**自定义平台安装指导说明书**



[1. 平台服务对象程序介绍和环境部署 5](#_Toc29165)

[1.1. 平台各服务对象程序介绍 5](#_Toc3092)

[1.2. 平台各服务组网图 6](#_Toc30328)

[1.3. 平台安装服务器要求 7](#_Toc29172)

[2. 服务安装配置 7](#_Toc24565)

[2.1. 服务器环境初始化 7](#_Toc839)

[2.1.1. 关闭selinux和防火墙 7](#_Toc25922)

[2.1.2. 创建自定义平台安装用户 7](#_Toc24488)

[2.1.3. 配置依赖服务yum源 7](#_Toc17123)

[2.2. JDK服务 8](#_Toc18331)

[2.2.1. 安装JDK1.8 8](#_Toc32044)

[2.2.2. 修改配置文件 8](#_Toc25720)

[2.2.3. 启动及验证 8](#_Toc23815)

[2.3. MySQL服务 8](#_Toc21681)

[2.3.1. 安装MySQL服务 8](#_Toc17313)

[2.3.2. 启动及验证 9](#_Toc18228)

[2.4. Redis服务 9](#_Toc5312)

[2.4.1. 配置Redis服务系统参数 9](#_Toc1789)

[2.4.2. 安装Redis服务 10](#_Toc13461)

[2.4.3. 修改配置文件 10](#_Toc8209)

[2.4.4. 服务启动和验证 10](#_Toc15681)

[2.5. Zookeeper服务 11](#_Toc20857)

[2.5.1. 安装Zookeeper服务 11](#_Toc29033)

[2.5.2. 修改配置文件 11](#_Toc23548)

[2.5.3. 服务启动和验证 11](#_Toc23697)

[2.6. ActiveMQ服务 12](#_Toc31643)

[2.6.1. 安装ActiveMQ 12](#_Toc22287)

[2.6.2. 服务启动和验证 12](#_Toc2605)

[2.7. Canal服务 13](#_Toc4821)

[2.7.1. 安装Canal服务 13](#_Toc31620)

[2.7.2. 相关配置 13](#_Toc16238)

[2.7.3. 服务启动和验证 13](#_Toc10238)

[2.8. 数据接口服务 14](#_Toc1746)

[2.8.1. 安装数据接口服务 14](#_Toc16506)

[2.8.2. 相关配置 14](#_Toc29192)

[2.8.3. 服务启动和验证 15](#_Toc24318)

[2.9. 系统配置服务 15](#_Toc31662)

[2.9.1. 安装系统配置服务 15](#_Toc408)

[2.9.2. 相关配置 16](#_Toc2010)

[2.9.3. 服务启动和验证 16](#_Toc26945)

[2.10. 基础数据服务 17](#_Toc29686)

[2.10.1. 安装基础数据服务 17](#_Toc25796)

[2.10.2. 相关配置 17](#_Toc18051)

[2.10.3. 服务启动和验证 17](#_Toc4091)

[2.11. 流程引擎服务 18](#_Toc1510)

[2.11.1. 安装流程引擎服务 18](#_Toc4578)

[2.11.2. 相关配置 19](#_Toc13993)

[2.11.3. 服务启动和验证 19](#_Toc30797)

[2.12. 数据库监测服务 20](#_Toc17636)

[2.12.1. 安装数据库监测服务 20](#_Toc1607)

[2.12.2. 相关配置 20](#_Toc31129)

[2.12.3. 服务启动和验证 21](#_Toc12895)

[2.13. 数据级联服务 21](#_Toc25677)

[2.13.1. 安装数据级联服务 21](#_Toc15473)

[2.13.2. 相关配置 22](#_Toc8749)

[2.13.3. 服务启动和验证 22](#_Toc16047)

[2.14. 搜索引擎工具 23](#_Toc30580)

[2.14.1. 修改服务器参数 23](#_Toc31733)

[2.14.2. 安装搜索引擎工具 23](#_Toc5363)

[2.14.3. 相关配置 23](#_Toc32169)

[2.14.4. 服务启动和验证 23](#_Toc29416)

[2.15. 搜索与计算引擎服务 24](#_Toc26040)

[2.15.1. 安装搜索与计算引擎服务 24](#_Toc13195)

[2.15.2. 相关配置 25](#_Toc15735)

[2.15.3. 服务启动和验证 25](#_Toc21238)

[2.16. 同步服务 28](#_Toc4194)

[2.16.1. 同步服务介绍 28](#_Toc22353)

[2.16.2. 安装同步服务 28](#_Toc23066)

[2.16.3. 相关配置 28](#_Toc2927)

[2.16.4. 服务启动和验证 29](#_Toc18704)

[2.17. FileBeat服务 29](#_Toc24381)

[2.17.1. 安装FileBeat服务 29](#_Toc9146)

[2.17.2. 相关配置 29](#_Toc9388)

[2.17.3. 服务启动和验证 30](#_Toc19530)

[2.18. 定时任务服务 30](#_Toc21767)

[2.18.1. 安装定时服务 30](#_Toc32570)

[2.18.2. 相关配置 30](#_Toc10702)

[2.18.3. 服务启动和验证 31](#_Toc2868)

[2.19. 系统监测服务 32](#_Toc28046)

[2.19.1. 安装系统监测服务 32](#_Toc29968)

[2.19.2. 相关配置 32](#_Toc20744)

[2.19.3. 服务启动和验证 32](#_Toc26746)

[2.20. 值班管理服务 33](#_Toc8788)

[2.20.1. 安装值班管理服务 33](#_Toc24742)

[2.20.2. 相关配置 33](#_Toc23747)

[2.20.3. 服务启动和验证 33](#_Toc4896)

[2.21. 电子地图服务 34](#_Toc16406)

[2.21.1. 安装电子地图服务 34](#_Toc6269)

[2.21.2. 服务启动和验证 34](#_Toc17183)

# 平台服务对象程序介绍和环境部署

## 平台各服务对象程序介绍

|  |  |  |
| --- | --- | --- |
| 系统名称 | 安装程序 | 服务简介 |
| 自定义平台 | 自定义应用服务 | 提供数据结构、动态页面和动态流程配置和管理功能 |
| 流程引擎服务 | 提供动态流程的运行支持、管理功能 |
| 数据库服务 | 存储自定义平台数据的载体 |
| 数据接口服务 | 作为自定义平台与其他平台及应用的接口网关，负责自定义平台与其他平台及应用的消息转发、控制指令路由、消息格式转换功能 |
| Zookeeper服务 | 分发服务与服务之间接口连接 |
| 搜索引擎服务 | 聚合了平台中搜索和统计需要使用的结构化数据，并进行分类和关联，以对自定义平台提供搜索、计算、统计功能支撑 |
| 数据变更服务 | 1.数据变更服务（DataWatcher），用于获取数据库变化并把变化传给平台其它服务。 2.数据变更服务除主体DataWatcher外，还依赖其它3个工具：ActiveMQ、Redis、Canal服务 3.主体 DataWatcher，利用Canal获取数据库变化，根据各服务的配置，对变化进行过滤，并把过滤后的变化按规定格式包装发给服务。 |
| ActiveMQ | 消息队列工具 Activemq，DataWatcher和其它的服务的通信通过Activemq作为中间站进行，用于缓存DataWatcher向其它服务发送出来的消息包，避免异常、断电等情况导致消息包丢失。 |
| Redis | 缓存工具 Redis，用于对DataWatcher正在处理的变化数据进行缓存，避免异常、断电等情况导致变化数据丢失。 |
| Canal服务 | 监听工具 Canal，用于监听数据库的变化。 |
| 定时服务 | 提供根据配置的时间规则，去触发1个定时任务。当任务执行时去搜索引擎去取输入数据，并加工推送到输出服务，输出服务返回执行结果 |

