

编译好的hadoop

集群规划

主机	角色
node1	NN DN RM NM
node2	SNN DN NM
node3	DN NM

上传文件

上传编译好的文件到/export/server

```
1  hadoop-3.3.0-Centos7-64-with-snappy.tar.gz
2  #解压
3
   tar zxvf hadoop-3.3.0-Centos7-64-with-snappy.tar.gz
```

修改配置文件

(路径hadoop-3.3.0/etc/hadoop)

- hadoop-env.sh

```
1  export JAVA_HOME=/export/server/jdk1.8.0_65
2
3  #文件最后添加
4  export HDFS_NAMENODE_USER=root
5  export HDFS_DATANODE_USER=root
6  export HDFS_SECONDARYNAMENODE_USER=root
7  export YARN_RESOURCEMANAGER_USER=root
8  export YARN_NODEMANAGER_USER=root
```

- core-site.xml

```
1  <!-- 设置默认使用的文件系统 Hadoop支持file、HDFS、GFS、ali|Amazon云等文件系统 -->
2  <property>
3      <name>fs.defaultFS</name>
4      <value>hdfs://node1:8020</value>
5  </property>
6
7  <!-- 设置Hadoop本地保存数据路径 -->
```

```

8 <property>
9     <name>hadoop.tmp.dir</name>
10    <value>/export/data/hadoop-3.3.0</value>
11 </property>
12
13 <!-- 设置HDFS web UI用户身份 -->
14 <property>
15     <name>hadoop.http.staticuser.user</name>
16     <value>root</value>
17 </property>
18
19 <!-- 整合hive 用户代理设置 -->
20 <property>
21     <name>hadoop.proxyuser.root.hosts</name>
22     <value>*</value>
23 </property>
24
25 <property>
26     <name>hadoop.proxyuser.root.groups</name>
27     <value>*</value>
28 </property>

```

- hdfs-site.xml

```

1 <!-- 设置SNN进程运行机器位置信息 -->
2 <property>
3     <name>dfs.namenode.secondary.http-address</name>
4     <value>node2:9868</value>
5 </property>

```

- mapred-site.xml

```

1 <!-- 设置MR程序默认运行模式： yarn集群模式 local本地模式 -->
2 <property>
3     <name>mapreduce.framework.name</name>
4     <value>yarn</value>
5 </property>
6
7 <!-- MR程序历史服务器端地址 -->
8 <property>

```

```

 9   <name>mapreduce.jobhistory.address</name>
10   <value>node1:10020</value>
11 </property>
12
13 <!-- 历史服务器web端地址 -->
14 <property>
15   <name>mapreduce.jobhistory.webapp.address</name>
16   <value>node1:19888</value>
17 </property>
18
19 <property>
20   <name>yarn.app.mapreduce.am.env</name>
21   <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
22 </property>
23
24 <property>
25   <name>mapreduce.map.env</name>
26   <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
27 </property>
28
29 <property>
30   <name>mapreduce.reduce.env</name>
31   <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
32 </property>

```

- yarn-site.xml

```

 1 <!-- 设置YARN集群主角色运行机器位置 -->
 2 <property>
 3   <name>yarn.resourcemanager.hostname</name>
 4   <value>node1</value>
 5 </property>
 6
 7 <!-- NodeManager上运行的附属服务。需配置成mapreduce_shuffle，才可运行MapReduce程序默认
 8 值: "" -->
 9 <property>
10   <name>yarn.nodemanager.aux-services</name>
11   <value>mapreduce_shuffle</value>
12 </property>
13
14 <!-- 是否将对容器实施物理内存限制 -->

