

Hadoop 运行环境搭建

第 1 章 Hadoop 运行环境搭建

1.1 模板虚拟机环境准备

0) 安装模板虚拟机，IP 地址 192.168.10.100、主机名称 **hadoop100**、内存 4G、硬盘 50G



模板虚拟机环境准备.docx

1) **hadoop100** 虚拟机配置要求如下（本文 Linux 系统全部以 CentOS-7.5-x86-1804 为例）

（1）使用 yum 安装需要虚拟机可以正常上网，yum 安装前可以先测试下虚拟机联网情况

```
[root@hadoop100 ~]# ping www.baidu.com
PING www.baidu.com (14.215.177.39) 56(84) bytes of data.
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=128
time=8.60 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=128
time=7.72 ms
```

（2）安装 epel-release

注：Extra Packages for Enterprise Linux 是为“红帽系”的操作系统提供额外的软件包，适用于 RHEL、CentOS 和 Scientific Linux。相当于是一个软件仓库，大多数 rpm 包在官方 repository 中是找不到的）。

```
[root@hadoop100 ~]# yum install -y epel-release
```

（3）注意：如果 Linux 安装的是最小系统版，还需要安装如下工具；如果安装的是 Linux 桌面标准版，不需要执行如下操作

➤ net-tool: 工具包集合，包含 ifconfig 等命令

```
[root@hadoop100 ~]# yum install -y net-tools
```

➤ vim: 编辑器

```
[root@hadoop100 ~]# yum install -y vim
```

2) 关闭防火墙，关闭防火墙开机自启

```
[root@hadoop100 ~]# systemctl stop firewalld
[root@hadoop100 ~]# systemctl disable firewalld.service
```

注意：在企业开发时，通常单个服务器的防火墙是关闭的。公司整体对外会设置非常安全的防火墙。

3) 创建 iflytek 用户，并修改 iflytek 用户的密码

```
[root@hadoop100 ~]# useradd iflytek
```

```
[root@hadoop100 ~]# passwd iflytek
```

4) 配置 iflytek 用户具有 root 权限，方便后期加 sudo 执行 root 权限的命令

```
[root@hadoop100 ~]# vim /etc/sudoers
```

修改/etc/sudoers 文件，在%wheel 这行下面添加一行，如下所示：

```
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)    ALL
iflytek  ALL=(ALL)    NOPASSWD:ALL
```

注意：iflytek 这一行不要直接放到 root 行下面，因为所有用户都属于 wheel 组，你先配置了 iflytek 具有免密功能，但是程序执行到%wheel 行时，该功能又被覆盖回需要密码。所以 iflytek 要放到%wheel 这行下面。

5) 在/opt 目录下创建文件夹，并修改所属主和所属组

(1) 在/opt 目录下创建 module、software 文件夹

```
[root@hadoop100 ~]# mkdir /opt/module
[root@hadoop100 ~]# mkdir /opt/software
```

(2) 修改 module、software 文件夹的所有者和所属组均为 iflytek 用户

```
[root@hadoop100 ~]# chown iflytek:iflytek /opt/module
[root@hadoop100 ~]# chown iflytek:iflytek /opt/software
```

(3) 查看 module、software 文件夹的所有者和所属组

```
[root@hadoop100 ~]# cd /opt/
[root@hadoop100 opt]# ll
总用量 12
drwxr-xr-x. 2 iflytek iflytek 4096 5月 28 17:18 module
drwxr-xr-x. 2 root    root    4096 9月 7 2017 rh
drwxr-xr-x. 2 iflytek iflytek 4096 5月 28 17:18 software
```

6) 卸载虚拟机自带的 JDK

注意：如果你的虚拟机是最小化安装不需要执行这一步。

```
[root@hadoop100 ~]# rpm -qa | grep -i java | xargs -n1 rpm -e --nodeps
```

- rpm -qa: 查询所安装的所有 rpm 软件包
- grep -i: 忽略大小写
- xargs -n1: 表示每次只传递一个参数
- rpm -e --nodeps: 强制卸载软件