## 平台各服务组网图



## 平台安装服务器要求

本次安装文档基于服务器环境：

|  |  |
| --- | --- |
| 操作系统 | CentOS7.5 |
| cpu | E5 |
| 内存 | 32G |
| 硬盘 | 1T |
| 网络 | 千兆以太网 |

本次安装全部单机方式，如果有集群安装需要，或者只需要安装部分服务，需另外估量。

# 服务安装配置

## 服务器环境初始化

注意：使用root用户执行

### 关闭selinux和防火墙

关闭防火墙：systemctl stop firewalld ; systemctl disable firewalld

关闭selinux：setenforce 0 ; sed -i '/SELINUX/s/enforcing/disabled/' /etc/selinux/config

### 创建自定义平台安装用户

创建用户：useradd -m iotmp

修改用户密码：echo 123456 | passwd --stdin iotmp

给用户授权：sed -i '$a%iotmp ALL=(ALL) NOPASSWD: ALL' /etc/sudoers

### 配置依赖服务yum源

清空外网yum源：rm -f /etc/yum.repos.d/\*

配置内网ftp源：

cat <<EOF >> /etc/yum.repos.d/ftp.repo

[ftp]

name=ftp

baseurl=ftp://192.168.8.88/centos7

enable=1

gpgcheck=0

EOF

以下除非特殊说明，全部使用iotmp用户执行安装

## JDK服务

说明：除了MySQL和Redis服务以外的服务都依赖于JDK

### 安装JDK1.8

创建基础服务安装目录并进入：cd && [ -d base ] || mkdir base ; cd base

下载JDK安装包：wget <ftp://192.168.8.88/base/jdk-8u181-jce-x64.tar.gz>

解压：tar xf jdk-8u181-jce-x64.tar.gz

### 修改配置文件

配置环境变量：sudo vim /etc/profile

在尾部添加以下内容：

* export JAVA\_HOME=/home/iotmp/base/jdk1.8
* export CLASSPATH=/home/iotmp/base/jdk1.8/lib/dt.jar:/home/iotmp/base/jdk1.8/lib/tools.jar
* export PATH=${PATH}:/home/iotmp/base/jdk1.8/bin

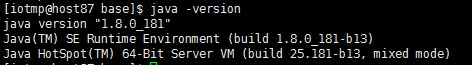
使配置文件生效：source /etc/profile

很重要：创建软链接：sudo ln -s /home/iotmp/base/jdk1.8/bin/java /usr/bin/java

### 启动及验证

启动：jdk不需要主动启动

验证jdk：java -version



## MySQL服务

### 安装MySQL服务

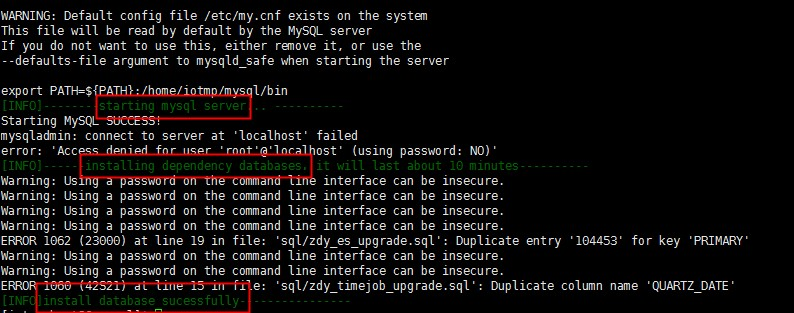
进入用户家目录：cd

下载mysql安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.MySQL.tgz>

解压：tar xf IOTMP.MySQL.tgz && cd mysql

执行一键安装脚本：./install.sh

下面是正常的运行结果：



### 启动及验证

安装过程已经附带启动

启动/关闭/重启：service mysql start/stop/restart

验证：使用navicat连接验证



## Redis服务

### 配置Redis服务系统参数

注意：使用root执行

参数1配置：echo "vm.overcommit\_memory = 1" >> /etc/sysctl.conf && sysctl -p

参数2配置：echo "echo never > /sys/kernel/mm/transparent\_hugepage/enabled" >> /etc/rc.local

安装编译依赖：yum -y install gcc

切换到iotmp用户：su - iotmp

### 安装Redis服务

切换到基础服务安装目录：cd && [ -d base ] || mkdir base ; cd base

下载安装包：wget <ftp://192.168.8.88/base/redis-4.0.11.tar.gz>

解压并进入目录：tar xf redis-4.0.11.tar.gz && mv redis-4.0.11 redis && cd redis

编译安装：make PREFIX=/home/iotmp/base/redis install

### 修改配置文件

添加path路径：sudo sed -i '$aexport PATH=${PATH}:/home/iotmp/base/redis/bin' /etc/profile

使path路径生效：source /etc/profile

修改配置文件：vim redis.conf

* bind 0.0.0.0
* requirepass Hy123456

### 服务启动和验证

注册redis服务：sudo vim /usr/lib/systemd/system/redis.service

注意：将下面绿色部分全部写入文件中，先输入i提示insert后复制粘贴进去

[Unit]

Description=Redis

After=network.target

[Service]

ExecStart=/home/iotmp/base/redis/src/redis-server /home/iotmp/base/redis/redis.conf --daemonize no

ExecStop=/home/iotmp/base/redis/src/redis-cli -h 127.0.0.1 -p 6379 shutdown

[Install]

WantedBy=multi-user.target

启动服务：sudo systemctl start redis

停止服务：sudo systemctl stop redis

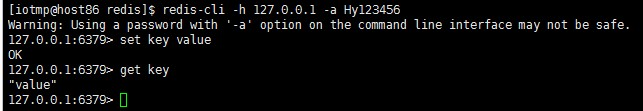
重启服务：sudo systemctl restart redis

创建开机自启动：sudo systemctl enable redis

查看服务状态：sudo systemctl status redis

验证服务启动：redis-cli -h 127.0.0.1 -a Hy123456

下面是正常运行的验证结果：



## Zookeeper服务

服务依赖：JDK1.8

### 安装Zookeeper服务

切换到基础服务安装目录：cd && [ -d base ] || mkdir base ; cd base

下载安装包：wget <ftp://192.168.8.88/base/zookeeper-3.4.12.tar.gz>

解压并进入安装目录：tar xf zookeeper-3.4.12.tar.gz && mv zookeeper-3.4.12 zookeeper && cd zookeeper

### 修改配置文件

拷贝配置文件：cp conf/zoo\_sample.cfg conf/zoo.cfg

修改配置文件：vim conf/zoo.cfg

* dataDir=/home/iotmp/base/zookeeper/data

添加path路径：sudo sed -i '$aexport PATH=${PATH}:/home/iotmp/base/zookeeper/bin' /etc/profile

使path路径生效：source /etc/profile

### 服务启动和验证

注册zookeeper服务：sudo vim /usr/lib/systemd/system/zookeeper.service

[Unit]

Description=ZooKeeper.service

After=network.target

[Service]

