

4、部署Python节点

- 1. 系统环境准备
 - 1.1. 防火墙配置
- 2. Python计算机点单机部署
 - 2.1. 安装前配置
 - 2.2. 安装Python主程序
 - 2.3. 离线安装Python插件包
 - 2.4. 部署数据挖掘引擎包
 - 2.5. 创建Python执行用户
 - 2.6. 启动Python执行代理
 - 2.7. 运维操作

Python节点主要用于机器学习的DBSACN算法和自定义模块的Python脚本扩展。如果没有用到这两个功能模块，可以不用部署python节点。



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

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。

1.1.1. 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（重启后生效）

```
systemctl disable firewalld
```

查看防火墙状态

```
systemctl status firewalld
```

1.1.2. 开启防火墙

相关服务及端口对照表：

服务名	需要开放端口
Python	8980

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：8980

```
firewall-cmd --zone=public --add-port=8980/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
```

1.1.3. 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
```

2. Python计算机点单机部署

2.1. 安装前配置



温馨提示

配置本地yum源，主机名映射，需要管理员权限。

① 配置本地yum源，参考文档：<https://jingyan.baidu.com/article/fdbd4277a508e8b89e3f4807.html>



注意事项

Centos7.4 以上，可不设置本地yum源。

② 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
```

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
```

在文件末尾添加以下内容：

```
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
```

2.2. 安装Python主程序



温馨提示

安装rpm包，需要管理员权限。

上传安装包到服务器的/data目录

① 安装依赖包

```
rpm -ivh bzip2-1.0.6-13.el7.x86_64.rpm
yum install gcc
```



注意事项

Centos7.4 以上，无需安装依赖包

② 安装Anaconda3并指定安装目录为/data/anaconda3



温馨提示

1、安装部署、启动python节点等，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。

```
cd /data
chmod +x Anaconda3-2019.10-Linux-x86_64.sh
bash Anaconda3-2019.10-Linux-x86_64.sh -b -p /data/anaconda3
```

③ 配置环境变量

```
vi /etc/profile
```

在末尾添加以下内容：

```
export PATH=/data/anaconda3/bin:$PATH
```

使配置生效，查看python版本

```
source /etc/profile
python --version
```

显示Python 3.7.4 表示安装成功。

2.3. 离线安装Python插件包

① 上传离线压缩包到服务器/data目录，并解压

```
cd /data
unzip plugin.zip
```

② 批量安装

安装时需指定解压出来的目录“./plugin”和文件“requirements.txt”的路径

```
cd /data
pip install --no-index --find-links=./plugin/ -r requirements.txt
```

2.4. 部署数据挖掘引擎包

上传安装包到服务器，并解压到指定目录

```
tar -zxvf SmartbiMiningEngine-V10.0.64186.21183.tar.gz -C /data
```



注意事项

数据挖掘引擎包版本需要与Smartbi版本一致。

2.5. 创建Python执行用户



温馨提示

1、创建用户需要使用管理员权限，如果已经是普通用户执行部署操作，可以不用再创建用户。

创建用户组、用户并设置密码

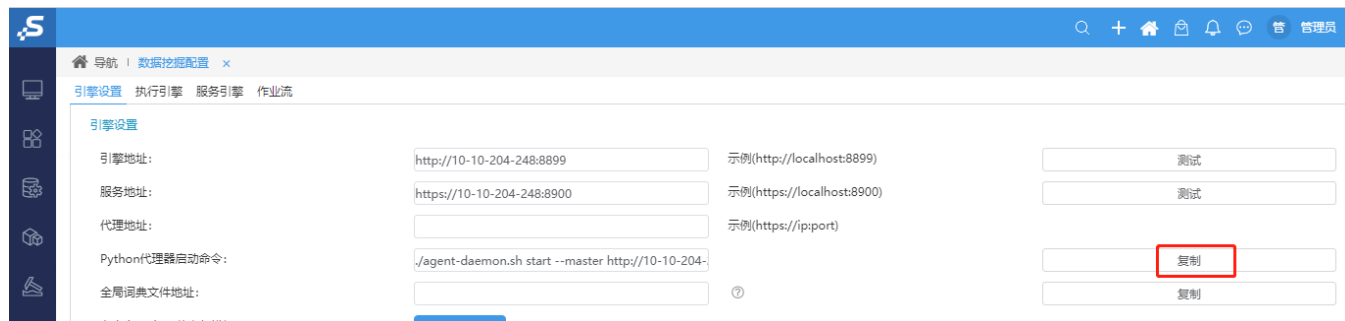
```
groupadd mining                #mining
useradd -g mining mining-ag    #(mining-ag)mining
passwd mining-ag               #mining-ag
```

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

2.6. 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，因为执行代理服务是会去执行在界面上定义的python脚本，如果执行用户权限过大，会存在安全问题。

```
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
```

若有换包更新数据挖掘版本，需要重启python代理

启动完成后可参考：[测试数据挖掘及其组件](#) 运行数据挖掘实验

2.7. 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。