7) 重启虚拟机

```
[root@hadoop100 ~]# reboot
```

1.2 克隆虚拟机

1) 利用模板机 hadoop100，克隆三台虚拟机：hadoop102、hadoop103、hadoop104

注意：克隆时，要先关闭 hadoop100。

2) 修改克隆机 IP，以下以 hadoop102 举例说明

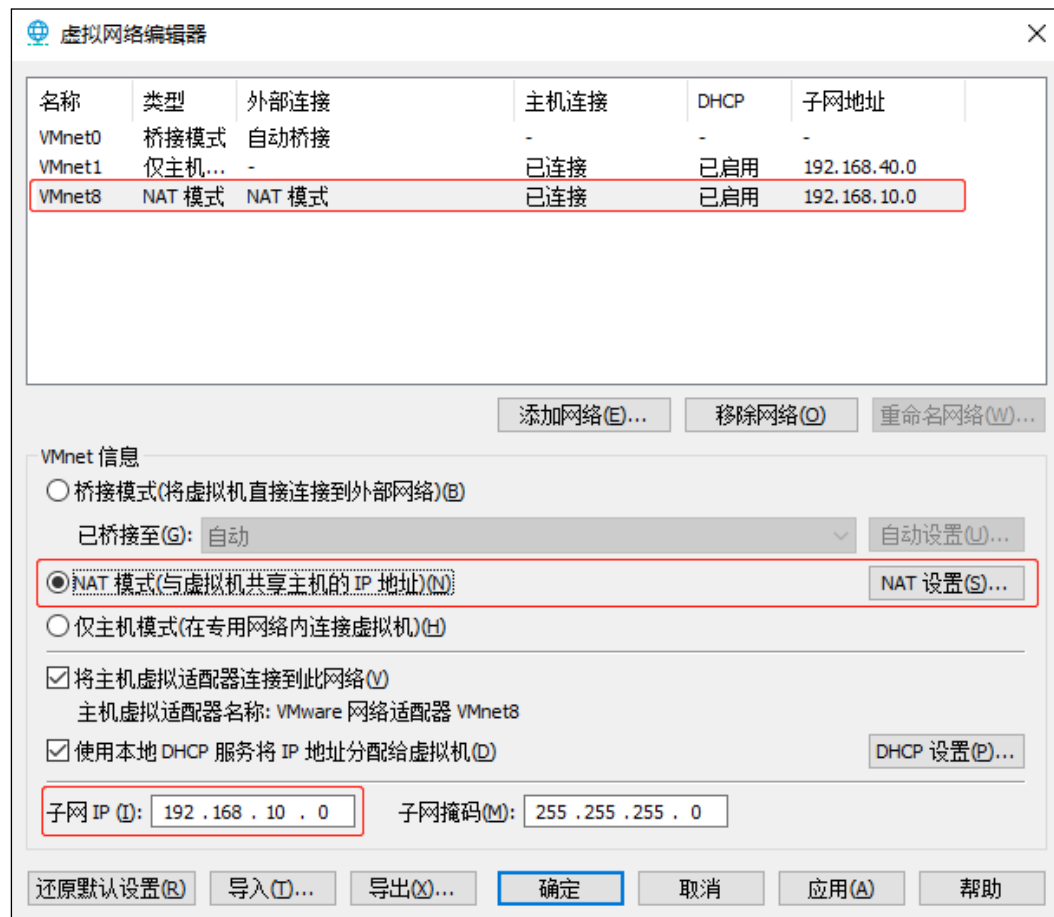
(1) 修改克隆虚拟机的静态 IP

```
[iflytek@hadoop100 ~]# sudo vim /etc/sysconfig/network-scripts/ifcfg-ens33
```

改成

```
DEVICE=ens33
TYPE=Ethernet
ONBOOT=yes
BOOTPROTO=static
NAME="ens33"
IPADDR=192.168.10.102
PREFIX=24
GATEWAY=192.168.10.2
DNS1=192.168.10.2
```

(2) 查看 Linux 虚拟机的虚拟网络编辑器，编辑->虚拟网络编辑器->VMnet8



NAT 设置

网络: vmnet8
子网 IP: 192.168.10.0
子网掩码: 255.255.255.0
网关 IP(G): 192.168.10.2

端口转发(F)

主机端口	类型	虚拟机 IP 地址	描述
------	----	-----------	----

添加(A)... 移除(R) 属性(P)

高级

☒ 允许活动的 FTP(F)
☒ 允许任何组织唯一标识符(O)
UDP 超时(以秒为单位)(U): 30
配置端口(C): 0
☐ 启用 IPv6(E)
IPv6 前缀(G): fd15:4ba5:5a2b:1008::/64

DNS 设置(D)... NetBIOS 设置(N)...

