3、部署Hadoop

Hadoop 分布式系统基础平台，主要存储计算任务的中间结果数据。

> **文档环境**
>
> 单机部署数据挖掘组件环境如下：

服务器IP	主机名	组件实例	部署目录
10.10.204.248	10-10-204-248	数据挖掘	/data
10.10.204.249	10-10-204-249	Spark，Hadoop	/data
10.10.204.250	10-10-204-250	Python	/data

> **注意事项**
>
> 数据挖掘数据量2000万以下时，可以不单独部署hadoop组件，提高数据挖掘服务器配置即可

1、系统环境准备

> **温馨提示**
>
> 配置防火墙，selinux相关操作，需要管理员权限。

1.1 防火墙配置

为了便于安装，建议在安装前关闭防火墙。使用过程中，为了系统安全可以选择启用防火墙，但必须启用服务相关端口。

1.关闭防火墙

临时关闭防火墙（立即生效）

```
systemctl stop firewalld
```

永久关闭防火墙（需重启后生效）

```
systemctl disable firewalld
```

查看防火墙状态

```
systemctl status firewalld
```

2.开启防火墙

相关服务及端口对照表：

服务名	需要开放端口
Hadoop	9864，9866，9867，9868，9870，9000

如果确实需要打开防火墙安装，需要给防火墙放开以下需要使用到的端口
开启端口：9864，9866，9867，9868，9870

```
firewall-cmd --zone=public --add-port=9864/tcp --permanent
firewall-cmd --zone=public --add-port=9866/tcp --permanent
firewall-cmd --zone=public --add-port=9867/tcp --permanent
firewall-cmd --zone=public --add-port=9868/tcp --permanent
firewall-cmd --zone=public --add-port=9870/tcp --permanent
firewall-cmd --zone=public --add-port=9000/tcp --permanent
```

配置完以后重新加载firewalld，使配置生效

```
firewall-cmd --reload
```

查看防火墙的配置信息

```
firewall-cmd --list-all
```

3.关闭selinux

临时关闭selinux，立即生效，不需要重启服务器。

```
setenforce 0
```

永久关闭selinux，修改完配置后需要重启服务器才能生效

```
sed -i 's/=enforcing/=disabled/g'  /etc/selinux/config
```

1.2 取消打开文件限制

修改/etc/security/limits.conf文件在文件的末尾加入以下内容：

```
vi /etc/security/limits.conf
```

在文件的末尾加入以下内容：

```
* soft nofile 65536
* hard nofile 65536
* soft nproc 131072
* hard nproc 131072
```

2、Hadoop单节点安装

> **温馨提示**
>
> 配置主机名映射，需要管理员权限。

2.1 配置主机名映射

将数据挖掘组件中的服务器主机名映射到hosts文件中

```
vi /etc/hosts
```

文件末尾添(根据实际环境信息设置，如果已设置，则无需重复)：

```
10.10.204.248 10-10-204-248
10.10.204.249 10-10-204-249
10.10.204.250 10-10-204-250
```

2.2 配置系统免密登录

> **说明**
>
> 文档中Spark与Hadoop部署在相同环境，则无需重复设置系统免密登陆

登陆服务器，生成密钥

```
ssh-keygen
```

输入ssh-keygen后，连续按三次回车，不用输入其它信息。

复制公钥到文件中：

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 0600 ~/.ssh/authorized_keys
```

测试是否设置成功

示例:

```
ssh root@10-10-204-249
```

如果不用输入密码，表示配置成功

2.3 安装JAVA环境

> **说明**
>
> 文档中Spark与Hadoop部署在相同环境，则无需重复设置JAVA环境

解压jdk到指定目录：

```
tar -zxvf jdk-8u181-linux-x64.tar.gz -C /data
```

添加环境变量

```
vi /etc/profile
```

在文件末尾添加下面内容：

```
export JAVA_HOME=/data/jdk1.8.0_181
export JAVA_BIN=$JAVA_HOME/bin
export CLASSPATH=:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export PATH=$PATH:$JAVA_BIN
```

让配置生效

```
source /etc/profile
```

验证安装

```
java -version
```

2.4 安装Hadoop

> **温馨提示**
>
> 1、安装部署、启动hadoop等，可以使用普通用户权限进行操作。
>
> 2、部署过程中用普通用户操作，则后续的所有运维操作等，都需要用普通用户来执行。如果切换其他用户操作，可能会因为权限问题导致服务启动失败。
>
> 3、hadoop的端口配置如果小于1024，也需要管理员权限才能启动服务。

2.4.1. 准备hadoop数据目录

创建临时目录

