【数据挖掘】V9.5升级到V10版本

数据挖掘从 V9.5 升级到 V10 版本的升级内容如下：

数据挖掘组件	V9.5版本	V10版本	更新内容	备注
实验引擎	√	√	数据挖掘引擎版本更新，添加执行引擎-一键推荐计算机点配置	
服务引擎	√	√	数据挖掘引擎版本更新	
Spark	√	√	Spark版本由2.4升级到3.1.3版本	需要使用Smartbi提供的安装包
Python执行节点	√	√	数据挖掘引擎版本更新，新增Python代理程序启动用户。	
Hadoop	×	√	V9.6版本中，新增Hadoop组件，用于节点中间数据存储。	需要使用Smartbi提供的安装包

一、数据挖掘引擎包更新

获取新版本的数据挖掘引擎安装包。

新数据挖掘引擎安装包解压缩后；

先备份<数据挖掘安装目录>/engine目录；

再删除<数据挖掘安装目录>/engine目录，然后上传新的engine目录，并重启数据挖掘引擎。

数据挖掘引擎安装包版本要和smartbi的war包版本一致，更新时需要同步更新Python节点中的引擎包。

二、Spark版本升级

> **注意事项**
>
> 需要使用Smartbi提供的Spark3.1.3安装包部署

1. 停止旧版本Spark

进入spark安装目录，执行命令停止spark2.4服务。

```
cd /data/spark-2.4.0-bin-hadoop2.7/sbin/     #spark
./stop-all.sh
```

> **注意事项**
>
> 注意，如果出现无法停止情况，可以通过jps查看Spark服务(Spark的进程名有Master,Worker,CoarseGrainedExecutorBackend)进程id，然后kill -9 进程id

2.配置系统免密登陆

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

3.安装Spark

解压Spark到指定目录

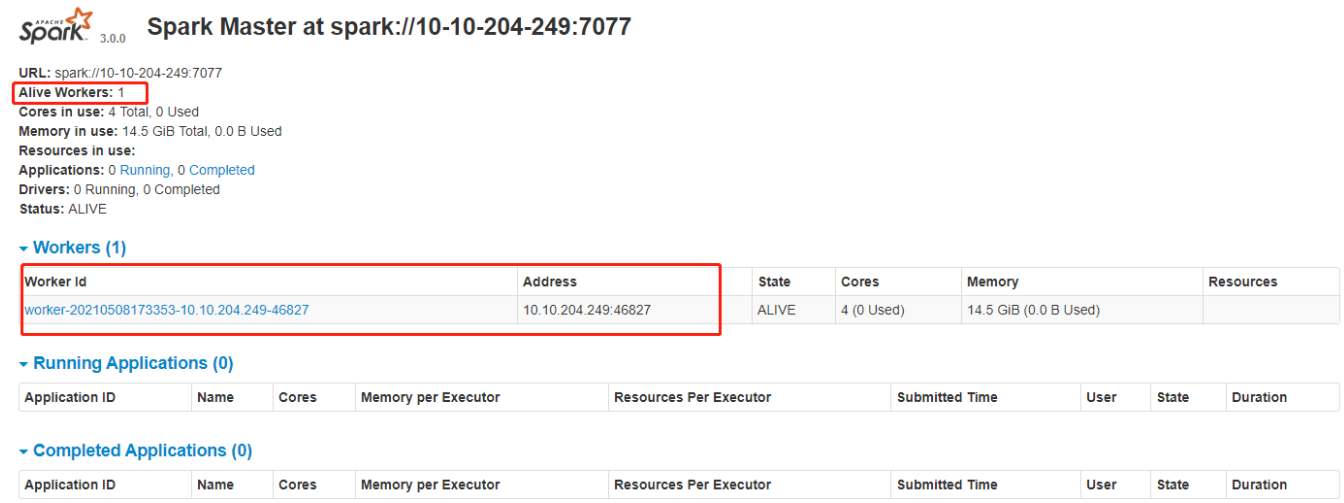
```
tar -zxvf spark-3.1.3-bin-hadoop3.2.tgz -C /data
```