确定 取消 帮助

(3) 查看 Windows 系统适配器 VMware Network Adapter VMnet8 的 IP 地址



(4) 保证 Linux 系统 ifcfg-ens33 文件中 IP 地址、虚拟网络编辑器地址和 Windows 系统 VM8 网络 IP 地址相同。

3) 修改克隆机主机名，以下以 **hadoop102** 举例说明

(1) 修改主机名称

```
[iflytek@hadoop100 ~]# sudo vim /etc/hostname
hadoop102
```

(2) 配置 Linux 克隆机主机名称映射 hosts 文件，打开/etc/hosts

```
[iflytek@hadoop100 ~]# sudo vim /etc/hosts
```

添加如下内容

```
192.168.10.100 hadoop100
192.168.10.101 hadoop101
192.168.10.102 hadoop102
192.168.10.103 hadoop103
192.168.10.104 hadoop104
```

```
192.168.10.105 hadoop105
192.168.10.106 hadoop106
192.168.10.107 hadoop107
192.168.10.108 hadoop108
```

4) 重启克隆机 hadoop102

```
[iflytek@hadoop100 ~]# sudo reboot
```

5) 修改 windows 的主机映射文件 (hosts 文件)

(1) 如果操作系统是 window7, 可以直接修改

①进入 C:\Windows\System32\drivers\etc 路径

②打开 hosts 文件并添加如下内容, 然后保存

```
192.168.10.100 hadoop100
192.168.10.101 hadoop101
192.168.10.102 hadoop102
192.168.10.103 hadoop103
192.168.10.104 hadoop104
192.168.10.105 hadoop105
192.168.10.106 hadoop106
192.168.10.107 hadoop107
192.168.10.108 hadoop108
```

(2) 如果操作系统是 window10, 先拷贝出来, 修改保存以后, 再覆盖即可

①进入 C:\Windows\System32\drivers\etc 路径

②拷贝 hosts 文件到桌面

③打开桌面 hosts 文件并添加如下内容

```
192.168.10.100 hadoop100
192.168.10.101 hadoop101
192.168.10.102 hadoop102
192.168.10.103 hadoop103
192.168.10.104 hadoop104
192.168.10.105 hadoop105
192.168.10.106 hadoop106
192.168.10.107 hadoop107
192.168.10.108 hadoop108
```

④将桌面 hosts 文件覆盖 C:\Windows\System32\drivers\etc 路径 hosts 文件

1.3 编写集群分发脚本 xsync

1) xsync 集群分发脚本

(1) 脚本实现

①在用的家目录/home/iflytek 下创建 bin 文件夹

```
[iflytek@hadoop102 ~]$ cd /home/iflytek
[iflytek@hadoop102 ~]$ mkdir bin
```

②在/home/iflytek/bin 目录下创建 xsync 文件, 以便全局调用

```
[iflytek@hadoop102 ~]$ cd /home/iflytek/bin
[iflytek@hadoop102 ~]$ vim xsync
```

在该文件中编写如下代码

```
#!/bin/bash

#1. 判断参数个数
if [ $# -lt 1 ]
then
    echo Not Enough Argument!
    exit;
fi

#2. 遍历集群所有机器
for host in hadoop102 hadoop103 hadoop104
do
    echo ===== $host =====
    #3. 遍历所有目录，挨个发送
    for file in $(ls -l $@)
    do
        #4 判断文件是否存在
        if [ -e $file ]
        then
            #5. 获取父目录
            pdir=$(cd -P $(dirname $file); pwd)
            #6. 获取当前文件的名称
            fname=$(basename $file)
            ssh $host "mkdir -p $pdir"
            rsync -av $pdir/$fname $host:$pdir
        else
            echo $file does not exists!
        fi
    done
done
```