```
mkdir -p /data/hdfs/tmp
```

创建namenode数据目录

```
mkdir -p /data/hdfs/name
```

创建datanode 数据目录

注意：这个目录尽量创建在空间比较大的目录，如果有多个磁盘，可以创建多个目录

```
mkdir -p /data/hdfs/data
```

2.4.2. 解压Hadoop到安装目录

```
tar -zxvf hadoop-3.2.3.tar.gz -C /data
```

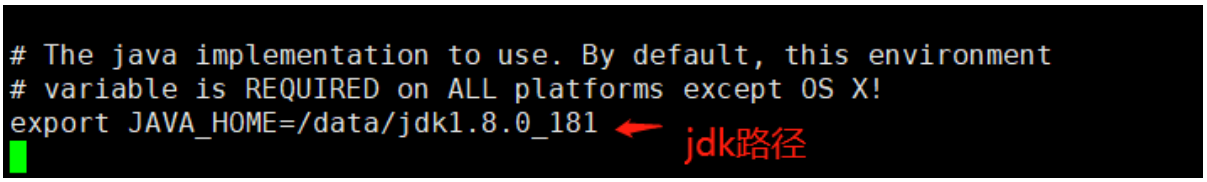
2.4.3. 修改hadoop配置

① 修改hadoop-env.sh

```
cd /data/hadoop-3.2.3/etc/hadoop
vi hadoop-env.sh
```

找到"export JAVA_HOME"，修改为如下所示(替换成实际环境的路径)：

```
export JAVA_HOME=/data/jdk1.8.0_181
```

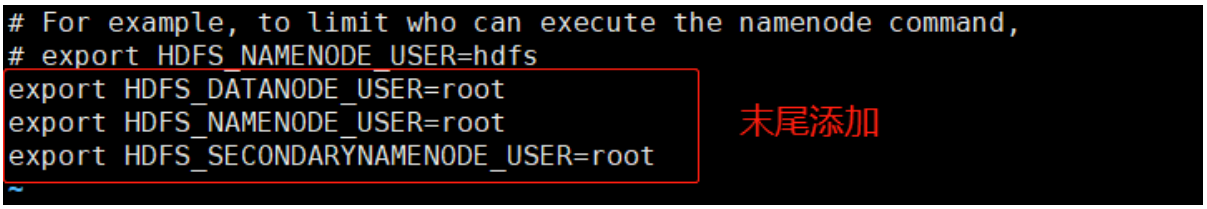


找到"export HDFS_NAMENODE_OPTS"，在下面添加一行

```
export HDFS_NAMENODE_OPTS="-XX:+UseParallelGC -Xmx4g"
```

添加启动用户， 在文件最后添加以下内容

```
export HDFS_DATANODE_USER=root
export HDFS_NAMENODE_USER=root
export HDFS_SECONDARYNAMENODE_USER=root
```



> ⚠ **关于启动用户**
>
> 启动用户可根据实际环境替换成实际的用户名

② 修改core-site.xml

```
cd /data/hadoop-3.2.3/etc/hadoop
vi core-site.xml
```

内容如下：

```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <!--  -->
        <value>hdfs://10-10-204-249:9000</value>
    </property>
    <property>
        <name>hadoop.tmp.dir</name>
        <!--  -->
        <value>file:/data/hdfs/tmp</value>
    </property>
    <property>
        <name>fs.trash.interval</name>
        <value>100800</value>
    </property>
    <property>
        <name>hadoop.security.authorization</name>
        <value>true</value>
    </property>
</configuration>
```

③ 修改hdfs-site.xml

```
cd /data/hadoop-3.2.3/etc/hadoop
vi hdfs-site.xml
```

内容如下：

```
<configuration>
    <property>
        <name>dfs.name.dir</name>
        <!--  -->
        <value>file:/data/hdfs/name</value>
    </property>
    <property>
        <name>dfs.data.dir</name>
        <!--  -->
        <value>file:/data/hdfs/data</value>
    </property>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.web.authentication.simple.anonymous.allowed</name>
        <value>false</value>
    </property>
    <property>
        <name>dfs.webhdfs.user.provider.user.pattern</name>
        <value>(?s)!(.+)</value>
    </property>
    <property>
        <name>dfs.datanode.max.transfer.threads</name>
        <value>16384</value>
    </property>
</configuration>
```

> ⚠ **建议**
>
> dfs.data.dir尽量配置在空间比较大的目录，可以配置多个目录，中间用逗号分隔

④ 修改hadoop-policy.xml

