

# CVMS 综合物联平台快速部署 操作手册

编写时间：2023 年 5 月

## 历史记录

版本号	描述	撰写人	撰写时间
V1.0	第一版	LionHeart Yang	2022 年 9 月 29 日星期四

1.1. 编写目的 .....	5
1.2. 声明 .....	5
1.3. 运行环境 .....	5
1.4. 定义 .....	5
1.5. 符号约定 .....	6
<b>2. 微服务结构图 .....</b>	<b>7</b>
<b>3. 注意事项 .....</b>	<b>7</b>
3.1. LINUX 基础教程 .....	7
3.2. WINDOWS 下打包.TAR.GZ .....	7
3.3. 文件上传指令 .....	8
3.4. TCPDUMP 使用 .....	8
3.5. IP 地址操作 .....	8
3.6. 端口操作 .....	8
3.7. 进程操作 .....	9
3.8. 目录操作 .....	10
3.9. VI 常规操作 .....	10
3.10. 防火墙操作指令 .....	10
3.11. 路径约定 .....	10
<b>4. 编译环境搭建 .....</b>	<b>10</b>
4.1. 下载地址 .....	10
4.2. 安装步骤 .....	10
<b>5. JDK 的安装 .....</b>	<b>12</b>
5.1. 下载地址 .....	12
5.2. 安装步骤 .....	12
<b>6. ACTIVEMQ 安装 .....</b>	<b>13</b>
6.1. 下载地址 .....	13
6.2. 参考网址 .....	13
6.3. 安装步骤 .....	13
6.4. 常见问题 .....	13
6.5. 设置开机启动 .....	14
<b>7. MYSQL 安装 .....</b>	<b>16</b>
7.1. 下载地址 .....	16
7.2. 创建用户组及用户 .....	16
7.3. 安装步骤 .....	16
7.4. 导入数据库 .....	20
7.5. 常用问题解决参考网址 .....	21
<b>8. REDIS 安装 .....</b>	<b>21</b>
8.1. 下载地址 .....	22
8.2. 安装步骤 .....	22

8.3. 设置开机启动 .....	23
8.4. 验证是否成功 .....	25
<b>9. NGINX 安装及 WEB 页面 .....</b>	<b>25</b>
9.1. 下载地址 .....	25
9.2. 安装步骤 .....	25
9.3. WEB 页面 .....	26
9.4. 设置开机启动 .....	27
9.5. 开启防火墙 .....	28
9.6. 验证是否成功及 WEB 页地址 .....	28
9.7. 使用域名访问平台 .....	28
<b>10. NACOS 安装 .....</b>	<b>31</b>
10.1. 下载地址 .....	31
10.2. 安装步骤 .....	31
10.3. 设置开机启动 .....	31
10.4. 启动指令 .....	33
10.5. 防火墙开放 .....	33
10.6. 验证是否成功 .....	33
10.7. 参数配置 .....	33
<b>11. ELK 安装（可选择是否部署） .....</b>	<b>34</b>
11.1. ELASTICSEARCH 安装 .....	35
11.2. KIBANA 安装 .....	36
11.3. LOGSTASH 安装 .....	36
11.4. FILEBEAT 安装 .....	37
<b>12. JAR 包部署 .....</b>	<b>37</b>
12.1. 下载地址 .....	37
12.2. 路径约定 .....	37
12.3. JAR 通用启动脚本格式 .....	38
12.4. JAR 启动 .....	38
12.5. 数据中心配置 .....	39
12.6. 验证 JAR 微服务是否运行 .....	41
<b>13. 部署 NODE.JS 环境 .....</b>	<b>42</b>
13.1. 下载地址 .....	42
13.2. 安装步骤 .....	42
<b>14. FFMPEG 完整安装 .....</b>	<b>43</b>
14.1. 下载地址 .....	43
14.2. 安装步骤 .....	43
<b>15. 启动转码服务 .....</b>	<b>45</b>
15.1. 下载地址 .....	45
15.2. 安装步骤 .....	45