③修改脚本 xsync 具有执行权限

```
[iflytek@hadoop102 bin]$ chmod 777 xsync
```

④测试脚本

```
[iflytek@hadoop102 bin]$ xsync xsync
```

1.4 SSH 无密登录配置

说明: 这里面只配置了 hadoop102、hadoop103 到其他主机的无密登录; 因为 hadoop102 配置的是 NameNode, hadoop103 配置的是 ResourceManager, 都要求对其他节点无密访问。

(1) hadoop102 上生成公钥和私钥:

```
[iflytek@hadoop102 .ssh]$ ssh-keygen -t rsa
```

然后敲 (三个回车), 就会生成两个文件 id_rsa (私钥)、id_rsa.pub (公钥)。

(2) 将 hadoop102 公钥拷贝到要免密登录的目标机器上

```
[iflytek@hadoop102 .ssh]$ ssh-copy-id hadoop102
[iflytek@hadoop102 .ssh]$ ssh-copy-id hadoop103
[iflytek@hadoop102 .ssh]$ ssh-copy-id hadoop104
```

(3) hadoop103 上生成公钥和私钥:

```
[iflytek@hadoop103 .ssh]$ ssh-keygen -t rsa
```

然后敲（三个回车），就会生成两个文件 `id_rsa`（私钥）、`id_rsa.pub`（公钥）。

（4）将 `hadoop103` 公钥拷贝到要免密登录的目标机器上

```
[iflytek@hadoop103 .ssh]$ ssh-copy-id hadoop102
[iflytek@hadoop103 .ssh]$ ssh-copy-id hadoop103
[iflytek@hadoop103 .ssh]$ ssh-copy-id hadoop104
```

1.5 JDK 准备

1) 卸载现有 JDK (3 台节点)

```
[iflytek@hadoop102 opt]# sudo rpm -qa | grep -i java | xargs -n1
sudo rpm -e --nodeps

[iflytek@hadoop103 opt]# sudo rpm -qa | grep -i java | xargs -n1
sudo rpm -e --nodeps

[iflytek@hadoop104 opt]# sudo rpm -qa | grep -i java | xargs -n1
sudo rpm -e --nodeps
```

（1）`rpm -qa`：表示查询所有已经安装的软件包

（2）`grep -i`：表示过滤时不区分大小写

（3）`xargs -n1`：表示一次获取上次执行结果的一个值

（4）`rpm -e --nodeps`：表示卸载软件

2) 用 XShell 工具将 JDK 导入到 `hadoop102` 的 `/opt/software` 文件夹下面

3) 在 Linux 系统下的 `opt` 目录中查看软件包是否导入成功

```
[iflytek@hadoop102 software]# ls /opt/software/
```

看到如下结果：

```
jdk-8u212-linux-x64.tar.gz
```

4) 解压 JDK 到 `/opt/module` 目录下

```
[iflytek@hadoop102 software]# tar -zxvf jdk-8u212-linux-
x64.tar.gz -C /opt/module/

[iflytek@hadoop102 module]$ mv jdk1.8.0_212/ jdk-1.8.0
```

5) 配置 JDK 环境变量

（1）新建 `/etc/profile.d/my_env.sh` 文件

```
[iflytek@hadoop102 module]# sudo vim /etc/profile.d/my_env.sh
```

添加如下内容，然后保存（`:wq`）退出。

```
#JAVA_HOME
export JAVA_HOME=/opt/module/jdk-1.8.0
export PATH=$PATH:$JAVA_HOME/bin
```

（2）让环境变量生效

```
[iflytek@hadoop102 software]$ source /etc/profile.d/my_env.sh
```

6) 测试 JDK 是否安装成功

```
[iflytek@hadoop102 module]# java -version
```

如果能看到以下结果、则 Java 正常安装。


```
java version "1.8.0_212"
```

7) 分发 JDK

```
[iflytek@hadoop102 module]$ xsync /opt/module/jdk-1.8.0
```

8) 分发环境变量配置文件

```
[iflytek@hadoop102 module]$ sudo /home/iflytek/bin/xsync  
/etc/profile.d/my_env.sh
```

9) 分别在 hadoop103、hadoop104 上执行 source

```
[iflytek@hadoop103 module]$ source /etc/profile.d/my_env.sh  
[iflytek@hadoop104 module]$ source /etc/profile.d/my_env.sh
```