```

```

14 <property>
15     <name>yarn.nodemanager.pmem-check-enabled</name>
16     <value>>false</value>
17 </property>
18
19 <!-- 是否将对容器实施虚拟内存限制。 -->
20 <property>
21     <name>yarn.nodemanager.vmem-check-enabled</name>
22     <value>>false</value>
23 </property>
24
25 <!-- 开启日志聚集 -->
26 <property>
27     <name>yarn.log-aggregation-enable</name>
28     <value>>true</value>
29 </property>
30
31 <!-- 设置yarn历史服务器地址 -->
32 <property>
33     <name>yarn.log.server.url</name>
34     <value>http://node1:19888/jobhistory/logs</value>
35 </property>
36
37 <!-- 保存的时间7天 -->
38 <property>
39     <name>yarn.log-aggregation.retain-seconds</name>
40     <value>604800</value>
41 </property>

```

- workers
 - workers 文件里面记录的是集群主机名

```

1 node1
2 node2
3 node3

```

分发同步hadoop安装包

```

1 cd /export/server
2

```

```
3 scp -r hadoop-3.3.0 root@node2:$PWD
4 scp -r hadoop-3.3.0 root@node3:$PWD
```

将hadoop添加到环境变量

```
1 vim /etc/profile
2
3 export HADOOP_HOME=/export/server/hadoop-3.3.0
4 export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
5
6 source /etc/profile
```

hadoop集群启动

- (==首次启动==) 格式化namenode
 - 首次启动 HDFS 时，必须对其进行格式化操作

```
1 hdfs namenode -format
2 #格式成功(其中会显示以下一句)
3 INFO common.Storage: Storage directory /export/data/hadoop-3.3.0/dfs/name has been
  successfully formatted.
```

如果是刚搭建好的集群可以使用 **start-all.sh** 启动全部服务 **stop-all.sh** 停止全部服务

主机	角色
node1	NN DN RM NM
node2	SNN DN NM
node3	DN NM

- 1.1单节点逐个启动

```
1 #在主节点上使用以下命令启动 HDFS NameNode:
2 hdfs --daemon start namenode
3 #在每个从节点上使用以下命令启动 HDFS DataNode:
4 hdfs --daemon start datanode
5 #在 node2 上使用以下命令启动 HDFS SecondaryNameNode:
6 hdfs --daemon start secondarynamenode
7
```

```
8 #在主节点上使用以下命令启动 YARN ResourceManager:
9 yarn --daemon start resourcemanager
10 #在每个从节点上使用以下命令启动 YARN nodemanager:
11 yarn --daemon start nodemanager
12 #如果想要停止某个节点上某个角色，只需要把命令中的 start 改为 stop 即可。
```

- 1.2脚本一键启动

```
1 start-dfs.sh
2 start-yarn.sh
3 #停止集群:
4 stop-dfs.sh
5 stop-yarn.sh
```

- web ui页面
 - 前提设置了hosts否则node需要换成ip
 - HDFS集群(默认端口9870): <http://node1:9870/>
 - YARN集群(默认端口8088): <http://node1:8088/>

启动 **jobHistoryServer** 服务进程（历史进程）

```
1 mapred --daemon start historyserver
2 #如果关闭的话 用下述命令
3 mapred --daemon stop historyserver
4 #页面访问 jobhistoryserver
5 http://node1:19888/jobhistory
```

HDFS 的垃圾桶机制

- 每一个文件系统都会有垃圾桶机制，便于将删除的数据回收到垃圾桶里面去，避免某些误操作删除一些重要文件。

- 垃圾桶机制配置

```
1 修改 core-site.xml :
2 那么可以按照生产上的需求设置回收站的保存时间，这个时间以分钟为单位，
3 例如 1440=24h=1 天。
4 <property>
5   <name>fs.trash.interval</name>
6   <value>1440</value>
7 </property>
8 然后重启 hdfs 集群
```

• 垃圾桶机制验证

- 1 如果启用垃圾箱配置，`dfs` 命令删除的文件不会立即从 HDFS 中删除。相反，
- 2 HDFS 将其移动到垃圾目录（每个用户在 `/user/<username>/Trash` 下都有自己的
- 3 垃圾目录）。只要文件保留在垃圾箱中，文件可以快速恢复。
- 4 使用 `skipTrash` 选项删除文件，该选项不会将文件发送到垃圾箱。它将从
- 5 HDFS 中完全删除。