Type=forking

ExecStart=/home/iotmp/base/zookeeper/bin/zkServer.sh start

[Install]

WantedBy=multi-user.target

启动服务：sudo systemctl start zookeeper

停止服务：sudo systemctl stop zookeeper

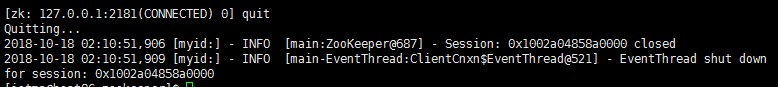
重启服务：sudo systemctl restart zookeeper

开机自启：sudo systemctl enable zookeeper

验证服务启动：zkCli.sh -server 127.0.0.1:2181



quit退出连接



## ActiveMQ服务

依赖服务：JDK1.8

### 安装ActiveMQ

切换到基础服务安装目录：cd && [ -d base ] || mkdir base ; cd base

下载activemq安装包：wget <ftp://192.168.8.88/base/apache-activemq-5.15.6-bin.tar.gz>

解压安装包：tar xf apache-activemq-5.15.6-bin.tar.gz && mv apache-activemq-5.15.6 activemq

### 服务启动和验证

注册activemq服务：sudo vim /usr/lib/systemd/system/activemq.service

[Unit]

Description=ActiveMQ.service

After=network.target

[Service]

Type=forking

ExecStart=/home/iotmp/base/activemq/bin/linux-x86-64/activemq start

[Install]

WantedBy=multi-user.target

启动服务：sudo systemctl start activemq

停止服务：sudo systemctl stop activemq

重启服务：sudo systemctl restart activemq

开机自启：sudo systemctl enable activemq

验证服务启动：ip:8161



## Canal服务

依赖服务：JDK1.8

### 安装Canal服务

切换到基础服务安装目录：cd && [ -d base ] || mkdir base ; cd base

下载canal安装包：wget <ftp://192.168.8.88/base/canal.deploy-1.1.0.tar.gz>

解压并进入安装目录：tar xf canal.deploy-1.1.0.tar.gz

### 相关配置

修改配置文件：vim canal/conf/dataWatcher/instance.properties

* canal.instance.master.address=mysql数据库服务安装地址:3306



### 服务启动和验证

注册canal服务：sudo vim /usr/lib/systemd/system/canal.service

[Unit]

Description=Canal.service

After=network.target

[Service]

Type=forking

ExecStart=/home/iotmp/base/canal/bin/startup.sh

[Install]

WantedBy=multi-user.target

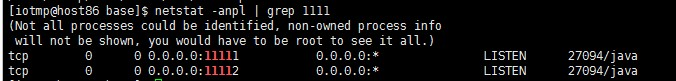
启动服务：sudo systemctl start canal

停止服务：sudo systemctl stop canal

重启服务：sudo systemctl restart canal

开机自启：sudo systemctl enable canal

验证服务启动情况：netstat -anpl | grep 1111



## 数据接口服务

依赖服务：JDK1.8、zookeeper、redis、mysql

### 安装数据接口服务

切换到安装目录：cd

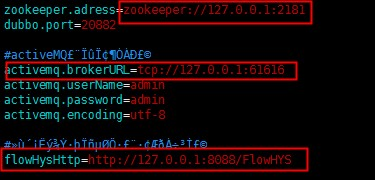
下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.DataBus.tgz>

解压：tar xf IOTMP.DataBus.tgz

### 相关配置

修改配置文件：vim DataBus/webapps/ESB/WEB-INF/classes/config/system.properties

* zookeeper.address=zookeeper服务安装的地址:2181
* activemq.brokerURL=tcp://activemq服务安装的地址:61616
* flowHysHttp=http://基础数据服务安装的地址:8088/FlowHYS



### 服务启动和验证

注册databus服务：sudo vim /usr/lib/systemd/system/databus.service

[Unit]

Description=DataBus.service

After=network.target

[Service]

Type=forking

ExecStart=/home/iotmp/DataBus/bin/startup.sh

[Install]

WantedBy=multi-user.target

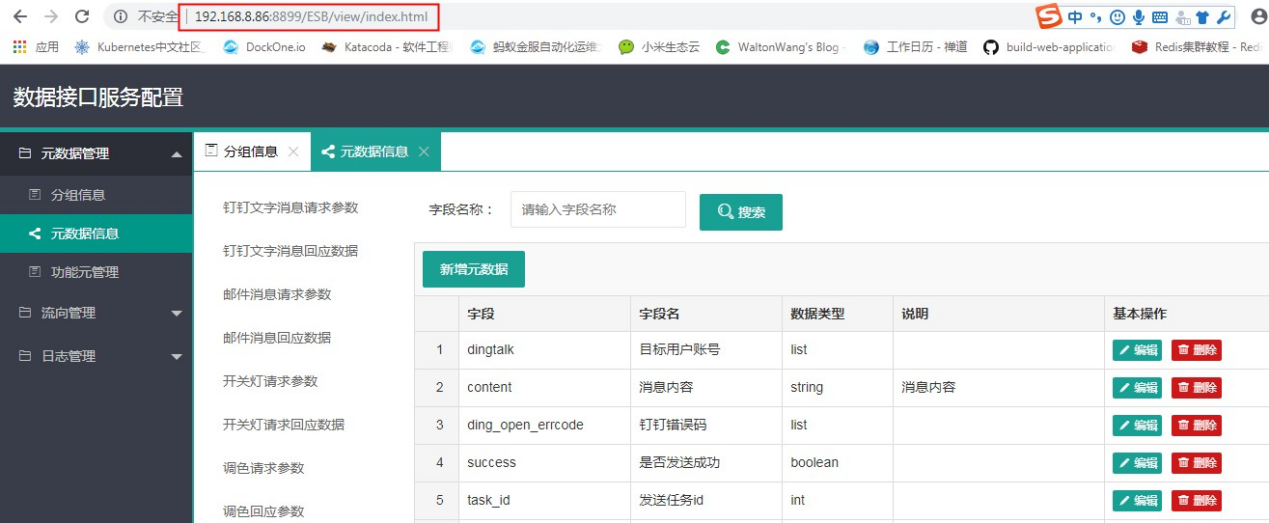
启动服务：sudo systemctl start databus

停止服务：sudo systemctl stop databus

重启服务：sudo systemctl restart databus

开机自启：sudo systemctl enable databus

验证服务启动：ip:8899/ESB/view/index.html



## 系统配置服务

依赖服务：JDK1.8

### 安装系统配置服务

切换到安装目录：cd

下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.SystemSettings.tgz>

解压：tar xf IOTMP.SystemSettings.tgz

### 相关配置

修改配置文件：vim SystemSettings/webapps/platformConfig/WEB-INF/classes/application.properties

* zookeeper.address=zookeeper服务安装地址:2181



### 服务启动和验证

注册systemsettings服务：sudo vim /usr/lib/systemd/system/systemsettings.service

[Unit]

Description=SystemSettings.service

After=network.target

[Service]

Type=forking

ExecStart=/home/iotmp/SystemSettings/bin/startup.sh

[Install]