第 2 章 Hadoop 完全分布式集群搭建

分析：

- 1) 准备 3 台客户机（关闭防火墙、静态 IP、主机名称）
- 2) 安装 JDK
- 3) 配置环境变量
- 4) 安装 Hadoop
- 5) 配置环境变量
- 6) 配置集群
- 7) 单点启动
- 8) 配置 ssh
- 9) 群起并测试集群

1.1 Hadoop 部署

1) 集群部署规划

注意：NameNode 和 SecondaryNameNode 不要安装在同一台服务器

注意：ResourceManager 也很消耗内存，不要和 NameNode、SecondaryNameNode 配置在同一台机器上。

	hadoop102	hadoop103	hadoop104
HDFS	NameNode		SecondaryNameNode
	DataNode	DataNode	DataNode
YARN		ResourceManager	
	NodeManager	NodeManager	NodeManager

2) 将 hadoop3.3.4.tar.gz 导入到 opt 目录下面的 software 文件夹下面

3) 进入到 Hadoop 安装包路径下

```
[iflytek@hadoop102 ~]$ cd /opt/software/
```

4) 解压安装文件到/opt/module 下面

```
[iflytek@hadoop102 software]$ tar -zxvf hadoop-3.3.4.tar.gz -C /opt/module/
```

5) 查看是否解压成功

```
[iflytek@hadoop102 software]$ ls /opt/module/hadoop-3.3.4
```

6) 重命名

```
[iflytek@hadoop102 software]$ mv /opt/module/hadoop-3.3.4 /opt/module/hadoop
```

7) 将 Hadoop 添加到环境变量

(1) 获取 Hadoop 安装路径

```
[iflytek@hadoop102 hadoop]$ pwd  
/opt/module/hadoop
```

(2) 打开/etc/profile.d/my_env.sh 文件

```
[iflytek@hadoop102 hadoop]$ sudo vim /etc/profile.d/my_env.sh
```

在 profile 文件末尾添加 JDK 路径: (shift+g)

```
#HADOOP_HOME  
export HADOOP_HOME=/opt/module/hadoop  
export PATH=$PATH:$HADOOP_HOME/bin  
export PATH=$PATH:$HADOOP_HOME/sbin
```

(3) 保存后退出

```
:wq
```

(4) 分发环境变量文件

```
[iflytek@hadoop102 hadoop]$ sudo /home/iflytek/bin/xsync /etc/profile.d/my_env.sh
```

(5) source 使之生效 (3 台节点)

```
[iflytek@hadoop102 module]$ source /etc/profile.d/my_env.sh  
[iflytek@hadoop103 module]$ source /etc/profile.d/my_env.sh  
[iflytek@hadoop104 module]$ source /etc/profile.d/my_env.sh
```

1.2 配置集群

1) 核心配置文件

配置 core-site.xml

```
[iflytek@hadoop102 ~]$ cd $HADOOP_HOME/etc/hadoop  
[iflytek@hadoop102 hadoop]$ vim core-site.xml
```

文件内容如下:

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  
  
<configuration>  
  <!-- 指定 NameNode 的地址 -->  
  <property>  
    <name>fs.defaultFS</name>
```

```

        <value>hdfs://hadoop102:8020</value>
    </property>
    <!-- 指定 hadoop 数据的存储目录 -->
    <property>
        <name>hadoop.tmp.dir</name>
        <value>/opt/module/hadoop/data</value>
    </property>

    <!-- 配置 HDFS 网页登录使用的静态用户为 iflytek -->
    <property>
        <name>hadoop.http.staticuser.user</name>
        <value>iflytek</value>
    </property>

    <!-- 配置该 iflytek(superUser) 允许通过代理访问的主机节点 -->
    <property>
        <name>hadoop.proxyuser.iflytek.hosts</name>
        <value>*</value>
    </property>
    <!-- 配置该 iflytek(superUser) 允许通过代理用户所属组 -->
    <property>
        <name>hadoop.proxyuser.iflytek.groups</name>
        <value>*</value>
    </property>
    <!-- 配置该 iflytek(superUser) 允许通过代理的用户 -->
    <property>
        <name>hadoop.proxyuser.iflytek.users</name>
        <value>*</value>
    </property>
</configuration>

```

2) HDFS 配置文件

配置 hdfs-site.xml

