# 一、事前准备

## 1.操作系统

**linux :CentOS 7**

## 2.运行环境

* Oracle JDK1.8
* Kafka成功安装，并能连接
* elasticsearch成功安装，并能连接
* SQL Server能够正确连接

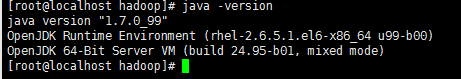
## 3.JDK安装部署

**注: 如果已经成功安装Oracle 1.8，则无需再进行操作!**

**说明: 一般CentOS自带了openjdk，但是我们要使用的是oracle官方的jdk，所以先行卸载CentOS的jdk，然后再安装在oracle下载好的JDK。**

首先输入 java -version

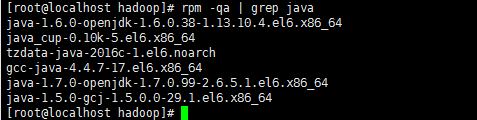
**查看是否安装了JDK，如果显示的是OpenJDk，就卸载**



输入

rpm -qa | grep java

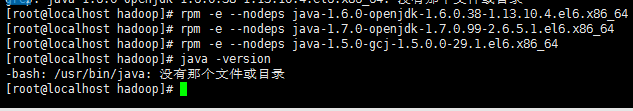
查看信息



然后输入:

rpm -e --nodeps “你要卸载JDK的信息”

如: rpm -e --nodeps java-1.7.0-openjdk-1.7.0.99-2.6.5.1.el6.x86\_64



确认没有了之后，解压下载下来的JDK

tar -xvf jdk-8u144-linux-x64.tar.gz

移动到opt/java文件夹中，没有就新建，然后将文件夹重命名为jdk1.8。

例如:

mv jdk1.8.0\_144 /opt/java

mv jdk1.8.0\_144 jdk1.8

然后编辑 profile 文件，添加如下配置

输入:

vim /etc/profile

添加:

例如:

export JAVA\_HOME=/opt/java/jdk1.8

export JRE\_HOME=/opt/java/jdk1.8/jre

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

export PATH=.:${JAVA\_HOME}/bin:$PATH

添加成功之后，输入

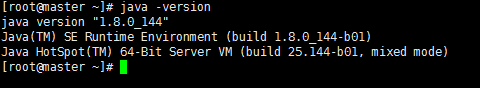
source /etc/profile

使配置生效之后

输入:

java -version

查看是否配置成功 出现 Java SE 标识，则表示Oracle JDK安装成功



# 二、项目部署

## 1.文件上传及授权

将项目通过FTP上传到 **/home/jars/fuxinServer** 目录下(没有该目录就新建)，并且该目录需要授权。

授权命令示例:

**chmod 777 /home/jars/fuxinServer**

该目录下应该包含如下文件**application.properties、logback.xml、fuxinServer.jar**。

## 2.修改配置文件

以下文件中红色部分需要修改，其他部分最好保持不变。

* **applicatiom.properties**

**具体配置说明如下:**

# 设置编码格式

banner.charset=UTF-8

server.tomcat.uri-encoding=UTF-8

spring.http.encoding.charset=UTF-8

spring.http.encoding.enabled=true

spring.http.encoding.force=true

spring.messages.encoding=UTF-8

## 默认的SqlServer数据源配置

spring.datasource.url=jdbc:sqlserver://192.169.2.204:1433;DatabaseName=pancm\_DB

spring.datasource.username=sa

spring.datasource.password=123zxcvbnm,./

spring.datasource.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver

## es配置

es:

elasticIps:http\://192.169.0.24\:9200

clusterName: bigData-cluster

## kafka地址配置

kafka.bootstrap\_servers = 192.169.0.23\:9092|192.169.0.24\:9092|192.169.0.25\:9092

kafka.topic\_name = PB\_BASIC\_PHONE

## Mybatis 配置

# 配置为 com.pancm.bean 指向实体类包路径。

mybatis.typeAliasesPackage=com.montnets.bean

# 配置为 classpath 路径下 mapper 包下，\* 代表会扫描所有 xml 文件。

mybatis.mapperLocations=classpath\:mapper/\*.xml

## 打印SQL语句

logging.level.com.montnets.dao=debug

## 自定义端口 默认 8080

server.port=8080

**备注:**

**1.spring.datasource.url为SQL Server数据库的路径。若迁入生产环境，则需改成生产环境的SQL Server 黑名单数据库地址 。**

**2.elasticIps为 elasticsearch 的连接Master地址,clusterName是es集群的名称。elasticIps为集群IP，多个请用逗号隔开 若迁入生产环境，则需改成生产环境的elasticsearch 地址。**