WantedBy=multi-user.target

启动服务：sudo systemctl start systemsettings

停止服务：sudo systemctl stop systemsettings

重启服务：sudo systemctl restart systemsettings

开机自启：sudo systemctl enable systemsettings

验证服务启动：ip:8905/platformConfig/



## 基础数据服务

依赖服务：JDK1.8、zookeeper服务、mysql服务、redis服务、activemq服务

### 安装基础数据服务

创建日志文件目录：sudo mkdir /iotmp/{formAtt,logs} -p

修改目录所有者：sudo chown -R iomtp:iotmp /iotmp

切换到iotmp用户家目录：cd

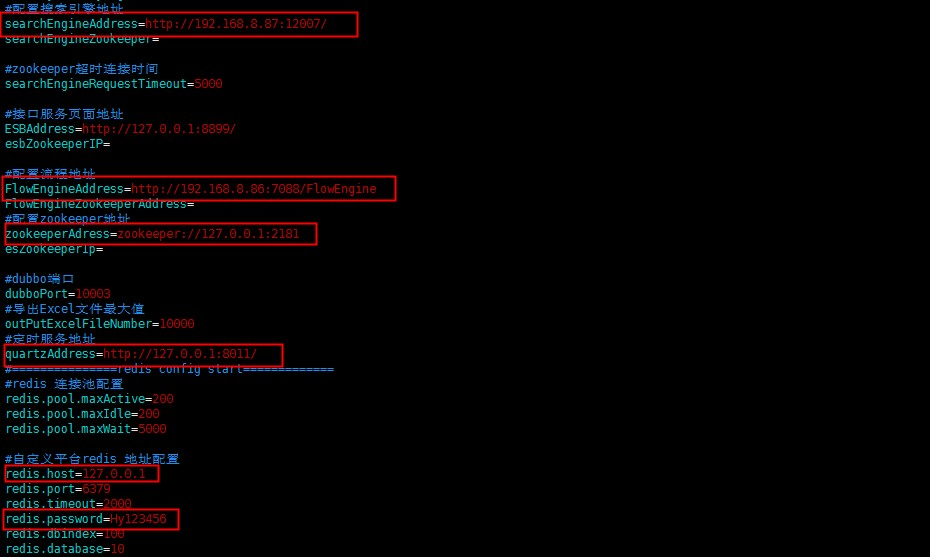
下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.FlowHYS.tgz>

解压：tar xf IOTMP.FlowHYS.tgz

### 相关配置

修改配置文件：vim FlowHYS/webapps/FlowHYS/WEB-INF/classes/config/system.properties

* jdbc\_url=jdbc:mysql://mysql服务安装地址:3306/basisdata?useUnicode=true&characterEncoding=UTF-8
* searchEngineAddress=<http://搜索引擎服务安装地址:12007/>
* FlowEngineAddress=<http://流程引擎服务安装地址:7088/FlowEngine>
* zookeeperAdress=zookeeper://zookeeper服务安装地址:2181
* quartzAddress=<http://定时任务服务安装地址:8011/>
* redis.host=redis服务安装地址
* redis.password=redis服务安装的时候配置的密码



### 服务启动和验证

注册flowhys服务：sudo vim /usr/lib/systemd/system/flowhys.service

[Unit]

Description=FlowHYS.service

After=network.target

[Service]

Type=forking

ExecStart=/home/iotmp/FlowHYS/bin/startup.sh

[Install]

WantedBy=multi-user.target

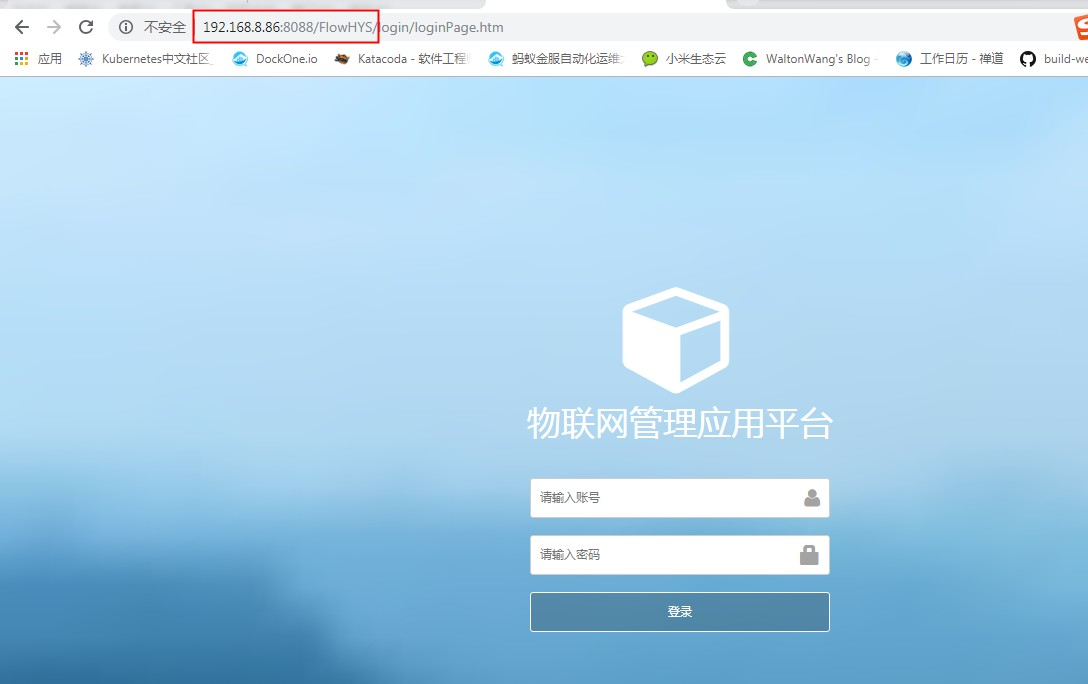
启动服务：sudo systemctl start flowhys

停止服务：sudo systemctl stop flowhys

重启服务：sudo systemctl restart flowhys

开机自启：sudo systemctl enable flowhys

验证服务启动：ip:8088/FlowHYS



## 流程引擎服务

服务依赖：JDK1.8、zookeeper服务、mysql服务、redis服务

### 安装流程引擎服务

切换到安装目录：cd

下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.FlowEngine.tgz>

解压：tar xf IOTMP.FlowEngine.tgz

### 相关配置

修改配置文件：vim FlowEngine/webapps/FlowEngine/WEB-INF/classes/config/system.properties

* jdbc\_url=jdbc:mysql://mysql服务安装地址:3306/basisdata?useUnicode=true&characterEncoding=UTF-8
* ESBAddress=http://数据接口服务安装地址:8899/
* searchEngineAddress=<http://搜索引擎服务安装地址:12007/>
* zookeeperAdress=zookeeper://zookeeper服务安装地址:2181
* quartzAddress=<http://定时任务服务安装地址:8011/>



### 服务启动和验证

注册flowengine服务：sudo vim /usr/lib/systemd/system/flowengine.service

[Unit]

Description=FlowEngine.service

After=network.target

[Service]

Type=forking

ExecStart=/home/iotmp/FlowEngine/bin/startup.sh

[Install]

