### 任务3：HDFS 集群主要配置文件讲解

　　Hadoop 默认提供了两种配置文件：

　　（1）一种是**只读的默认配置文件**，包括 core-default.xml、hdfs-default.xml、mapred-default.xml 和 yarn-default.xml，这些文件包含了 Hadoop 系统各种默认配置参数；

　　（2）另一种是 **Hadoop 集群自定义配置时编辑的配置文件**（这些文件多数没有任何配置内容，都存在于 Hadoop 安装包下的 etc/hadoop 目录中），包括 hadoop-env.sh、yarn-env.sh、core-site.xml、hdfs-site.xml、mapred-site.xml、yarn-site.xml 和 slaves 共 7 个文件，可以根据需要在这些文件中对默认配置文件中的参数进行修改，Hadoop 会**优先**选择这些配置文件中的参数。

　　接下来，先通过下表对 Hadoop 集群搭建可能涉及的主要配置文件及功能进行描述。

| **配置文件** | **功能描述** |
| --- | --- |
| hadoop-env.sh | 配置 Hadoop 运行所需的环境变量 |
| yarn-env.sh | 配置 YARN 运行所需的环境变量 |
| core-site.xml | Hadoop核心全局配置文件，可在其它配置文件中引用该文件 |
| hdfs-site.xml | HDFS 配置文件，继承 core-site.xml 配置文件 |
| mapred-site.xml | MapReduce 配置文件，继承 core-site.xml 配置文件 |
| yarn-site.xml | YARN 配置文件，继承 core-site.xml 配置文件 |
| slaves | Hadoop 集群所有从节点（DataNode 和 NodeManager）列表 |

　　（1）前 2 个配置文件都是用来指定 Hadoop 和 YARN 所需运行环境。

* hadoop-env.sh：用来保证 Hadoop 系统能够正常执行 HDFS 的守护进程 NameNode、SecondaryNameNode 和 DataNode；
* yarn-env.sh：用来保证 YARN 的守护进程 ResourceManager 和 NodeManager 能正常执行。

　　（2）slaves 文件存储了当前集群的所有从节点的列表。

　　（3）其它 4 个配置文件都是用来设置集群运行参数的，在这些配置文件中可以使用 Hadoop 默认配置文件中的参数进行配置来优化 Hadoop 集群，从而使集群更加稳定高效。

　　Hadoop 提供的默认配置文件 core-default.xml、hdfs-default.xml、mapred-default.xml 和 yarn-default.xml 中的参数非常之多，这里不便一一展示说明。读者在具体使用时可以通过访问 Hadoop官方文档<https://hadoop.apache.org/docs/r2.7.7/>，进入到文档最底部的 Configuration 部分进行学习和查看。

**3.1 配置环境变量hadoop-env.sh**

echo $JAVA\_HOME

vim /root/software/hadoop-2.7.7/etc/hadoop/hadoop-env.sh

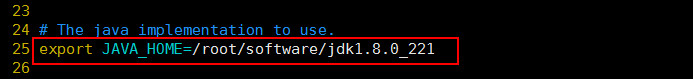


图5

**3.2 配置核心组件core-site.xml**

该文件\*\*是 Hadoop 的核心配置文件，其目的是配置 HDFS 地址、端口号，以及临时文件目录。\*\*使用如下命令打开 “core-site.xml” 文件：

vim /root/software/hadoop-2.7.7/etc/hadoop/core-site.xml

将下面的配置内容添加到 <configuration></configuration> 中间：

<!-- HDFS集群中NameNode的URI（包括协议、主机名称、端口号），默认为 file:/// -->

<property>

<name>fs.defaultFS</name>

<!-- 用于指定NameNode的地址 -->

<value>hdfs://localhost:9000</value>

</property>

<!-- Hadoop运行时产生文件的临时存储目录 -->

<property>

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

<value>/root/hadoopData/temp</value>

</property>

#### 3.3 配置文件系统hdfs-site.xml

该文件主要**用于配置 HDFS 相关的属性**，例如复制因子（即数据块的副本数）、NameNode 和 DataNode 用于存储数据的目录等。在完全分布式模式下，默认数据块副本是 **3 份**。 使用如下命令打开“hdfs-site.xml”文件：

vim /root/software/hadoop-2.7.7/etc/hadoop/hdfs-site.xml

　将下面的配置内容添加到 <configuration></configuration> 中间：

<!-- NameNode在本地文件系统中持久存储命名空间和事务日志的路径 -->

<property>

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

<value>/root/hadoopData/name</value>

</property>

<!-- DataNode在本地文件系统中存放块的路径 -->

<property>

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

<value>/root/hadoopData/data</value>

</property>

<!-- 数据块副本的数量，默认为3 -->

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

#### 3.4 配置slaves文件

该文件**用于记录 Hadoop 集群所有从节点（HDFS 的 DataNode 和 YARN 的 NodeManager 所在主机）的主机名**，用来配合**一键启动**脚本启动集群从节点（并且还需要保证关联节点配置了 SSH 免密登录）。

vim /root/software/hadoop-2.7.7/etc/hadoop/slaves

我们看到其**默认内容为localhost**，因为我们搭建的是伪分布式集群，就只有一台主机，所以从节点也需要放在此主机上，所以此配置文件无需修改。

### 任务4：配置Hadoop系统环境变量

　（1）首先打开/etc/profile文件：

vim /etc/profile

　（2）在文件底部添加如下内容：

# 配置Hadoop的安装目录

export HADOOP\_HOME=/root/software/hadoop-2.7.7

# 在原PATH的基础上加入Hadoop的bin和sbin目录

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

　　注意：

* export 是把这两个变量导出为全局变量。
* 大小写必须严格区分。

　　另外：

* \*\*bin：\*\*存放操作Hadoop相关服务（HDFS、YARN）的脚本，但是通常使用sbin目录下的脚本。
* \*\*sbin：\*\*该目录存放Hadoop管理脚本，主要包含HDFS和YARN中各类服务的启动/关闭脚本。
* （3）让配置文件立即生效，使用如下命令：