<b>16. FAQ.....</b>	<b>46</b>
16.1. GA/T 1400 检查参数配置 .....	46
16.1.1. 数据库配置.....	46
16.1.2. 图片保存路径设置.....	46
16.1.3. WebSocket 通讯地址.....	47
16.1.4. ActiveMQ 参数配置.....	47
16.1.5. nacos 参数配置.....	48

# 引言

## 1.1. 编写目的

为项目系统管理人员部署 CVMS 综合物联平台服务端提供支撑

## 1.2. 声明

- a. 软件名称：CVMS 综合物联管理平台
- b. 使用用户：系统管理员
- c. 本手册适用所有版本服务端部署
- d. 本手册可能包含技术上不准确的地方、或与产品功能及操作不相符的地方、或印刷错误。我司将根据产品功能的增强或变化而更新本手册的内容，并将定期改进及更新本手册中描述的软硬件产品。更新的内容将会在本手册的新版本中加入，恕不另行通知
- e. 本手册中内容仅为用户提供参考指导作用，不保证与实物完全一致，请以实物为准。
- f. 部分图片为示意图，请以软件实际界面为准。

## 1.3. 运行环境



	建议最低配置	推荐配置
CPU	E3	E5 及以上
分辨率	1024 * 768	1920 * 1080
内存	16G	16G 及以上（如果含业务系统建议 32GB 及以上）
显卡		建议使用独立显卡
操作系统	ubuntu 20.04	ubuntu 20.04

## 1.4. 定义

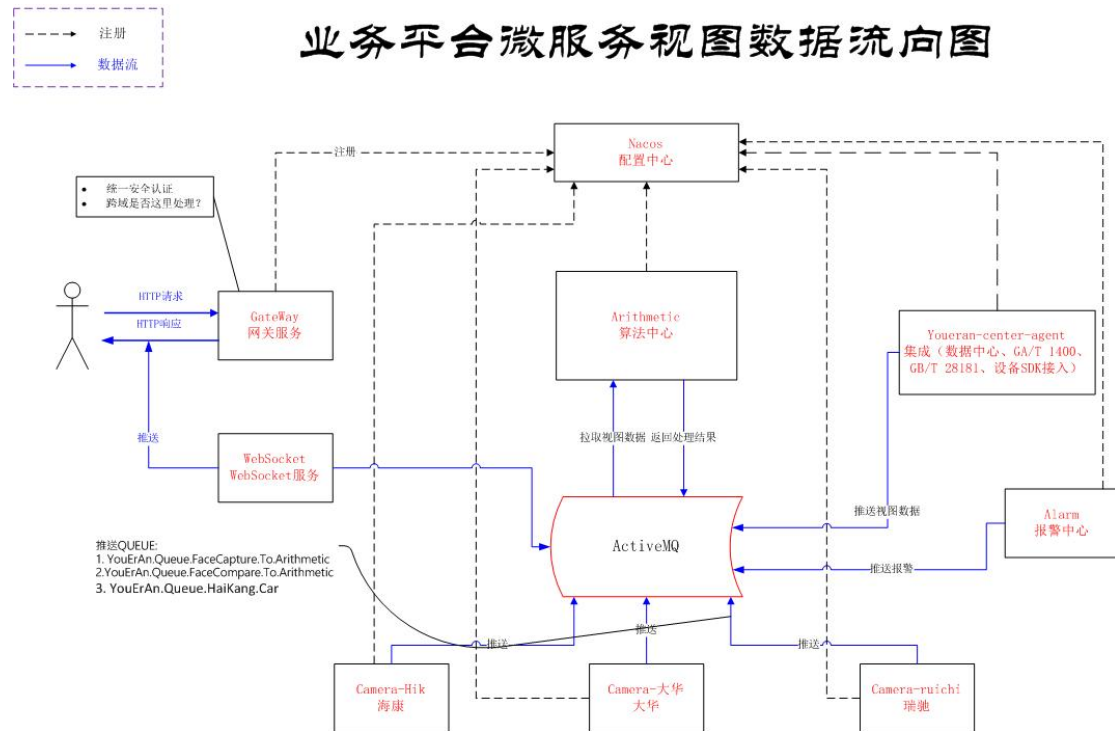
业务模块(红色字体必须安装)	模块执行程序	默认端口	描述
服务注册中心	Nacos	8848	微服务注册中心、全局参数配置中心
消息队列	ActiveMQ	61616、8161 (UI)	用于推送数据和负载均衡
远程字典服务	redis	6379	用于分布式缓存
中心服务	youeran-center-agent.jar	9090	中心平台 (集成数据中心、GA/T 1400、GB/T 28181、设备 SDK 接入)
业务网关微服务	youeran-gateway.jar	8020	提供路由、权限认证

