Hadoop 历史版本安装包下载地址

<https://archive.apache.org/dist/hadoop/common/>

1. 配置虚拟机网络，做好节点之间的主机名映射，节点之间免密登入

虚拟机桥接模式配置：<https://blog.csdn.net/weixin_39158271/article/details/78302309>

节点间免密登入：<https://blog.csdn.net/l1394049664/article/details/82528372>

注意关闭防火墙：

service iptables stop && chkconfig iptables off

服务器目录约定：

所有的按照文件均放在/root/apps下

/root/apps/package  放置下载的安装包（只在主节点下，其他几点通过scp命令安装）

/root/apps/java安装jdk

备注：

/etc/hostname 作用：给服务器起一个名称

/etc/hosts 作用： 类似Windows下的hosts文件，配置节点的ip和域名对应关系

1、在VMware中构建三台centos7服务器，每个节点对应的hostname和IP如下

node1 192.168.254.101

node2  192.168.254.102

node3 192.168.254.103

用户名和密码：

root   111111

2、设置node1和node2、node3免密钥登录，便于在主节点的配置快速复制到从节点

这里node1设置为master节点，node2和node3位从节点，详细的免密钥登录参考

centos7多个节点之间实现免密钥登录

3、在主节点node1安装jdk然后同步到其他节点

首先下载jdk-7u80-linux-i586.tar.gz包上传到服务器，然后使用tar命令解压安装包

tar zxvf jdk-7u80-linux-x64.tar.gz -C /root/apps/java

配置环境变量 vi /etc/profile，把下面的配置添加到最后

rpm -qa | grep java

for old in `rpm -qa | grep java`;do rpm -e --nodeps $old;done

# jdk

export JAVA\_HOME=/root/apps/jdk1.8.0\_221

export JRE\_HOME=/$JAVA\_HOME/jre

export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

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

然后使用：source /etc/profile

然后使用scp命令，把安装包和环境变量配置同步到其他节点

同步复制安装包：

scp -r /root/apps/java/jdk1.8.0\_80 node2:/root/apps/java/jdk1.8.0\_80

scp /etc/profile node2:/etc/profile

其他节点类似操作

二、安装Hadoop

约定目录结构

/root/apps/data               HDFS的数据目录

/root/apps/data/hadoop/dfs/name         存放元数据，在hdfs-site.xml中配置

/root/apps/data/hadoop/dfs/data           存放数据目录，在hdfs-site.xml中配置

/root/apps/temp  Hadoop的临时文件的目录，在core-site.xml中配置

1、安装Hadoop

首先下载hadoop-2.8.1.tar.gz安装包，并且上传到服务器上，然后使用下面的命令解压安装包

tar zxvf hadoop-2.8.1.tar.gz -C //root/apps/

2、配置Hadoop的环境变量

export HADOOP\_HOME=/root/apps/hadoop-2.8.1

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

3、修改Hadoop的配置文件

这里要涉及到的配置文件有7个：

/root/apps/hadoop-2.8.1/etc/hadoop/hadoop-env.sh        Hadoop环境变量

/root/apps/hadoop-2.8.1/etc/hadoop/yarn-env.sh

/root/apps/hadoop-2.8.1/etc/hadoop/slaves                     配置从节点

/root/apps/hadoop-2.8.1/etc/hadoop/core-site.xml           Hadoop核心全局配置文件，其他文件的配置项可以覆盖它的配置项

/root/apps/hadoop-2.8.1/etc/hadoop/hdfs-site.xml           HDFS配置文件，该模板的属性继承于core-site.xml

/root/apps/hadoop-2.8.1/etc/hadoop/mapred-site.xml      MapReduce的配置文件，该模板的属性继承于core-site.xml

//root/apps/hadoop-2.8.1/etc/hadoop/yarn-site.xml

其中mapred-site.xml默认不存在的，可以通过复制mapred-site.xml.template文件获得。

配置文件1：hadoop-env.sh

/root/apps JAVA\_HOME=//root/apps/java/jdk1.7.0\_80

配置文件2：yarn-env.sh

/root/apps JAVA\_HOME=//root/apps/java/jdk1.7.0\_80

配置文件3：slaves （这个文件里面保存所有slave节点）

node2

node3

配置文件4：core-site.xml

<configuration>

/指定hdfs的主端口 namenode要放在哪台机器上

<property>

<name>fs.defaultFS</name>

<value>hdfs://node1:9000</value>

</property>

//流缓冲区大小 128MB

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

<!--用来指定使用hadoop时产生文件的存放目录-->

<property>

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

<value>file://root/apps/temp</value>

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

</property>

</configuration>

配置文件5：hdfs-site.xml

<configuration>

//Namenode HTTP服务器地址和端口。

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>node1:9001</value>

</property>

//存贮在本地的名字节点数据镜象的目录,作为名字节点的冗余备份

<property>

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

<value>file://root/apps/data/hadoop/dfs/name</value>

</property>

//数据节点的块本地存放目录

<property>

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

<value>file://root/apps/data/hadoop/dfs/data</value>

</property>

//备份数

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

//使WebHDFS（REST API）在Namenodes和数据节点

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

</configuration>

配置文件6：mapred-site.xml

<configuration>

//告诉hadoop以后MR(Map/Reduce)运行在YARN上

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>node1:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>node1:19888</value>

</property>

</configuration>

配置文件7：yarn-site.xml

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>node1:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>node1:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value> node1:8031</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value> node1:8033</value>

</property>

</configuration>

4、启动

在启动之前记得格式化namenode空间：

hadoop namenode -format

./start-dfs.sh 启动hdfs

./start-yarn.sh 启动yarn

node1:50070 hdfs管理界面

node1:8088 yarn 管理界面

Resource Manager一般安装在一台专门的机器上

Node Manager应该与HDFS中的data node重叠在一起