WantedBy=multi-user.target

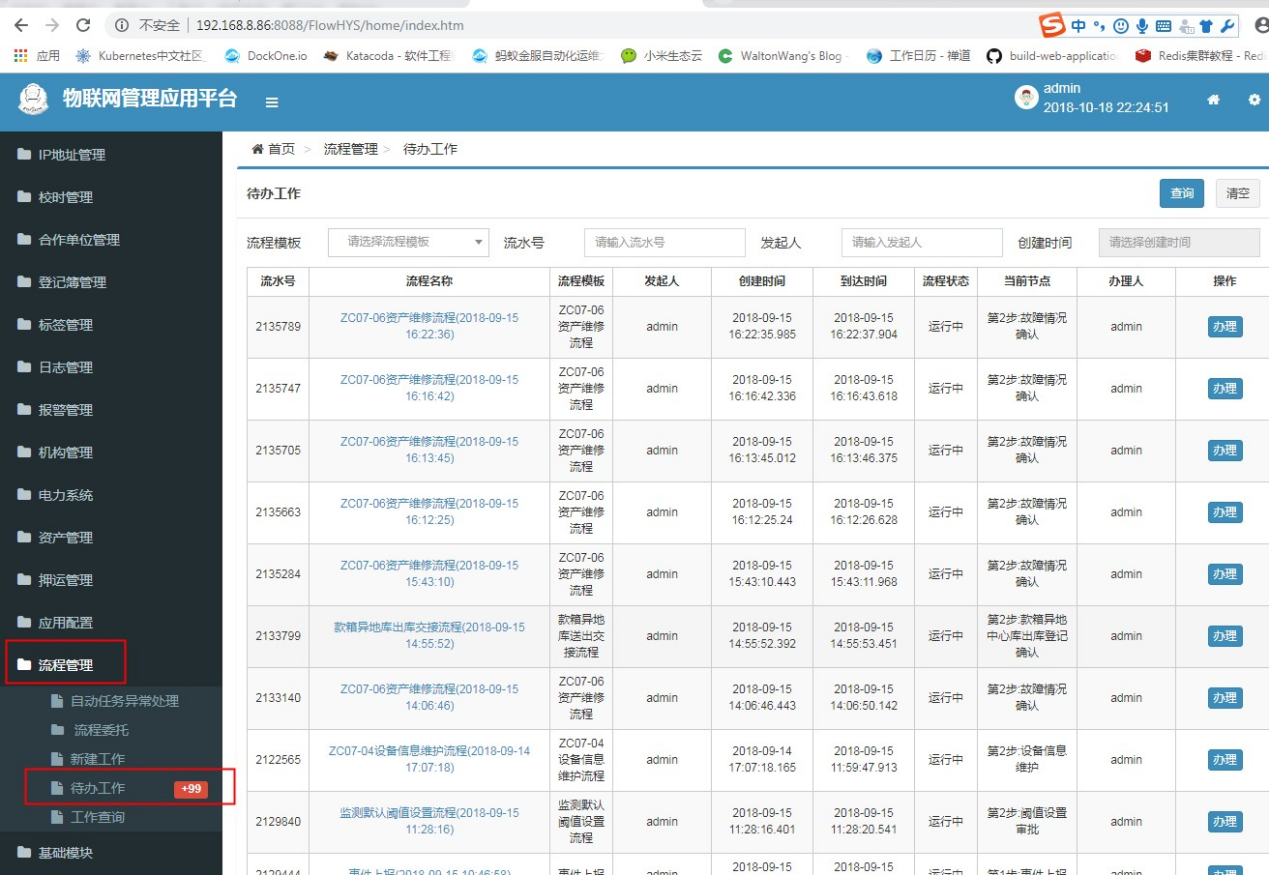
启动服务：sudo systemctl start flowengine

停止服务：sudo systemctl stop flowengine

重启服务：sudo systemctl restart flowengine

开机自启：sudo systemctl enable flowengine

验证服务启动：登录物联网管理平台-->流程管理-->待办工作



## 数据库监测服务

服务依赖：JDK1.8、zookeeper服务、mysql服务、redis服务、canal服务、activemq服务

### 安装数据库监测服务

切换到安装目录：cd

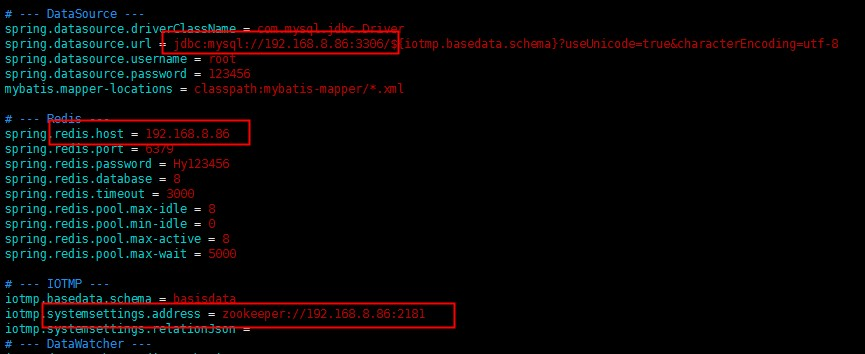
下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.DataWatcher.tgz>

解压：tar xf IOTMP.DataWatcher.tgz && cd DataWatcher/

### 相关配置

修改配置文件：vim config/application.properties

* spring.datasource.url：mysql数据库服务安装地址
* spring.redis.host：redis服务安装地址
* iotmp.systemsettings.address：zookeeper服务安装地址



### 服务启动和验证

注册datawatcher服务：sudo vim /usr/lib/systemd/system/datawatcher.service

[Unit]

Description=DataWatcher.service

After=network.target

[Service]

Type=forking

ExecStart=/home/iotmp/DataWatcher/startup.sh

[Install]

WantedBy=multi-user.target

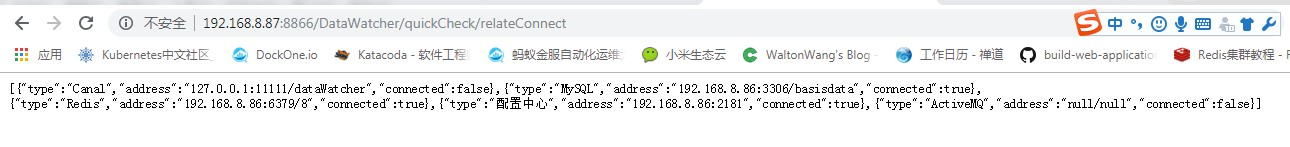
启动服务：sudo systemctl start datawatcher

停止服务：sudo systemctl stop datawatcher

重启服务：sudo systemctl restart datawatcher

开机自启：sudo systemctl enable datawatcher

验证服务启动：ip:8866/DataWatcher/quickCheck/relateConnect



## 数据级联服务

服务依赖：JDK1.8、zookeeper服务、mysql服务、redis服务

### 安装数据级联服务

切换到安装目录：cd

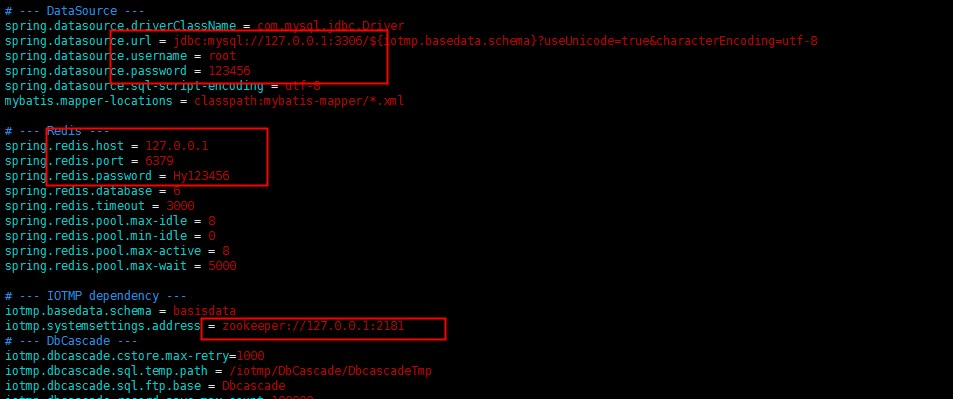
下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.DbCascade.tgz>