```
[iflytek@hadoop102 hadoop]$ vim hdfs-site.xml
```

文件内容如下：

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
    <!-- nn web 端访问地址-->
    <property>
        <name>dfs.namenode.http-address</name>
        <value>hadoop102:9870</value>
    </property>

    <!-- 2nn web 端访问地址-->
    <property>
        <name>dfs.namenode.secondary.http-address</name>
        <value>hadoop104:9868</value>
    </property>

    <!-- 测试环境指定 HDFS 副本的数量 1 -->
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>

```

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

3) YARN 配置文件

配置 yarn-site.xml

```
[iflytek@hadoop102 hadoop]$ vim yarn-site.xml
```

文件内容如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
  <!-- 指定 MR 走 shuffle -->
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>

  <!-- 指定 ResourceManager 的地址-->
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>hadoop103</value>
  </property>

  <!-- 环境变量的继承 -->
  <property>
    <name>yarn.nodemanager.env-whitelist</name>

<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_
DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HO
ME</value>
  </property>

  <!-- yarn 单个容器允许分配的最大最小内存 -->
  <property>
    <name>yarn.scheduler.minimum-allocation-mb</name>
    <value>512</value>
  </property>
  <property>
    <name>yarn.scheduler.maximum-allocation-mb</name>
    <value>4096</value>
  </property>

  <!-- yarn 容器允许管理的物理内存大小 -->
  <property>
    <name>yarn.nodemanager.resource.memory-mb</name>
    <value>4096</value>
  </property>

  <!-- 关闭 yarn 对物理内存和虚拟内存的限制检查 -->
  <property>
    <name>yarn.nodemanager.pmem-check-enabled</name>
    <value>false</value>
  </property>
  <property>
    <name>yarn.nodemanager.vmem-check-enabled</name>
    <value>false</value>
```

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

4) MapReduce 配置文件

配置 mapred-site.xml

```
[iflytek@hadoop102 hadoop]$ vim mapred-site.xml
```

文件内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
  <!-- 指定 MapReduce 程序运行在 Yarn 上 -->
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

5) 配置 workers

```
[iflytek@hadoop102 hadoop]$ vim /opt/module/hadoop/etc/hadoop/workers
```

在该文件中增加如下内容：

```
hadoop102
hadoop103
hadoop104
```

注意：该文件中添加的内容结尾不允许有空格，文件中不允许有空行。

1.3 配置历史服务器

为了查看程序的历史运行情况，需要配置一下历史服务器。具体配置步骤如下：

1) 配置 mapred-site.xml

```
[iflytek@hadoop102 hadoop]$ vim mapred-site.xml
```

在该文件里面增加如下配置。

```
<!-- 历史服务器端地址 -->
<property>
  <name>mapreduce.jobhistory.address</name>
  <value>hadoop102:10020</value>
</property>

<!-- 历史服务器 web 端地址 -->
<property>
  <name>mapreduce.jobhistory.webapp.address</name>
  <value>hadoop102:19888</value>
</property>
```

1.4 配置日志的聚集

日志聚集概念：应用运行完成以后，将程序运行日志信息上传到 HDFS 系统上。

日志聚集功能好处：可以方便的查看到程序运行详情，方便开发调试。

注意：开启日志聚集功能，需要重新启动 NodeManager 、 ResourceManager 和

HistoryManager。

开启日志聚集功能具体步骤如下：

1) 配置 yarn-site.xml

```
[iflytek@hadoop102 hadoop]$ vim yarn-site.xml
```

在该文件里面增加如下配置。

```
<!-- 开启日志聚集功能 -->
<property>
  <name>yarn.log-aggregation-enable</name>
  <value>true</value>
</property>

<!-- 设置日志聚集服务器地址 -->
<property>
  <name>yarn.log.server.url</name>
  <value>http://hadoop102:19888/jobhistory/logs</value>
</property>

<!-- 设置日志保留时间为 7 天 -->
<property>
  <name>yarn.log-aggregation.retain-seconds</name>
  <value>604800</value>
</property>
```

1.5 分发 Hadoop