```
cd /data/hadoop-3.2.3/etc/hadoop
vi hadoop-policy.xml
```

内容如下:

```
<configuration>
  <property>
    <name>security.client.protocol.acl</name>
    <value>*</value>
    <description>ACL for ClientProtocol, which is used by user code
    via the DistributedFileSystem.
    The ACL is a comma-separated list of user and group names. The user and
    group list is separated by a blank. For e.g. "alice,bob users,wheel".
    A special value of "*" means all users are allowed.</description>
  </property>

  <!-- ip, pythonipsparkiphadoopipSmartbiMPPip-->
  <!-- ETL/SmartbiMPPhdfs-->
  <!--  -->
  <property>
    <name>security.client.protocol.hosts</name>
    <value>10.10.204.248,10.10.204.249,10.10.204.250</value>
  </property>
  <!-- end -->

  <property>
    <name>security.client.datanode.protocol.acl</name>
    <value>*</value>
    <description>ACL for ClientDatanodeProtocol, the client-to-datanode protocol
    for block recovery.
    The ACL is a comma-separated list of user and group names. The user and
    group list is separated by a blank. For e.g. "alice,bob users,wheel".
    A special value of "*" means all users are allowed.</description>
  </property>

  <!-- ip,pythonipsparkiphadoopipSmartbiMPPip-->
  <!-- ETL/SmartbiMPPhdfs-->
  <!--  -->
  <property>
    <name>security.client.datanode.protocol.hosts</name>
    <value>10.10.204.248,10.10.204.249,10.10.204.250</value>
  </property>
  <!-- end -->

  <property>
    <name>security.datanode.protocol.acl</name>
    <value>*</value>
    <description>ACL for DatanodeProtocol, which is used by datanodes to
    communicate with the namenode.
    The ACL is a comma-separated list of user and group names. The user and
    group list is separated by a blank. For e.g. "alice,bob users,wheel".
    A special value of "*" means all users are allowed.</description>
  </property>

  <!-- hadoop-policy.xml -->
  <!-- hadoop-policy.xml -->
  <!-- ... -->

</configuration>
```

注意

hadoop-policy.xml配置文件仅添加两处配置项；

新增的security.client.protocol.hosts，security.client.datanode.protocol.hosts两个配置项中的值，要替换成实际环境的IP地址；

此配置文件是限制可以访问hadoop节点的服务器ip，提高hadoop应用的安全性。

2.4.4. 配置hadoop环境变量

```
vi /etc/profile
```

在文件末尾添加下面内容：

```
export HADOOP_HOME=/data/hadoop-3.2.3
export PATH=$PATH:$HADOOP_HOME/bin
```

让配置生效

```
source /etc/profile
```

2.4.5. 启动Hadoop

①格式化hadoop

```
cd /data/hadoop-3.2.3/
./bin/hdfs namenode -format
```

> ⚠ 仅第一次启动时需要执行格式化Hadoop操作，后续启动无需进行此操作

②启动hadoop

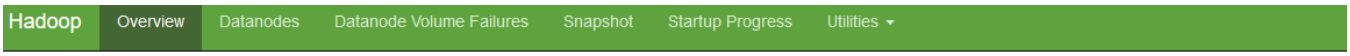
```
cd /data/hadoop-3.2.3/
./sbin/start-dfs.sh
```

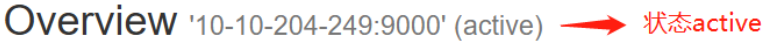
③创建中间数据存储目录