代理微服务	youeran-agent.jar	9580	统一注册服务中心, 监听每个服务是否在线
设备管理微服务	youeran-camera.jar	9501	提供设备管理服务
算法微服务	youeran-arithmetic.jar	9800	提供人、车等算法
报警处理微服务	youeran-alarm.jar	9801	告警统一服务中心
报表处理微服务	youeran-report.jar	9700	1. 摄像设备地图报表 2. 获取所有区域及报表
智慧安防平台模块	youeran-wisdom.jar	9750	客户端接口服务
WebSocket 推送浏览器	youeran-websocket.jar	9299	提供后台与浏览器通过 ws 交互
日志服务	youeran-log		提供分布式日志写入服务
报警主机	youeran-alarm-host.jar	9978	艾礼富报警主机接入
瑞驰运维数据接入微服务	youeran-operation	9900	
华海运维终端接入微服务	youeran-ops-alarm	9802	
东黎周界终端接入微服务	youeran-perimeter-alarm	9910	
护路系统微服务	youeran-hulu.jar	9598	
MQTT	youeran-mqtt.jar	9400	
华力联动告警微服务	youeran-huali.jar	9878、9300	提供华力联动告警服务
公众号/小程序微服务	youeran-wechat.jar	9558	
大华视图/报警接入微服务	youeran-camera-dahua.jar	9550	
海康视图/报警接入微服务	youeran-camera-haikang.jar	9500	
海康平台接入微服务	youeran-hkplatform	9910	主要接入人/车/违停等
瑞驰 EMP 接入微服务	youeran-ruichi-push	9595	瑞驰 EMP 平台接入
党建微服务	youeran-party.jar	9601	
指纹录入串口服务	youeran-fingerprint	9980	提供门禁指纹录入服务
发卡器串口服务	youeran-fakaqi	9979	提供门禁卡片读取服务
门禁微服务	youeran-menjin	9999	提供门禁相关服务
云天励飞报警处理微服务	youeran-ytlf-alarm	9910	接收云天励飞人脸告警

## 1.5. 符号约定


符号	含义	描述
	注意	表示有潜在风险, 如果忽视这些文本, 可能导致错误或不可预知的结果
	说明	表示是正文的附加信息, 是对正文的强调和补充。

## 2. 微服务架构图



## 3. 注意事项

以下软件的安装方式是压缩包的方式安装的, 您可以通过 rpm、yum 命令安装对应的软件, 不一定完全按照这个文档。

	注意	前提对版本兼容性非常熟悉, 否则建议采用离线安装方式。
---	----	-----------------------------

### 3.1. linux 基础教程

<http://c.biancheng.net/view/1068.html>

### 3.2. windows 下打包.tar.gz

下载 7-zip 工具, 先添加到 tar 然后再添加到 gzip

### 3.3. 文件上传指令

Linux rzsz 命令

1、安装 lrzsz 服务

```
yum install -y lrzsz
```

2、非覆盖上传

```
[root@localhost software]# rz -be
```

-b 以二进制方式，默认为文本方式

-e 对所有控制字符转义

3、覆盖上传（解决传输失败的错误）

```
[root@vm-202175161450001 lib]# rz -bey
```

-y 覆盖目的路径的同名文件并上传；

综上所述：建议使用 **【rz -bey】** 命令。

### 3.4. tcpdump 使用

常用语法：tcpdump -i [网卡名] -nn host [主机 ip]