**3.server.port 是该服务的端口。**

**4.kafka.bootstrap\_servers是kafka集群的地址，kafka.topic\_name是kafka 消息队列的名称。若迁入生产环境，则需改成生产环境对应的地址和消息队列名称。**

## 3.程序启动和停止

* 启动

在linux 中切换到程序目录

输入:

cd /home/jars/fuxinServer

然后输入 ：

nohup java -jar fuxinServer.jar >/dev/null 2>&1 &

启动程序之后回车，如果输入jps查看到jar程序正在运行并且在**/home/jars/fuxinServer/logs**目录下的**mylog-\*.txt**查看打印日志出现程序成功启动信息，则表示程序已经成功启动！

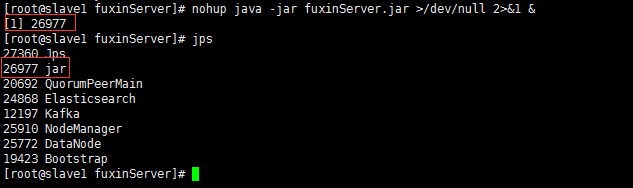
命令示例：tail -500f /home/jars/fuxinServer/logs/mylog\*.txt

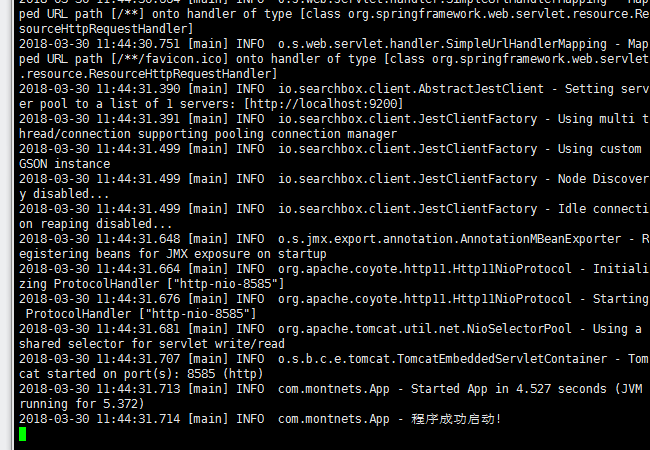
注:如果有多个mylog日志， mylog后面的\*应以实际天数位置。

例如:

tail -500f /home/jars/fuxinServer/logs/mylog-2018-03-30.0.txt

示例图如下:



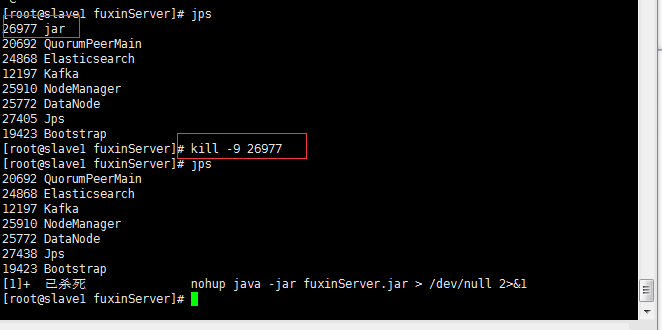


* 停止

如果需要停止该程序，在linux 中 输入jps 命令，查看jar的运行进程，然后输入 kill -9 jar的进程。

进程杀死后，可以输入jps命令查看。

例如:



## 4.出现的错误解决办法

### 1.程序启动之后，日志文件中出现地址已经在使用

说明该端口已经被占用，将配置文件的port端口换成未被使用的就行了!

### 2.程序启动之后，DB或ES连接失败!

首先检查配置文件的中地址是否正确，然后在检查该服务器是否能够正确和DB或ES服务器ping通，若都没问题，再次启动程序!