错误1:运行hadoop3官方自带mr实力出错

- 错误信息

```
1 Error: Could not find or load main class org.apache.hadoop.mapreduce.v2.app.MRAppMaster
2
3 Please check whether your etc/hadoop/mapred-site.xml contains the below configuration:
4 <property>
5   <name>yarn.app.mapreduce.am.env</name>
6   <value>HADOOP_MAPRED_HOME=${full path of your hadoop distribution directory}</value>
7 </property>
8 <property>
9   <name>mapreduce.map.env</name>
10  <value>HADOOP_MAPRED_HOME=${full path of your hadoop distribution directory}</value>
11 </property>
12 <property>
13   <name>mapreduce.reduce.env</name>
14   <value>HADOOP_MAPRED_HOME=${full path of your hadoop distribution directory}</value>
15 </property>
```

- 解决mapred-site.xml,增加以下配置

```
1 <property>
2   <name>yarn.app.mapreduce.am.env</name>
3   <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
4 </property>
5 <property>
6   <name>mapreduce.map.env</name>
7   <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
8 </property>
```

```
9 <property>
10   <name>mapreduce.reduce.env</name>
11   <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
12 </property>
```

- 如果出现bash v3.2+ is required. Sorry.错误
- 可以尝试将默认 shell 更改为 bash

```
1 chsh -s /bin/bash
```

未编译的hadoop和环境配置

- 编译环境软件安装目录

```
1 mkdir -p /export/server
```

Hadoop编译安装

- 安装编译相关依赖

```
1 yum install gcc gcc-c++ make autoconf automake libtool curl lzo-devel zlib-devel
  openssl openssl-devel ncurses-devel snappy snappy-devel bzip2 bzip2-devel lzo lzo-devel
  lzop libXtst zlib -y
2
  yum install -y doxygen cyrus-sasl* saslwrapper-devel*
```

- 手动安装cmake

```
1 #yum卸载已安装cmake 版本低
2 yum erase cmake
3
4 #解压
5 tar zxvf CMake-3.19.4.tar.gz
6
7 #编译安装
8 cd /export/server/CMake-3.19.4
9
10 ./configure
11
12 make && make install
13
14 #验证
```



```
15 [root@node4 ~]# cmake -version
16 cmake version 3.19.4
17
18 #如果没有正确显示版本 请断开SSH连接 重写登录
```

- 手动安装snappy

```
1 #卸载已经安装的
2
3 rm -rf /usr/local/lib/libsnappy*
4 rm -rf /lib64/libsnappy*
5
6 #上传解压
7 tar zxvf snappy-1.1.3.tar.gz
8
9 #编译安装
10 cd /export/server/snappy-1.1.3
11 ./configure
12 make && make install
13
14 #验证是否安装
15 [root@node4 snappy-1.1.3]# ls -lh /usr/local/lib |grep snappy
16 -rw-r--r-- 1 root root 511K Nov  4 17:13 libsnappy.a
17 -rwxr-xr-x 1 root root  955 Nov  4 17:13 libsnappy.la
18 lrwxrwxrwx 1 root root   18 Nov  4 17:13 libsnappy.so -> libsnappy.so.1.3.0
19 lrwxrwxrwx 1 root root   18 Nov  4 17:13 libsnappy.so.1 -> libsnappy.so.1.3.0
20 -rwxr-xr-x 1 root root 253K Nov  4 17:13 libsnappy.so.1.3.0
```

- 安装配置JDK 1.8

```
1 #解压安装包
2 tar zxvf jdk-8u65-linux-x64.tar.gz
3
4 #配置环境变量
5 vim /etc/profile
6
7 export JAVA_HOME=/export/server/jdk1.8.0_65
8 export PATH=$PATH:$JAVA_HOME/bin
9 export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
10
11 source /etc/profile
```

```
12
13 #验证是否安装成功
14 java -version
15
16 java version "1.8.0_65"
17 Java(TM) SE Runtime Environment (build 1.8.0_65-b17)
18 Java HotSpot(TM) 64-Bit Server VM (build 25.65-b01, mixed mode)
19 You have new mail in /var/spool/mail/root
```