启动Spark

```
cd /data/spark-3.1.3-bin-hadoop3.2.tgz/sbin
 ./start-all.sh
```

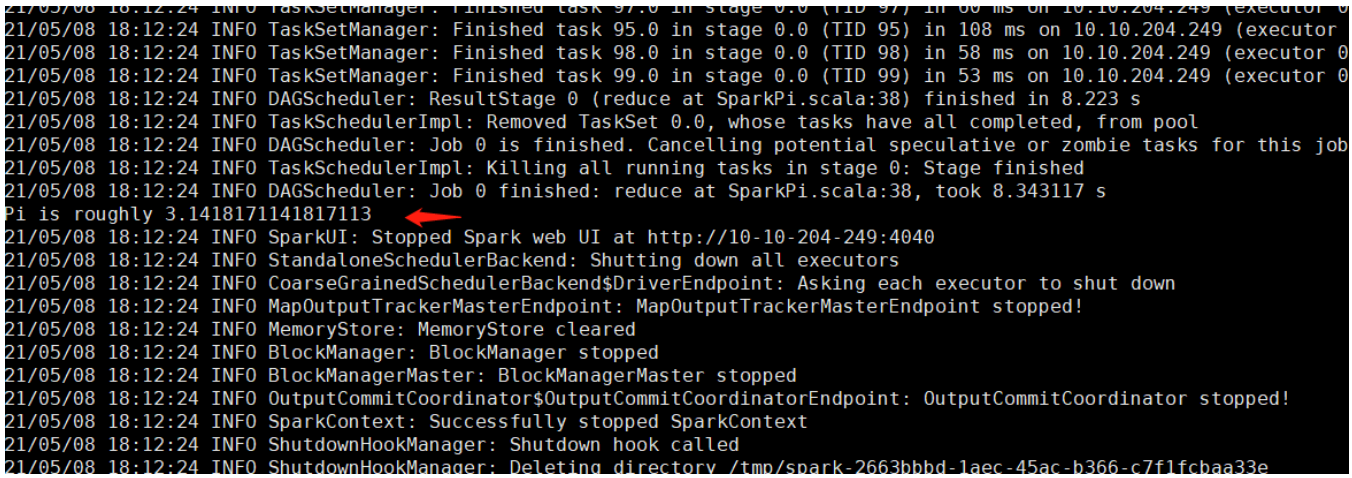
4. 检查Spark

在浏览器中输入：http://master节点的ip:8080，查看集群状态



在spark节点提交任务测试进入/data/spark-3.1.3-bin-hadoop3.2/bin目录，执行以下命令(注意将”节点IP”替换对应的IP或主机名)

```
./spark-submit --class org.apache.spark.examples.SparkPi --master spark://IP:7077 /data/spark-3.1.3-bin-hadoop3.2/examples/jars/spark-examples_2.12-3.0.0.jar 100
```



运行得出圆周率Pi的近似值3.14即部署成功。

5. 运维操作

启动/停止spark服务

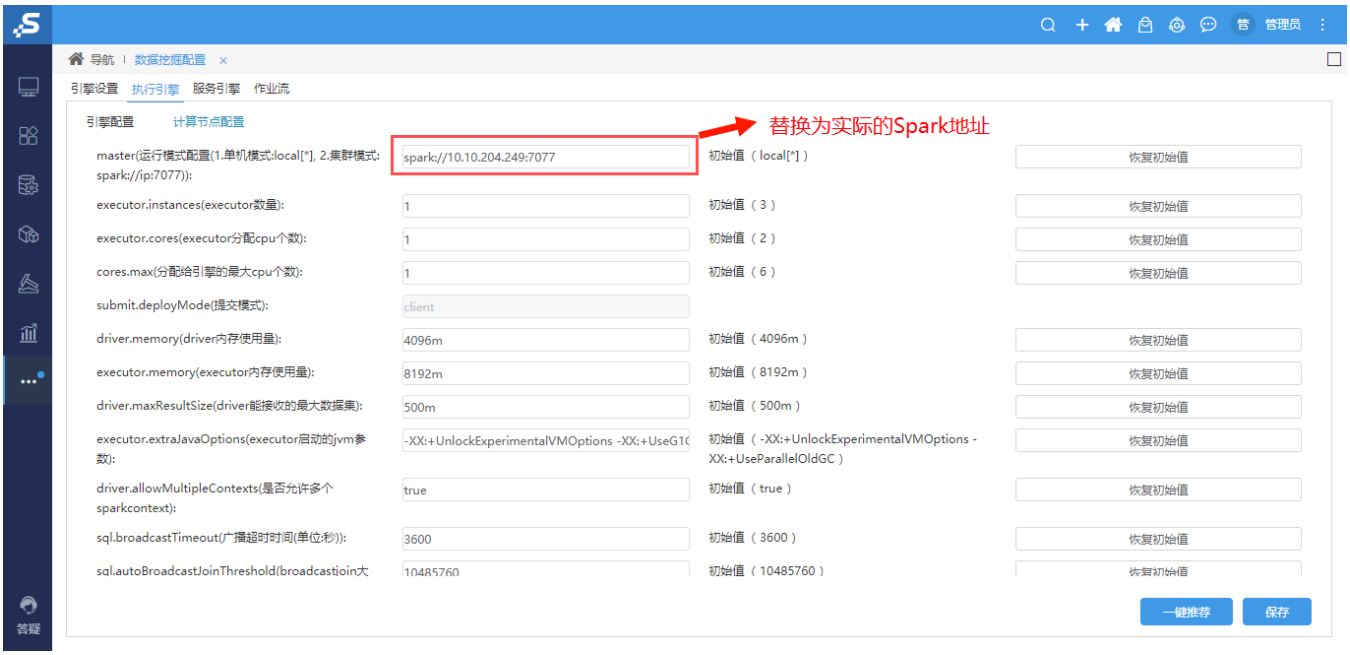
```
cd /data/spark-3.1.3-bin-hadoop3.2/sbin
./start-all.sh    #spark
./stop-all.sh     #spark
```

