	Cdh	Cdh
Slave2	Cdh	Cdh

### 1.4 目录规划

名称	路径
所有软件目录	/home/cdh/app
脚本目录	/home/cdh/tools
数据目录	/home/cdh/data

## 2. 环境检查

### 2.1 时钟同步

统一时区

```
cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

NTP（网络时间协议）时钟同步

```
yum install ntp //下载安装 ntp
```

```
ntpdate pool.ntp.org 同步时间
```

### 2.2 Hosts 文件配置

配置集群所有节点 ip 与 hostname 的映射关系

```
192.168.8.130 master
```

```
192.168.8.131 slave1
```

192.168.8.132 slave2

### 1.3 关闭防火墙

查看防火墙状态

`service iptables status`

永久关闭防火墙

`chkconfig iptables off`

临时关闭防火墙

`service iptables stop`

### 1.4 ssh 面密码登录

首先每个节点单独配置 ssh 免密码登录

切换到用户根目录

`mkdir .ssh`

`ssh-keygen -t rsa`

进入.ssh 文件

`cd .ssh`

`cat id_rsa.pub >> authorized_keys`

退回到根目录

`chmod 700 .ssh`

`chmod 600 .ssh/*`

`ssh master`

将 slave1 和 slave2 的公钥 id\_rsa.pub 拷贝到 master 中的 authorized\_keys 文件中。

`cat ~/.ssh/id_rsa.pub | ssh cdh@master 'cat >> ~/.ssh/authorized_keys'`

然后将 master 中的 authorized\_keys 文件分发到 slave1 和 slave2 节点上面。

`scp -r authorized_keys cdh@slave1:~/.ssh/`

`scp -r authorized_keys cdh@slave2:~/.ssh/`

然后 master、slave1 和 slave2 就可以免密码互通

### 1.5 集群脚本工具准备

创建/home/cdh/tools 脚本存放目录，将以下脚本(见本地目录)上传至该目录

`mkdir /home/cdh/tools`

`deploy.conf`    `deploy.sh`    `runRemoteCmd.sh`

给脚本添加执行权限

`chmod u+x deploy.sh`

`chmod u+x runRemoteCmd.sh`

配置脚本环境变量

`vi ~/.bashrc`

`PATH=/home/cdh/tools:$PATH`

`export PATH`

批量创建各个节点相应目录

```
runRemoteCmd.sh "mkdir /home/cdh/app" all
```

```
runRemoteCmd.sh "mkdir /home/cdh/data" all
```

### 3. Jdk 安装

1.1 下载 jdk-8u51-linux-x64.tar.gz 上传至 master 节点的/home/cdh/app 目录下

1.2 解压: tar -zxvf jdk-8u51-linux-x64.tar.gz

1.3 创建软连接: ln -s jdk1.8.0\_51 jdk

1.4 配置环境变量

```
vi ~/.bashrc
```

```
JAVA_HOME=/home/cdh/app/jdk
```

```
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

```
PATH=$JAVA_HOME/bin:/home/cdh/tools:$PATH
```

```
export JAVA_HOME CLASSPATH PATH
```

1.5 保存并使得配置文件生效: source ~/.bashrc

1.6 查看 jdk 是否安装成功: java -version

1.7 将 jdk1.8.0\_51 安装包分发到 slave1 和 slave2 节点

```
deploy.sh jdk1.8.0_51 /home/cdh/app/ slave
```

然后做同样的操作, 确保每个节点 jdk 安装成功

### 4. Zookeeper 安装

1.1 下载 zookeeper-3.4.5-cdh5.13.0.tar.gz 上传至 master 节点的/home/cdh/app 目录下

1.2 解压: tar -zxvf zookeeper-3.4.5-cdh5.13.0.tar.gz

1.3 创建软连接: ln -s zookeeper-3.4.5-cdh5.13.0 zookeeper

1.4 修改 zoo.cfg 配置文件

复制一份 zoo.cfg 配置文件

```
cp zoo_sample.cfg zoo.cfg
```

修改 zoo.cfg 配置文件

(详情见本地文件)

1.5 将 Zookeeper 安装目录整体分发到 slave1 和 slave2 节点

```
deploy.sh zookeeper-3.4.5-cdh5.13.0 /home/cdh/app/ slave
```

并分别创建软连接

```
ln -s zookeeper-3.4.5-cdh5.13.0 zookeeper
```

1.6 所有节点创建 zoo.cfg 配置文件中的数据目录和日志目录

```
runRemoteCmd.sh "mkdir -p /home/cdh/data/zookeeper/zkdata" all
```

```
runRemoteCmd.sh "mkdir -p /home/cdh/data/zookeeper/zkdata" all
```

1.7 Master、slave1、slave2 节点, 进入/home/cdh/data/zookeeper/zkdata 目录, 创建文件 myid, 里面的内容分别填充为: 1、2、3

```
[cdh@master zkdata]$ vi myid
```

```
[cdh@master zkdata]$ cat myid
```

```
1
```

```
[cdh@slave1 zkdata]$ vi myid
```

```
[cdh@slave1 zkdata]$ cat myid
```

```
2
```

```
[cdh@slave2 zkdata]$ vi myid
```

```
[cdh@slave2 zkdata]$ cat myid
```