解压：tar xf IOTMP.DbCascade.tgz

### 相关配置

修改配置文件：vim DbCascade/webapps/DbCascade/WEB-INF/classes/config/application.properties

* spring.datasource .url：mysql数据库服务安装地址
* spring.redis.host：redis服务安装地址
* spring.redis.password：redis服务访问密码
* iotmp.systemsettings.address：zookeeper服务安装地址



### 服务启动和验证

注册dbcascade服务：sudo vim /usr/lib/systemd/system/dbcascade.service

[Unit]

Description=DbCascade.service

After=network.target

[Service]

Type=forking

ExecStart=/home/iotmp/DbCascade/bin/startup.sh

[Install]

WantedBy=multi-user.target

启动服务：sudo systemctl start dbcascade

停止服务：sudo systemctl stop dbcascade

重启服务：sudo systemctl restart dbcascade

开机自启：sudo systemctl enable dbcascade

验证服务启动：ip:6688/DbCascade/quickCheck/relateConnect



## 搜索引擎工具

### 修改服务器参数

注意：使用root用户

* 修改进程最大内存区域：echo 'vm.max\_map\_count=262144' >> /etc/sysctl.conf && sysctl -p
* 修改文件最大句柄数：e

echo '\* soft nofile 65536 ' >> /etc/security/limits.conf

echo '\* hard nofile 65536' >> /etc/security/limits.conf

ulimit -n 65536

### 安装搜索引擎工具

注意：切换回iotmp用户：su - iotmp

切换到安装目录：cd

下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.ElasticSearch.tgz>

解压：tar xf IOTMP.ElasticSearch.tgz && cd elasticsearch-6.0.0

### 相关配置

jvm虚拟机参数：vim config/jvm./home/hyesion

* -Xms1g
* -Xmx1g

修改绿色部分1g内存为当前服务器内存的1/8

### 服务启动和验证

注册elasticsearch服务：sudo vim /usr/lib/systemd/system/elasticsearch.service

注意：将下面绿色部分全部写入文件中，先输入i提示insert后复制粘贴进去

[Unit]

Description=Elasticsearch.service

After=network-online.target

[Service]

Environment=ES\_HOME=/home/iotmp/elasticsearch-6.0.0

Environment=ES\_PATH\_CONF=/home/iotmp/elasticsearch-6.0.0/config

Environment=PID\_DIR=/var/run/elasticsearch

EnvironmentFile=-/home/iotmp/elasticsearch-6.0.0/config/elasticsearch.yml

WorkingDirectory=/home/iotmp/elasticsearch-6.0.0

User=iotmp

Group=iotmp

ExecStart=/home/iotmp/elasticsearch-6.0.0/bin/elasticsearch

StandardOutput=journal

StandardError=inherit

LimitNOFILE=65536

LimitNPROC=4096

LimitAS=infinity

LimitFSIZE=infinity

TimeoutStopSec=0

KillSignal=SIGTERM

KillMode=process

SendSIGKILL=no

SuccessExitStatus=143

[Install]

WantedBy=multi-user.target

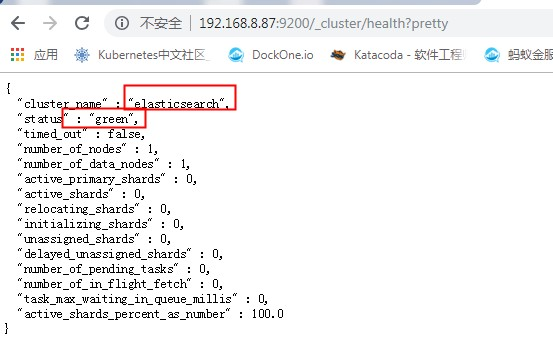
启动服务：sudo systemctl start elasticsearch

停止服务：sudo systemctl stop elasticsearch

重启服务：sudo systemctl restart elasticsearch

开机自启：sudo systemctl enable elasticsearch

验证服务启动：ip:9200/\_cluster/health?pretty



## 搜索与计算引擎服务

服务依赖：JDK1.8、zookeeper服务、mysql服务、redis服务、activemq服务、数据库变更服务

### 安装搜索与计算引擎服务

切换到安装目录：cd

下载搜索与计算引擎服务安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.HyEs.tgz>

下载同步服务按照包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.Logstash.tgz>

解压：tar xf IOTMP.Logstash.tgz && tar xf IOTMP.HyEs.tgz && cd HyEs

### 相关配置

修改配置文件：vim application.properties

* spring.master.datasource.url：mysql服务安装地址
* spring.custom.datasource.slave.url：mysql服务安装地址
* spring.redis.host：redis服务安装地址
* spring.activemq.broker-url：activemq服务安装地址
* zookeeper.adress：zookeeper服务安装地址

修改dubbo服务配置：vim dubbo/zookeepDubbo.xml

* 2181端口对应的ip为zookeeper服务所在服务器ip

修改es服务对接参数：vim ElasticsearchProperties.properties

* 修改3306端口前面地址为MySQL服务所在服务器ip

### 服务启动和验证

注册HyEs服务：sudo vim /usr/lib/systemd/system/hyes.service

注意：将下面绿色部分全部写入文件中，先输入i提示insert后复制粘贴进去

[Unit]

Description=HyEs.service

After=network.target elasticsearch.service

[Service]

Type=simple

ExecStart=/home/iotmp/base/jdk1.8/bin/java -jar /home/iotmp/HyEs/es.jar

[Install]

WantedBy=multi-user.target

启动服务：sudo systemctl start hyes

停止服务：sudo systemctl stop hyes

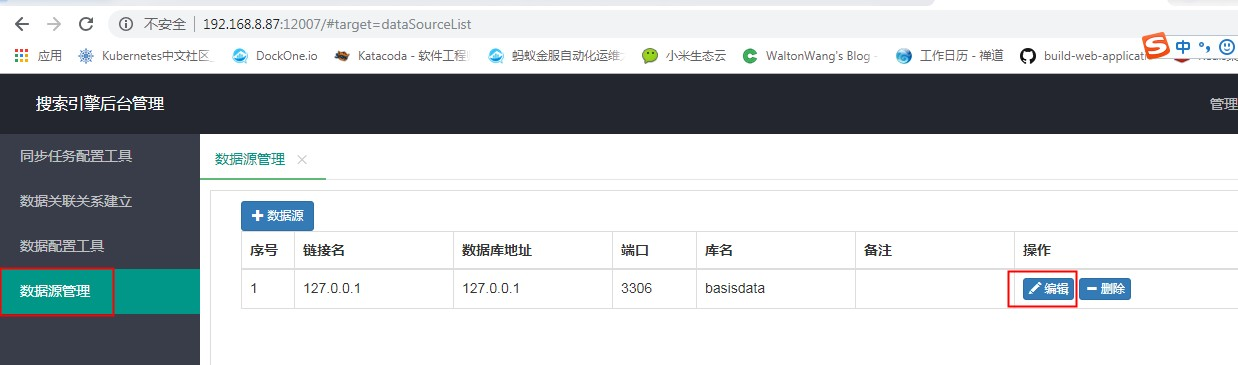
重启服务：sudo systemctl restart hyes

开机自启：sudo systemctl enable hyes

验证服务启动：ip:12007

数据源管理：

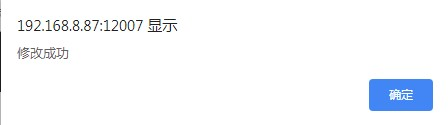
1. 点击数据源管理，编辑



1. 填写正确的mysql数据源信息后提交：



1. 提示修改成功



同步任务配置工具：

1. 点击“同步任务配置工具”进行配置文件，点击“编辑”，修改文件路径和文件名：



* 文件路径：/home/iotmp/elasticsearch-6.0.0/logstash-6.0.0/bin/mysql/
* 文件名：jdbc12.conf