Spark集群部署参考文档：部署Spark集群

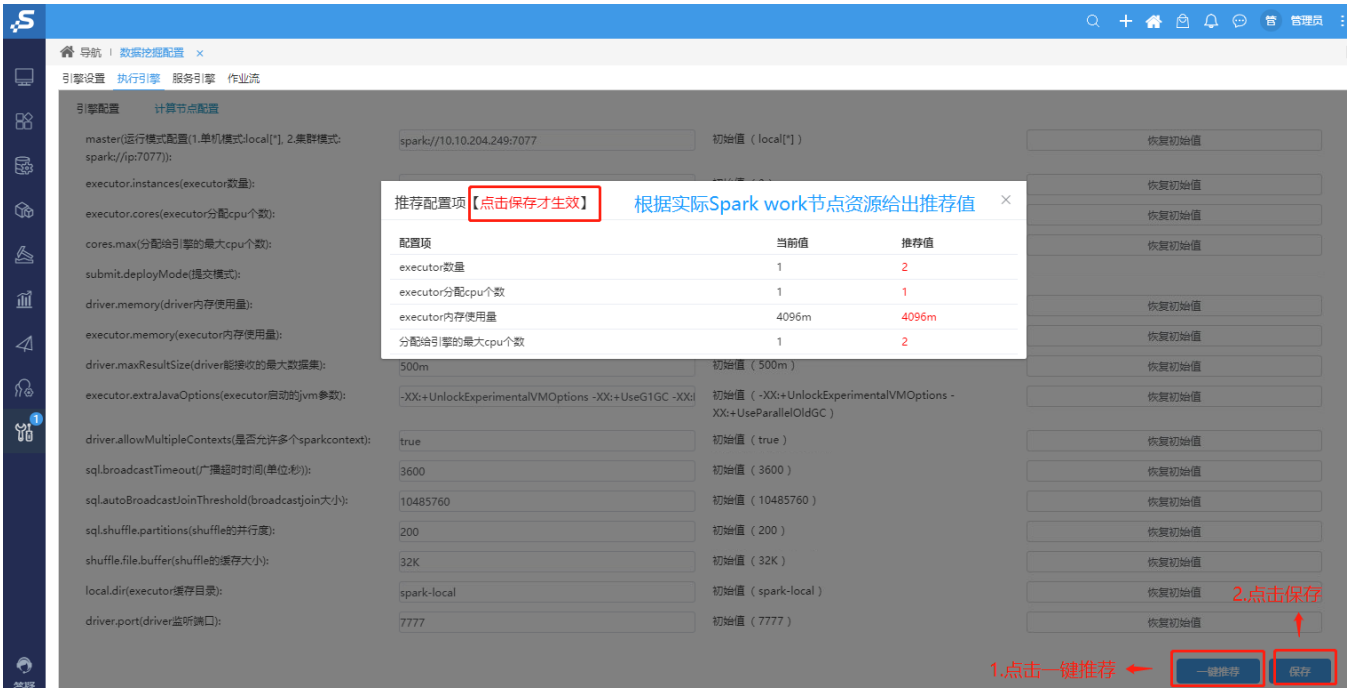
6. Smartbi连接Spark

浏览器访问smartbi，打开**系统运维－数据挖掘配置－执行引擎—计算节点配置**，参考下图设置，修改完成后点击保存

配置spark计算节点：



配置Spark节点资源，点击一键推荐，系统会根据Spark work节点的服务器资源，生成推荐的配置(如果使用推荐值，记得点击保存，否则配置不生效)：



Spark升级配置完成。

三、部署Hadoop组件

在 Smartbi V9.6 版本中，数据挖掘新增了Hadoop组件，用于存储挖掘实验的节点中间数据。【Hadoop非必选组件，可根据需求选择是否安装】

若是当前 Hadoop 版本低于3.2.2，则需参考以下步骤备份数据，再参考本章节文档部署 Hadoop 3.2.2版本。

（1）停止hadoop

```
cd /data/hadoop-2.7.3/
./sbin/stop-dfs.sh
```

（2）迁移数据

```
#
mkdir -p /data/backups
#
mv /data/hdfs/ /data/backups
# Hadoop
mv /data/hadoop-2.7.3/ /data/backups
```

1、系统环境准备

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

```
systemctl stop firewalld
```

永久关闭防火墙

```
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

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
```