- 安装配置maven

```
1 #解压安装包
2 tar zxvf apache-maven-3.5.4-bin.tar.gz
3
4 #配置环境变量
5 vim /etc/profile
6
7 export MAVEN_HOME=/export/server/apache-maven-3.5.4
8 export MAVEN_OPTS="-Xms4096m -Xmx4096m"
9 export PATH=$MAVEN_HOME/bin:$PATH
10
11 source /etc/profile
12
13 #验证是否安装成功
14 [root@node4 ~]# mvn -v
15 Apache Maven 3.5.4
16
17 #添加maven 阿里云仓库地址 加快国内编译速度
18 vim /export/server/apache-maven-3.5.4/conf/settings.xml
19
20 <mirrors>
21     <mirror>
22         <id>alimaven</id>
23         <name>aliyun maven</name>
24         <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
25         <mirrorOf>central</mirrorOf>
26     </mirror>
27 </mirrors>
```

- 安装ProtoBuf 3.7.1 - serde = serializer deserializer

```
1 #卸载之前版本的protobuf
2
3 #解压
4 tar zxvf protobuf-3.7.1.tar.gz
5
6 #编译安装
7 cd /export/server/protobuf-3.7.1
8 ./autogen.sh
9 ./configure
10 make && make install
11
12 #验证是否安装成功
13 [root@node4 protobuf-3.7.1]# protoc --version
14 libprotoc 3.7.1
```

- 编译hadoop

```
1 #上传解压源码包
2 tar zxvf hadoop-3.3.0-src.tar.gz
3
4 #编译
5 cd /root/hadoop-3.3.0-src
6
7 mvn clean package -Pdist,native -DskipTests -Dtar -Dbundle.snappy -
  Dsnappy.lib=/usr/local/lib
8
9 #参数说明:
10
11 Pdist,native : 把重新编译生成的hadoop动态库;
12 DskipTests : 跳过测试
13 Dtar : 最后把文件以tar打包
14 Dbundle.snappy : 添加snappy压缩支持【默认官网下载的是不支持的】
15 Dsnappy.lib=/usr/local/lib : 指snappy在编译机器上安装后的库路径
```

基础环境

```
1 # 主机名 hosts映射
2 vim /etc/hosts
3
4 127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
```

```
5  :::1          localhost localhost.localdomain localhost6 localhost6.localdomain6
6
7  192.168.227.151 node1.itcast.cn node1
8  192.168.227.152 node2.itcast.cn node2
9  192.168.227.153 node3.itcast.cn node3
10
11 # JDK 1.8安装 上传 jdk-8u65-linux-x64.tar.gz到/export/server/目录下
12 cd /export/server/
13 tar zxvf jdk-8u65-linux-x64.tar.gz
14
15 #配置环境变量
16 vim /etc/profile
17
18 export JAVA_HOME=/export/server/jdk1.8.0_65
19 export PATH=$PATH:$JAVA_HOME/bin
20 export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
21
22 #重新加载环境变量文件
23 source /etc/profile
24
25 # 集群时间同步
26 yum install ntpdate
27 ntpdate ntp5.aliyun.com
28
29 # 防火墙关闭
30 firewall-cmd --state      #查看防火墙状态
31 systemctl stop firewalld.service  #停止firewalld服务
32 systemctl disable firewalld.service  #开机禁用firewalld服务
33
34 # ssh免密登录
35
36 #node1生成公钥私钥（一路回车）
37 ssh-keygen
38
39 #node1配置免密登录到node1 node2 node3
40 ssh-copy-id node1
41 ssh-copy-id node2
42 ssh-copy-id node3
```