-i : 指定网络接口，对于多个网络接口有用

-n : 显示 IP 地址，不查主机名。当 DNS 不起作用时常用到这个参数

-nn : 不显示协议和端口名。即显示 IP 地址和端口

-w : 将抓包程序保存到文件

### 3.5. IP 地址操作

- 查看 IP

```
ip addr
```

- 修改 IP 地址

```
sudo vi 00-installer-config.yaml
```

- 重启网络服务

```
sudo /etc/init.d/networking restart
```

- 参考网址

<https://blog.csdn.net/inthat/article/details/106738225>

### 3.6. 端口操作

查看端口：-t 显示 TCP -u 显示 UDP -n 不解析服务名称 -l 只显示监听状态端口

-p 显示监听端口的进程(ubuntu 上需要 sudo) -a 显示监听及连接端口

```
ss -tnpl
```



## 3.7. 进程操作

- `ps -ef | grep pronom` #查看进程信息
- `ps -aux | grep pronom` #查看进程信息
- `top -ci` #查看进程 CPU 及内存，相当于 windows 的任务管理器

参数说明：

d : 改变显示的更新速度，或是在交互式指令列( interactive command)按 s  
 q : 没有任何延迟的显示速度，如果使用者是有 superuser 的权限，则 top 将会以最高的优先序执行  
 c : 切换显示模式，共有两种模式，一是只显示执行档的名称，另一种是显示完整的路径与名称  
 S : 累积模式，会将已完成或消失的子进程 ( dead child process ) 的 CPU time 累积起来  
 s : 安全模式，将交互式指令取消，避免潜在的危机  
 i : 不显示任何闲置 (idle) 或无用 (zombie) 的进程  
 n : 更新的次数，完成后将会退出 top  
 b : 批次档模式，搭配 “n” 参数一起使用，可以用来将 top 的结果输出到档案内

top 命令使用过程中，还可以使用一些交互的命令来完成其它参数的功能。这些命令是通过快捷键启动的。

<空格>: 立刻刷新。

P: 根据 CPU 使用大小进行排序。

T: 根据时间、累计时间排序。

e: 切换内存显示单位，默认为 KB

q: 退出 top 命令。

m: 切换显示内存信息。

t: 切换显示进程和 CPU 状态信息。

c: 切换显示命令名称和完整命令行。

M: 根据使用内存大小进行排序。

W: 将当前设置写入 ~/.toprc 文件中。这是写 top 配置文件的推荐方法。

- free 命令用来显示内存的使用情况，使用权限是所有用户。

格式

```
free [-b|-k|-m] [-o] [-s delay] [-t] [-V]
```

主要参数

-b -k -m: 分别以字节 (KB、MB) 为单位显示内存使用情况。

-s delay: 显示每隔多少秒数来显示一次内存使用情况。

-t: 显示内存总和列。

-o: 不显示缓冲区调节列。

应用实例

free 命令是用来查看内存使用情况的主要命令。和 top 命令相比，它的优点是使用简单，并且只占用很少的系统资源。

通过 -s 参数可以使用 free 命令不间断地监视有多少内存存在使用，这样可以把它当作一个方便实时监控器。

```
#free -b -s5
```

### 3.8. 目录操作

```
mkdir #创建目录
rm -r dir #删除目录提示
rm -rf dir #删除目录及子目录，不做任何提示，请谨慎使用
```

### 3.9. vi 常规操作

i: 插入模式

set number: 显示行号

/: 查找模式，输入查找内容，回车，n 向后查找 N 向前查找

u: 撤销上一步的操作

U: 该命令会一次性撤销自上次移动到当前行以来做过的所有操作，再使用一次 U 命令则撤销之前的 U 命令所做的操作，恢复被撤销的内容

o: 插入一行

:0 或者 :1 或 gg 文件首行

:\$或 G 文件尾行

:ngg 或 nG, n 为指定行数

### 3.10. 防火墙操作指令

参考地址: [https://blog.csdn.net/weixin\\_44792344/article/details/109674599](https://blog.csdn.net/weixin_44792344/article/details/109674599)