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

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)：

```
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

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

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

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-3.2.2.tar.gz -C /data
```

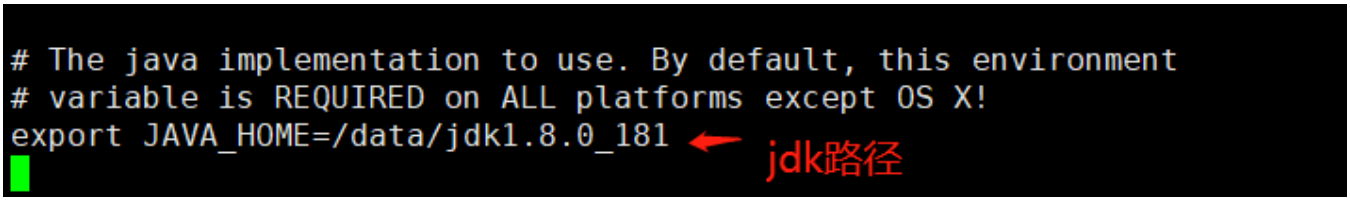
2.4.3. 修改hadoop配置

① 修改hadoop-env.sh

```
cd /data/hadoop-3.2.2/etc/hadoop
vi hadoop-env.sh
```

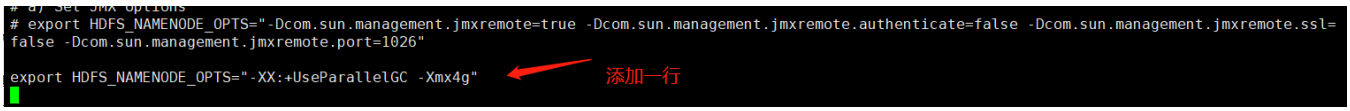
找到"export JAVA_HOME"，修改为如下所示(替换成实际环境的路径)：

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



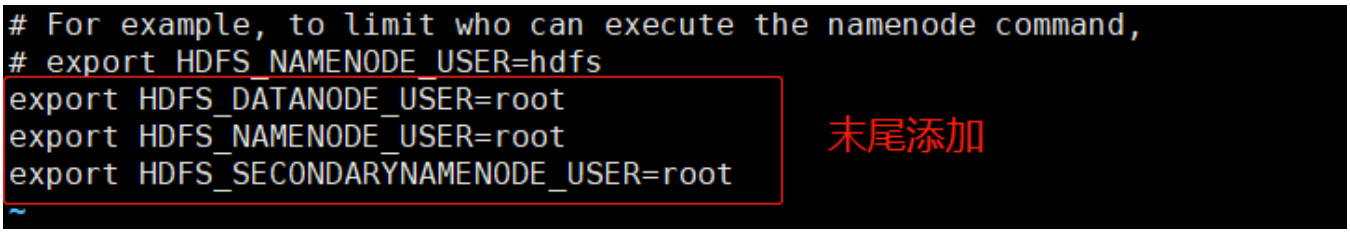
找到"export HDFS_NAMENODE_OPTS"，在下面添加一行

```
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-3.2.2/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>
```