```
hdfs dfs -mkdir /mining
hdfs dfs -chown mining:mining /mining
```

2.4.6. 验证安装

①在浏览器输入： http://HadoopIP:9870/dfshealth.html#tab-overview 检查集群状态





Started:	Wed May 12 14:52:23 +0800 2021
Version:	3.2.2, r7a3bc90b05f257c8ace2f76d74264906f0f7a932
Compiled:	Sun Jan 03 17:26:00 +0800 2021 by hexiaoqiao from branch-3.2.2
Cluster ID:	CID-fd6e69c9-6362-4d2a-9af7-1f2e4c171b77
Block Pool ID:	BP-1820793709-10.10.204.249-1620802331424

Summary

Security is off.

Safemode is off.

2 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 2 total filesystem object(s).

Heap Memory used 210.24 MB of 397.5 MB Heap Memory. Max Heap Memory is 3.56 GB.

Non Heap Memory used 54.69 MB of 55.86 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

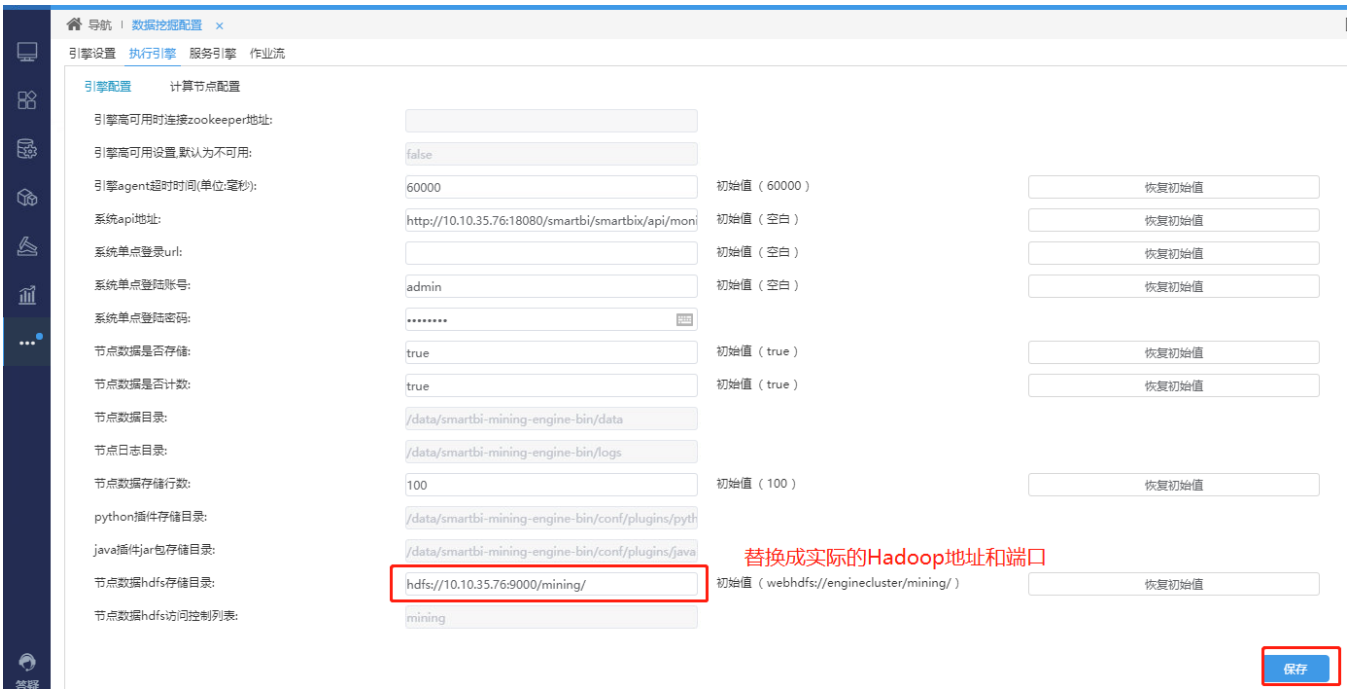
Configured Capacity:	492.03 GB
Configured Remote Capacity:	0 B
DFS Used:	24 KB (0%)
Non DFS Used:	87.54 GB
DFS Remaining:	379.47 GB (77.12%)
Block Pool Used:	24 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0) 此处为1
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0

②检查mining目录是否创建成功

```
hdfs dfs -ls /    #/mining
```

2.5 Smartbi连接Hadoop

打开**系统运维－数据挖掘配置－执行引擎—引擎配置**，参考下图修改hadoop地址(根据实际环境修改)，修改完成后点击保存：



配置完成后可参考： 测试数据挖掘及其组件 运行数据挖掘实验

2.6 运维操作

停止hadoop

```
cd /data/hadoop-3.2.3/
./sbin/stop-dfs.sh
```

启动hadoop

```
cd /data/hadoop-3.2.3/
./sbin/start-dfs.sh
```

查看日志
hadoop的日志路径：/data/hadoop-3.2.3/logs
安装部署或者使用中有问题，可能需要根据日志来分析解决。