1. 输入正确的数据同步文件路径后，点击提交



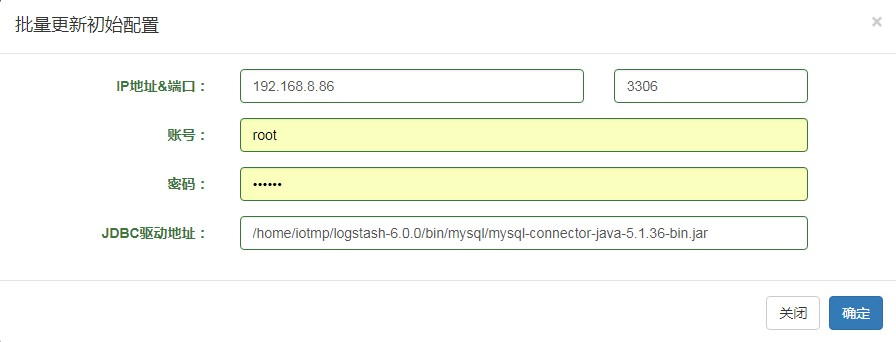
1. 提示修改成功



点击配置文件，进入配置文件界面，在界面点击批量更新初始配置，在批量更新初始配置输入以下信息：









点击“生成文件”





## 同步服务

### 同步服务介绍

同步服务包括：

数据库数据同步服务：对应配置文件为logstash-6.0.0/bin/mysql/jdbc12.conf

日志同步服务：对应配置文件为logstash-6.0.0/bin/mysql/jdbc13.conf，需要配合filebeat服务使用

上下级es日志同步：对应配置文件为logstash-6.0.0/bin/mysql/log\_cascade.conf，需要上下级es环境

jdbc驱动路径：/home/iotmp/logstash-6.0.0/bin/mysql/mysql-connector-java-5.1.36-bin.jar

### 安装同步服务

切换到安装目录：cd

下载安装包：上面搜索与计算引擎中已经下载并解压了同步服务的安装包

### 相关配置

数据库数据同步服务：点击生成文件实际就是对数据同步服务的配置文件的修改，也可以直接修改对应配置文件内容：vim logstash-6.0.0/bin/mysql/jdbc12.conf

* jdbc\_connection\_string => "jdbc:mysql://192.168.8.86:3306/basisdata"
* jdbc\_user => "root"
* jdbc\_password => "123456"
* jdbc\_driver\_library => "/home/iotmp/logstash-6.0.0/bin/mysql/mysql-connector-java-5.1.36-bin.jar"
* hosts => ["localhost:9200"]

日志同步服务：vim logstash-6.0.0/bin/mysql/jdbc13.conf

* hosts => ["localhost:9200"]

上下级es日志同步：vim logstash-6.0.0/bin/mysql/log\_cascade.conf

* input：hosts => ["localhost:9200"] 下级elasticsearch服务ip地址
* ouput：hosts => ["localhost:9200"] 上级elasticsearch服务ip地址

### 服务启动和验证

注册logstash服务：sudo vim /usr/lib/systemd/system/logstash.service

注意：将下面绿色部分全部写入文件中，先输入i提示insert后复制粘贴进去

其中红色部分jdbc13.conf是同步日志的配置文件，同步数据服务只需要修改配置文件名称即可

[Unit]

Description=Logstash.service

After=network.target elasticsearch.service

Requires=elasticsearch.service

[Service]

Type=simple

ExecStart=/home/iotmp/logstash-6.0.0/bin/logstash -f /home/iotmp/logstash-6.0.0/bin/mysql/jdbc13.conf

[Install]

WantedBy=multi-user.target

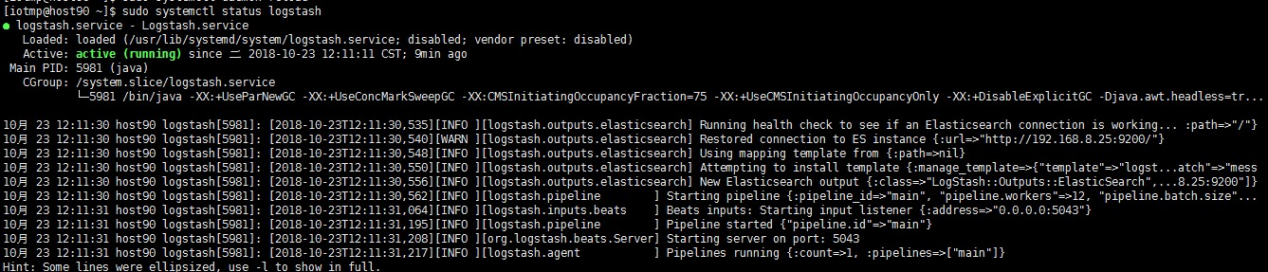
启动服务：sudo systemctl start logstash

停止服务：sudo systemctl stop logstash

重启服务：sudo systemctl restart logstash

开机自启：sudo systemctl enable logstash

验证服务启动：sudo systemctl status logstash



## FileBeat服务

### 安装FileBeat服务

切换到安装目录：cd

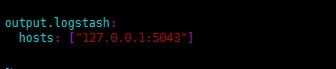
下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.FileBeat.tgz>

解压：tar xf IOTMP.FileBeat.tgz && cd filebeat/

### 相关配置

修改配置文件：vim filebeat.yml

* 修改地址为：日志同步logstash服务的地址 (logstash -f mysql/jdbc13.conf)



### 服务启动和验证

注册filebeat服务：sudo vim /usr/lib/systemd/system/filebeat.service

[Unit]

Description=filebeat.service

After=network.target logstash.service

Requires=logstash.service

[Service]

Type=simple

ExecStart=/home/iotmp/filebeat/filebeat

[Install]