④ 修改hdfs-site.xml

```
cd /data/hadoop-3.2.2/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.webhdfs.enabled</name>
        <value>false</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

⑤ 修改hadoop-policy.xml

```
cd /data/hadoop-3.2.2/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, pythonipsparkiphadoopip-->
  <!--  -->
  <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,pythonipsparkiphadoopip-->
  <!--  -->
  <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-3.2.2
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-3.2.2/
./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

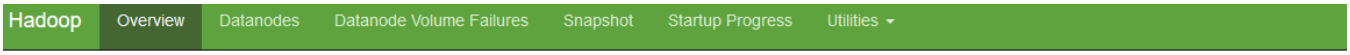
```
cd /data/hadoop-3.2.2/
./sbin/start-dfs.sh
```

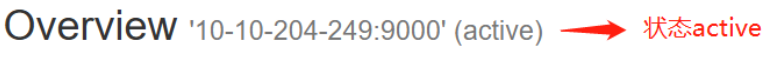
③ 创建中间数据存储目录

```
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 运维操作

停止hadoop

```
cd /data/hadoop-3.2.2/
./sbin/stop-dfs.sh
```

启动hadoop

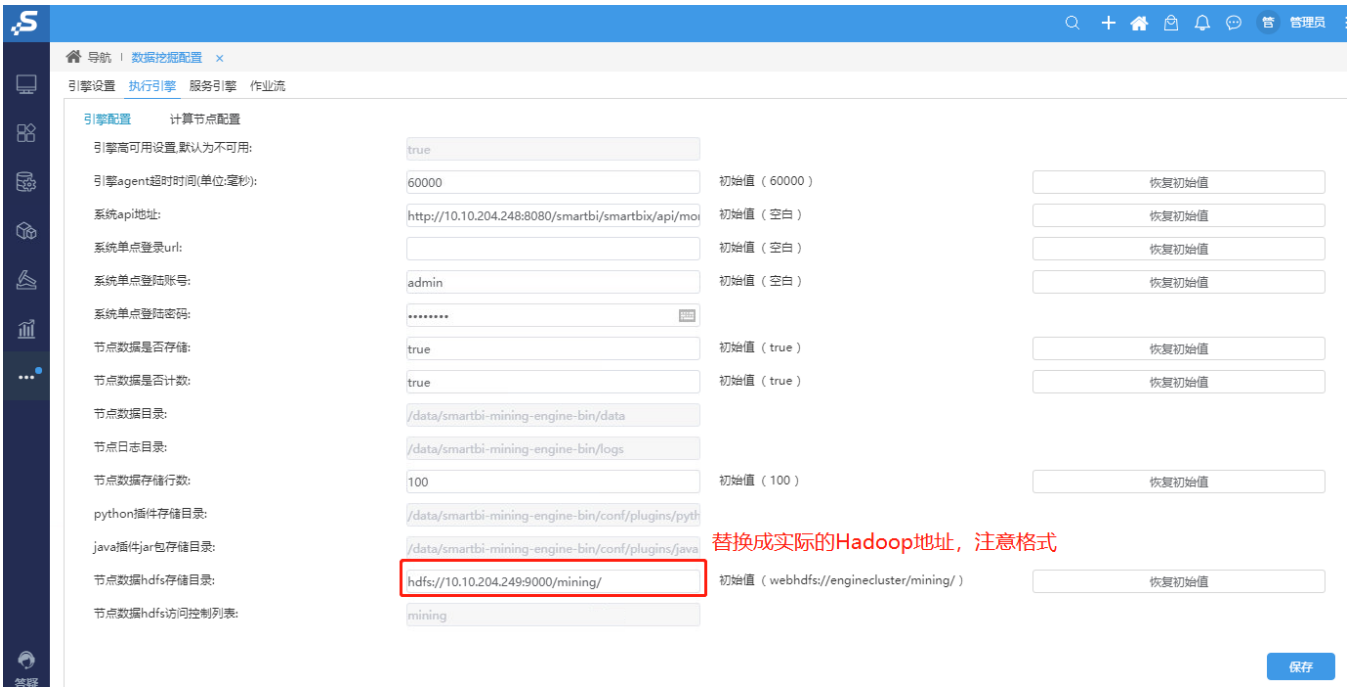
```
cd /data/hadoop-3.2.2/
./sbin/start-dfs.sh
```

查看日志
hadoop的日志路径：/data/hadoop-2.7.3/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。

Hadoop集群部署参考：部署Hadoop集群

3、设置执行引擎连接Hadoop

浏览器访问Smartbi，打开**系统运维—数据挖掘配置－执行引擎－引擎配置** 找到“节点数据hdfs存储目录”配置项，填写Hadoop地址



注意事项

如果是Hadoop集群，上图中节点数据hdfs存储目录需要填写Hadoop管理节点的IP

四、Python执行节点更新

1. 停止旧Python服务

进入安装Python计算节点的服务器，进入目录，停止Python服务