```
[iflytek@hadoop102 hadoop]$ xsync /opt/module/hadoop/
```

1.6 群起集群

1) 启动集群

(1) **如果集群是第一次启动**，需要在 hadoop102 节点格式化 NameNode（注意格式化之前，一定要先停止上次启动的所有 namenode 和 datanode 进程，然后再删除 data 和 log 数据）

```
[iflytek@hadoop102 bin]$ cd /opt/module/hadoop/
[iflytek@hadoop102 hadoop]$ bin/hdfs namenode -format
```

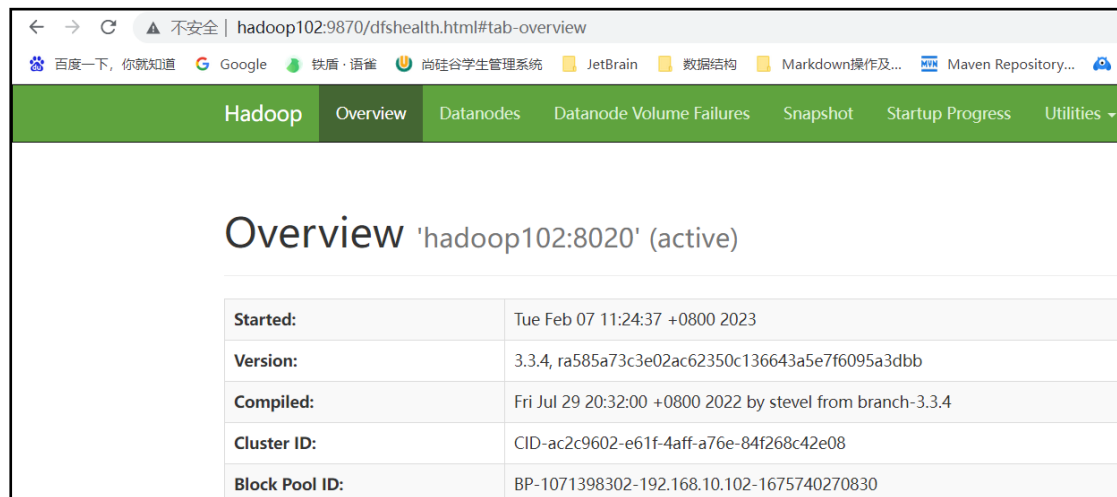
(2) 启动 HDFS

```
[iflytek@hadoop102 hadoop]$ sbin/start-dfs.sh
```

(3) **在配置了 ResourceManager 的节点（hadoop103）**启动 YARN

```
[iflytek@hadoop103 hadoop]$ sbin/start-yarn.sh
```

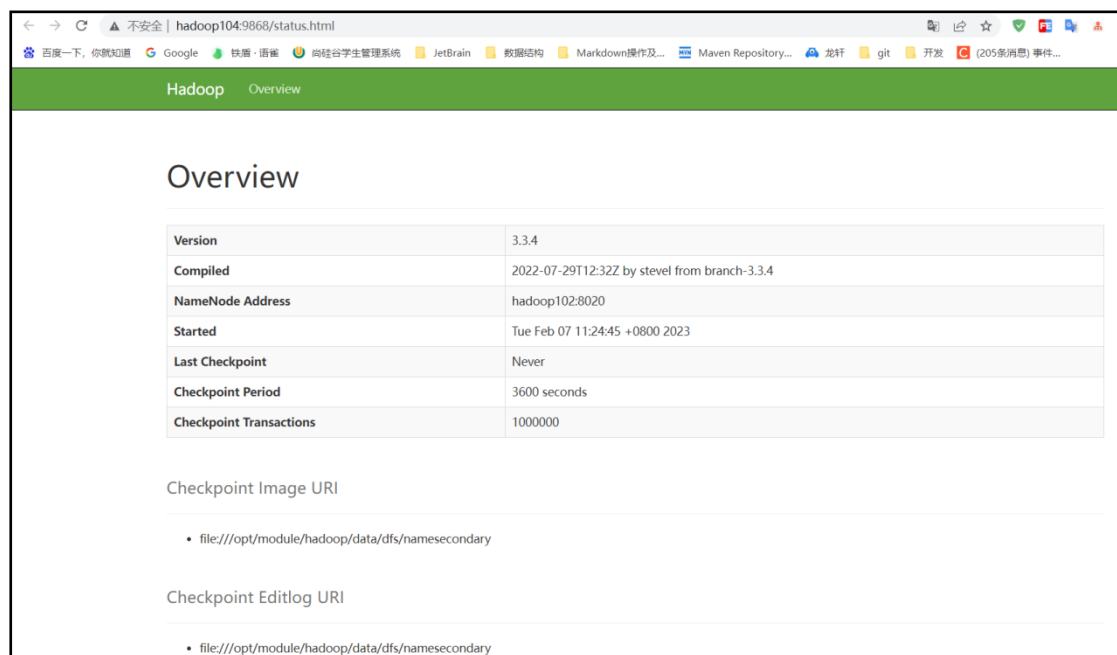
(4) Web 端查看 HDFS 的 Web 页面：<http://hadoop102:9870/>



(5) Web 端查看 SecondaryNameNode

① 浏览器中输入: <http://hadoop104:9868/status.html>

② 查看 SecondaryNameNode 信息



1.7 Hadoop 群起脚本

(1) 在/home/iflytek/bin 目录下创建 hdp.sh