source /etc/profile

（4）检测 Hadoop 环境变量是否设置成功，使用如下命令查看 Hadoop 版本：

hadoop version

执行此命令后，若是出现 Hadoop 版本信息说明配置成功：

### 任务5：HDFS 集群测试

#### 5.1 格式化文件系统

　　具体指令如下：

hdfs namenode –format

执行格式化指令后，必须出现有“successfully formatted”信息才表示格式化成功，(倒数第八行最后)

　　若是未出现 “successfully formatted” 信息，就需要查看指令是否正确，或者之前 HDFS 集群的安装和配置是否正确，若是正确，则需要删除所有主机的 “/root/hadoopData” 文件夹，重新执行格式化命令，对 HDFS 集群进行格式化。

　　另外需要特别注意的是，上述格式化指令**只需要在HDFS集群初次启动前执行即可**，后续重复启动就不再需要执行格式化了。

#### 5.2 启动和关闭HDFS集群

针对 HDFS 集群的启动，启动方式有两种，一种是**单节点逐个启动**；另一种是**使用脚本一键启动**。

##### 1. 单节点逐个启动和关闭

　　单节点逐个启动的方式，需要参照以下方式逐个启动 HDFS 集群服务需要的相关服务进程，具体步骤如下：

　　（1）在本机上使用以下指令启动 NameNode 进程：

hadoop-daemon.sh start namenode

jps

解析：

* jps 命令:（全称：Java Virtual Machine Process Status Tool）。安装 JDK 后在%JAVA\_HOME%/bin目录下面自带的一个Java工具。能够显示系统当前运行的Java程序及其进程号。
* 其中 424 和 350 是进程的 PID，也就是进程号。

　　（2）在本机上使用以下指令启动 DataNode 进程：

hadoop-daemon.sh start datanode

　　（3）在本机上使用以下指令启动 SecondaryNameNode 进程：

hadoop-daemon.sh start secondarynamenode

　　另外，当需要停止相关服务进程时，只需要将上述指令中的start更改为stop即可。

##### 2. 脚本一键启动和关闭

**启动集群最常使用的方式是使用脚本一键启动**，前提是需要配置 SSH 免密登录。

* 在本机上使用如下方式一键启动 HDFS 集群：

start-dfs.sh

　打印信息：

* 在本机上启动了 NameNode 守护进程
* 在本机上启动了 DataNode 守护进程
* 在配置的一个特定节点0.0.0.0（本机）上启动 SecondaryNameNode 守护进程

　我们可以一键启动 HDFS 集群，同样也可以一键关闭 HDFS 集群，只需要将 start 改为 stop 即可，即stop-dfs.sh。

#### 5.3 查看进程启动情况

　　在本机执行 jps 命令，在打印结果中会看到 4 个进程，分别是 NameNode、SecondaryNameNode、Jps、和DataNode，如果出现了这 4 个进程表示进程启动成功。如下图所示：

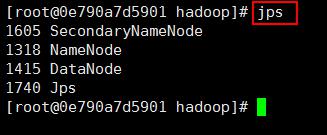


图13

#### 5.4 通过UI查看 HDFS 运行状态

　　HDFS 集群正常启动后，它默认开放了**50070**端口，**用于监控HDFS集群**。通过UI可以方便地进行集群的管理和查看，只需要在本地操作系统的浏览器输入**集群服务的IP和对应的端口号**即可访问。

　　通过本机的浏览器访问http://localhost:50070或http://本机IP地址:50070查看 HDFS 集群状态，效果如下图所示：

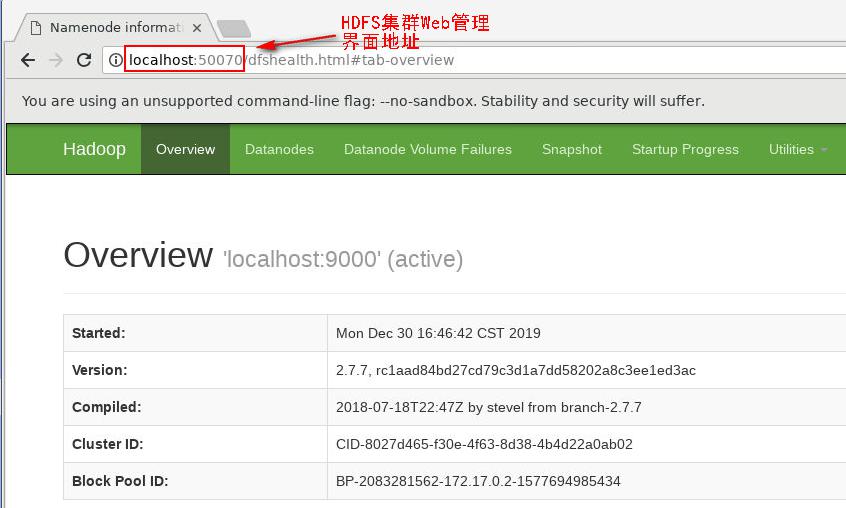


图14

　　从上图可以看出，通过UI可以正常访问 Hadoop 集群的 HDFS 界面，并且页面显示正常，同时通过 UI 可以更方便地进行状态管理和查看。