```
cd /opt/smartbi-mining-engine-bin/engine/sbin
./python-daemon.sh stop
```

注意事项

注意，如果出现无法停止情况，可以通过jps查看python服务进程id，然后 kill -9 进程id

如果V9.5版本已经部署Python执行节点，Python版本无需额外升级，沿用之前的即可（应为Python 3.7.4）

2. 更新引擎包

更新方式，参考数据挖掘的更新方式，如果python执行节点跟实验引擎在同台机器，这步骤可以省略

3. 添加主机名映射

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)：

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

4. 创建Python执行用户

创建用户组、用户并设置密码

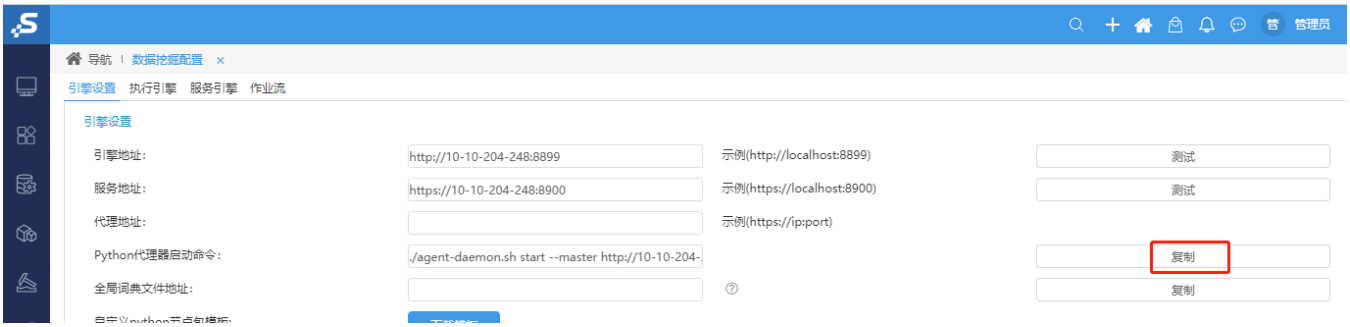
```
groupadd mining                  #mining
useradd -g mining mining-ag      #(mining-ag)mining
passwd mining-ag                 #mining-ag
```

给引擎安装目录附权限(为了使用mining-ag用户启动执行代理程序时候，有权限创建agent-data跟agent-logs目录)

```
chgrp mining /data/smartbi-mining-engine-bin
chmod 775 /data/smartbi-mining-engine-bin
```

5. 启动Python执行代理

① 浏览器访问Smartbi，打开**系统运维－数据挖掘配置－引擎设置**，复制Python代理器启动命令



> **注意事项**
>
> 复制Python代理器启动命令前，请确认数据挖掘引擎能正常测试连接成功

② 登录到部署Python节点机器，并切换到mining-ag用户

> **注意事项**
>
> 为了避免出现安全问题，一定要切换到mining-ag用户去启动执行代理服务，不要使用root用户安装或带有sudo权限的用户来启动执行代理服务

```
su - mining-ag
```

进入引擎启动目录

```
cd /data/smartbi-mining-engine-bin/engine/sbin
```

把拷贝命令粘贴，并执行，例如：

```
./agent-daemon.sh start --master http://10-10-204-248:8899 --env python
```

等待Python节点启动成功即可。

6. 运维操作

1、更新Python数据挖掘引擎包

Smartbi更新war包版本时，Python执行节点需要同步更新对应版本的数据挖掘引擎。

Python节点集群部署参考：部署Python节点集群