```
[iflytek@hadoop102 bin]$ cd /home/iflytek/bin
[iflytek@hadoop102 bin]$ vim hdp.sh
```

(2) 写入以下内容:

```
#!/bin/bash
if [ $# -lt 1 ]
then
    echo "No Args Input..."
    exit ;
fi
```

```

case $1 in
"start")
    echo " ===== 启动 hadoop 集群 ====="

    echo " ----- 启动 hdfs -----"
    ssh hadoop102 "/opt/module/hadoop/sbin/start-dfs.sh"
    echo " ----- 启动 yarn -----"
    ssh hadoop103 "/opt/module/hadoop/sbin/start-yarn.sh"
    echo " ----- 启动 historyserver -----"
    ssh hadoop102 "/opt/module/hadoop/bin/mapred --daemon start
historyserver"
;;
"stop")
    echo " ===== 关闭 hadoop 集群 ====="

    echo " ----- 关闭 historyserver -----"
    ssh hadoop102 "/opt/module/hadoop/bin/mapred --daemon stop
historyserver"
    echo " ----- 关闭 yarn -----"
    ssh hadoop103 "/opt/module/hadoop/sbin/stop-yarn.sh"
    echo " ----- 关闭 hdfs -----"
    ssh hadoop102 "/opt/module/hadoop/sbin/stop-dfs.sh"
;;
*)
    echo "Input Args Error..."
;;
esac

```

(3) 增加权限

```
[iflytek@hadoop102 bin]$ chmod 777 hdp.sh
```

(4) 启动集群

```
[iflytek@hadoop102 bin]$ hdp.sh start
```

日志如下。

```

===== 启动 hadoop 集群 =====
----- 启动 hdfs -----
Starting namenodes on [hadoop102]
Starting datanodes
Starting secondary namenodes [hadoop104]
----- 启动 yarn -----
Starting resourcemanager
Starting nodemanagers
----- 启动 historyserver -----

```

查看进程。

```
[iflytek@hadoop102 bin]$ xcall jps
```

结果如下。

```

----- hadoop102 -----
3074 Jps
2116 NameNode
2245 DataNode
2761 JobHistoryServer
2590 NodeManager

```



```

----- hadoop103 -----
3270 NodeManager
2952 DataNode
3148 ResourceManager
3854 Jps
----- hadoop104 -----
1889 DataNode
2100 NodeManager
2446 Jps
1967 SecondaryNameNode

```

(5) 停止集群

```
[iflytek@hadoop102 bin]$ hdp.sh stop
```

日志如下

```

===== 关闭 hadoop 集群 =====
----- 关闭 historyserver -----
----- 关闭 yarn -----
Stopping nodemanagers
Stopping resourcemanager
----- 关闭 hdfs -----
Stopping namenodes on [hadoop102]
Stopping datanodes
Stopping secondary namenodes [hadoop104]

```

查看进程。

```
[iflytek@hadoop102 bin]$ xcall jps
```

结果如下。

```

----- hadoop102 -----
3691 Jps
----- hadoop103 -----
4221 Jps
----- hadoop104 -----
2647 Jps

```