### 3

#### 1.8 各个节点配置 Zookeeper 环境变量

```
vi ~/.bashrc
JAVA_HOME=/home/cdh/app/jdk
ZOOKEEPER_HOME=/home/cdh/app/zookeeper
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
PATH=$JAVA_HOME/bin:/home/cdh/tools:$ZOOKEEPER_HOME/bin:$PATH
export JAVA_HOME CLASSPATH PATH ZOOKEEPER_HOME
保存并使之生效: source ~/.bashrc
```

#### 1.9 测试运行 Zookeeper

启动 Zookeeper

```
runRemoteCmd.sh "/home/cdh/app/zookeeper/bin/zkServer.sh start" all
```

查看 Zookeeper 进程

```
runRemoteCmd.sh "jps" all
```

查看 Zookeeper 状态

```
runRemoteCmd.sh "/home/cdh/app/zookeeper/bin/zkServer.sh status" all
```

### 5. Hdfs 安装

#### 1.1 下载 hadoop-2.6.0-cdh5.13.0.tar.gz, 上传至 master 节点的/home/cdh/app 目录下

#### 1.2 解压: tar -zxvf hadoop-2.6.0-cdh5.13.0.tar.gz

#### 1.3 创建软连接: ln -s hadoop-2.6.0-cdh5.13.0 hadoop

#### 1.4 修改 hdfs 配置文件: core-site.xml、hdfs-site.xml、slaves、hadoop-env.sh

#### 1.5 将 hadoop 安装目录整体分发到 slave1 和 slave2 节点

```
deploy.sh hadoop-2.6.0-cdh5.13.0 /home/cdh/app/ slave
```

slave1 和 slave2 分别创建软连接

```
ln -s hadoop-2.6.0-cdh5.13.0 hadoop
```

#### 1.6 配置 hadoop 环境变量

```
vi ~/.bashrc
JAVA_HOME=/home/cdh/app/jdk
ZOOKEEPER_HOME=/home/cdh/app/zookeeper
HADOOP_HOME=/home/cdh/app/hadoop
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
PATH=$JAVA_HOME/bin:/home/cdh/tools:$ZOOKEEPER_HOME/bin:$HADOOP_
HOME/bin:$PATH
export JAVA_HOME CLASSPATH PATH ZOOKEEPER_HOME HADOOP_HOME
保存并使之生效
source ~/.bashrc
```

#### 1.7 测试运行 hdfs

##### 1.7.1.1 所有节点启动 Zookeeper:

```
runRemoteCmd.sh "/home/cdh/app/zookeeper/bin/zkServer.sh start" all
```

##### 1.7.1.2 所有节点启动 journalnode:

```
runRemoteCmd.sh "/home/cdh/app/hadoop/sbin/hadoop-daemon.sh
start journalnode" all
```

##### 1.7.1.3 Master 节点格式化 namenode

```
bin/hdfs namenode -format
```

- 1.7.1.4 Master 节点格式化 zkfc  
bin/hdfs zkfc -formatZK
- 1.7.1.5 Master 节点启动 namenode  
bin/hdfs namenode
- 1.7.1.6 Slave1 节点同步 master 节点元数据信息  
bin/hdfs namenode -bootstrapStandby
- 1.7.1.7 Ctrl+c 关闭 master 节点 namenode 进程
- 1.7.1.8 关闭所有节点 journalnode  
runRemoteCmd.sh "/home/cdh/app/hadoop/sbin/hadoop-daemon.sh  
stop journalnode" all
- 1.7.1.9 一键启动 hdfs: sbin/start-dfs.sh
- 1.7.1.10 一键关闭 hdfs: sbin/stop-dfs.sh
- 1.7.1.11 Web 界面查看 hdfs: <http://master:50070>

## 6. Yarn 安装

- 1.1 Master 节点修改 yarn 配置文件 yarn-site.xml、mapred-site.xml (详情见本地文件)
- 1.2 将 yarn-site.xml、mapred-site.xml 分发到 slave1 和 slave2 节点  
deploy.sh mapred-site.xml /home/cdh/app/hadoop-2.6.0-cdh5.13.0/etc/hadoop  
slave  
deploy.sh yarn-site.xml /home/cdh/app/hadoop-2.6.0-cdh5.13.0/etc/hadoop slave
- 1.3 Master 启动 yarn: sbin/start-yarn.sh
- 1.4 Slave1 节点启动 ResourceManager: sbin/yarn-daemon.sh start resourcemanager
- 1.5 Web 界面查看 yarn: <http://master:8088>
- 1.6 查看 ResourceManager 状态  
bin/yarn rmadmin -getServiceState rm1  
bin/yarn rmadmin -getServiceState rm2
- 1.7 运行测试 Wordcount
  - 1.7.1.1 Master 本地目录新建文件 djt.txt  
[cdh@master hadoop]\$ cat djt.txt  
hadoop spark  
hadoop spark  
hadoop spark
  - 1.7.1.2 Hdfs 文件系统创建 djt 目录  
[cdh@master hadoop]\$ bin/hdfs mkdir /djt  
[cdh@master hadoop]\$ bin/hdfs dfs -ls /
  - 1.7.1.3 将 djt.txt 文件上传至/djt 目录下  
bin/hdfs dfs -put djt.txt /djt/  
bin/hdfs dfs -ls /djt
  - 1.7.1.4 运行 Wordcount  
bin/hadoop jar  
share/hadoop/mapreduce2/hadoop-mapreduce-examples-2.6.0-cdh5.13.  
0.jar wordcount /djt/djt.txt /djt/output
  - 1.7.1.5 查看运行结果  
bin/hdfs dfs -cat /djt/output/\*  
hadoop 3



spark 3

大讲台科技