WantedBy=multi-user.target

启动服务：sudo systemctl start filebeat

停止服务：sudo systemctl stop filebeat

重启服务：sudo systemctl restart filebeat

开机自启：sudo systemctl enable filebeat

## 定时任务服务

服务依赖：JDK1.8、zookeeper服务、mysql服务、redis服务

### 安装定时服务

切换到安装目录：cd

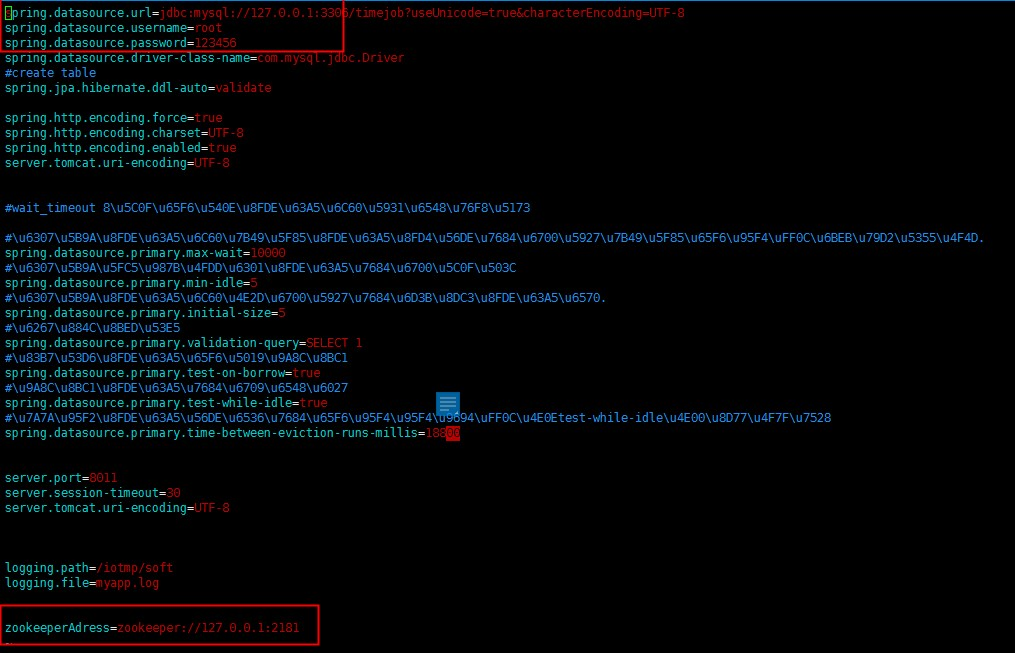
下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.TimeJob.tgz>

解压：tar xf IOTMP.TimeJob.tgz && cd TimeJob

### 相关配置

修改配置文件：vim application.properties

* spring.datasource.url：mysql数据库服务的安装地址
* zookeeperAddress：zookeeper服务的安装地址



### 服务启动和验证

注册timejob服务：sudo vim /usr/lib/systemd/system/timejob.service

[Unit]

Description=TimeJob.service

After=network.target

[Service]

Type=simple

ExecStart=/home/iotmp/base/jdk1.8/bin/java -jar /home/iotmp/TimeJob/timeJob.jar

[Install]

WantedBy=multi-user.target

启动服务：sudo systemctl start timejob

停止服务：sudo systemctl stop timejob

重启服务：sudo systemctl restart timejob

开机自启：sudo systemctl enable timejob

验证服务启动：ip:8011/swagger-ui.html



## 系统监测服务

### 安装系统监测服务

切换到安装目录：cd

下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.InspectManage.tgz>

解压：tar xf IOTMP.InspectManage.tgz && cd InspectManage/

### 相关配置

修改配置文件：vim application.properties

* iotmp.zookeeper.address：zookeeper服务安装地址



### 服务启动和验证

注册inspectmanage服务：sudo vim /usr/lib/systemd/system/inspectmanage.service

注意：将下面绿色部分全部写入文件中，先输入i提示insert后复制粘贴进去

[Unit]

Description=inspectmanage.service

After=network.target

[Service]

Type=simple

ExecStart=/home/iotmp/base/jdk1.8/bin/java -jar /home/iotmp/InspectManage/inspectmanage.jar

[Install]

WantedBy=multi-user.target

启动服务：sudo systemctl start inspectmanage

停止服务：sudo systemctl stop inspectmanage

重启服务：sudo systemctl restart inspectmanage

开机自启：sudo systemctl enable inspectmanage

验证服务启动：ip:9155/execute/searchTask



## 值班管理服务

服务依赖：jdk1.8

### 安装值班管理服务

切换到安装目录：cd

下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.WorkClass.tgz>

解压：tar xf IOTMP.WorkClass.tgz

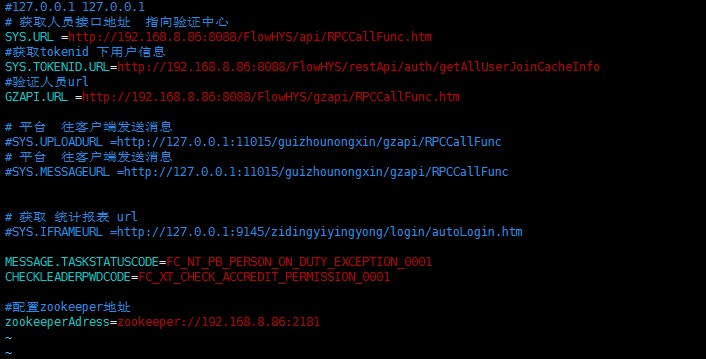
### 相关配置

修改数据库服务连接配置：vim WorkClass/webapps/WorkClass/WEB-INF/classes/config/db.properties

* db.url：mysql数据库服务安装地址

修改基础数据服务连接配置：vim WorkClass/webapps/WorkClass/WEB-INF/classes/config/system.properties

* SYS.URL：基础数据服务安装地址
* SYS.TOKENID.URL：基础数据服务安装地址
* GZAPI.URL：基础数据服务安装地址
* zookeeperAdress：zookeeper服务安装地址



### 服务启动和验证

注册workclass服务：sudo vim /usr/lib/systemd/system/workclass.service

[Unit]

Description=WorkClass.service

After=network.target

[Service]

Type=forking

ExecStart=/home/iotmp/WorkClass/bin/startup.sh

[Instal]

WantedBy=multi-user.target

启动服务：sudo systemctl start workclass

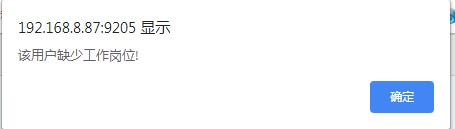
停止服务：sudo systemctl stopp workclass

重启服务：sudo systemctl restart workclass

开机自启：sudo systemctl enable workclass

验证服务启动：

ip:9205/WorkClass/workClass.jsp?tokenId=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1aWQiOiJmYjIwYzY0MjJjMDE0NmE0OGM4OWU5NDZjYTM5NjBmZSIsImlhdCI6MTUzOTA0OTc0Njk5M30.LavpLGuAjHFRdR0AMiHHdfLsH1ZYsJ9D-4pwItmhWCA#



## 电子地图服务

服务依赖：jdk1.8

### 安装电子地图服务

切换到安装目录：cd

下载安装包：wget <ftp://192.168.8.88/IOTMP/2.0.20181019/IOTMP.GisMap.tgz>

解压：tar xf IOTMP.GisMap.tgz

### 服务启动和验证

注册gismap服务：sudo vim /usr/lib/systemd/system/gismap.service

[Unit]

Description=GisMap.service

After=network.target

[Service]

Type=forking

ExecStart=/home/iotmp/GisMap/bin/startup.sh

[Instal]

WantedBy=multi-user.target

服务启动：sudo systemctl start gismap

停止服务：sudo systemctl stop gismap

重启服务：sudo systemctl restart gismap

开机自启：sudo systemctl enable gismap

验证服务启动：ip:8084/cctv/view/app/electronicmap/gismapdemo.html