### 3.11. 路径约定

- 软件拷贝目录: /usr/local/software
- jar 包拷贝目录: /usr/local/jar

## 4. 编译环境搭建

### 4.1. 下载地址

<http://183.56.218.31:18090/downloads/linux/gcclib.tar.gz>

### 4.2. 安装步骤

- 1) cd /usr/local/software
- 2) tar -xzvf gcclib.tar.gz
- 3) 安装 gcc

```
cd /usr/local/software/gcclib/gcc
```

```
rpm -Uvh *.rpm --nodeps --force
```

```
gcc -v
```

```
root@localhost libtool-2.4.2# gcc -v
Using built-in specs.
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
Target: x86_64-redhat-linux
Configured with: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap --enable-shared --enable-threads=posix --enable-checking=release --with-system-libs --enable_cxx_abis --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-linker-hash-style=gnu --enable-languages=C,C++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initfini-array --disable-lto --with-isl=/build/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/isl-install --with-cloog=/build/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Thread model: posix
gcc version 4.8.5 20150623 (Red Hat 4.8.5-16) (GCC)
```

#### 4) 安装 gcc-c++

```
cd /usr/local/software/gcclib/gcc-c++
```

```
rpm -Uvh *.rpm --nodeps --force
```

```
g++ -v
```

```
root@localhost libtool-2.4.2# g++ -v
Using built-in specs.
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
Target: x86_64-redhat-linux
Configured with: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap --enable-shared --enable-threads=posix --enable-checking=release --with-system-libs --enable_cxx_abis --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-linker-hash-style=gnu --enable-languages=C,C++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initfini-array --disable-lto --with-isl=/build/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/isl-install --with-cloog=/build/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Thread model: posix
gcc version 4.8.5 20150623 (Red Hat 4.8.5-16) (GCC)
```

#### 5) 安装 pcre

```
cd /usr/local/software/gcclib/pcre-8.35
```

```
chmod +x configure
```

```
./configure
```

```
make && make install
```

执行./configure 提示: C compiler cannot creat executables, 如下图所示:

```
root@ubuntu:/usr/local/software/gcclib/pcre-8.35# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether make supports nested variables... (cached) yes
checking for style of include used by make... GNU
checking for gcc... gcc
checking whether the C compiler works... no
configure: error: in `/usr/local/software/gcclib/pcre-8.35':
configure: error: C compiler cannot create executables
See `config.log' for more details
```

执行 `sudo apt-get install build-essential` 即可。

#### 6) 安装 libtool

```
cd /usr/local/software/gcclib/libtool-2.4.2
```

```
chmod +x configure  
./configure  
make && make install
```

## 5. JDK 的安装

**\*特别注意（20220929）：**

目前 ubuntu, jar 包运行的环境为 ORACLE JDK1.8

Version: 1.8

### 5.1. 下载地址

<http://183.56.218.31:18090/downloads/linux/jdk-8u172-linux-x64.tar.gz>

### 5.2. 安装步骤

1) cd /usr/local/software

2) 解压 tar -xzvf jdk-8u172-linux-x64.tar.gz

3) 配置 JDK 环境变量配置 JDK 环境变量

➤ vi /etc/profile

➤ 拷贝以下脚本到文件底部

```
#java environment  
  
export JAVA_HOME=/usr/local/software/jdk1.8.0_172  
  
export CLASSPATH=.:$JAVA_HOME/jre/lib/rg.jar:$JAVA_HOME/lib/dt.jar:$JAVAR_HOME/lib/tools.jar  
  
export PATH=$PATH:$JAVA_HOME/bin
```

➤ :wq 保存

4) 让环境变量生效并验证: source /etc/profile

5) java -version 验证是否安装成功

```
[root@localhost mysql]# java -version
java version "1.8.0_172"
Java(TM) SE Runtime Environment (build 1.8.0_172-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.172-b11, mixed mode)
[root@localhost mysql]#
```

## 6. ActiveMQ 安装

Version: 5.15.13

### 6.1. 下载地址

<http://183.56.218.31:18090/downloads/linux/apache-activemq-5.15.13-bin.tar.gz>

### 6.2. 参考网址

<http://www.wjsh.net/lucia557-p-7602155.html>

### 6.3. 安装步骤

- 1) `cd /usr/local/software`
- 2) `sudo tar -xzf apache-activemq-5.15.13-bin.tar.gz`
- 3) `sudo chown -R ${username} apache-activemq-5.15.13`
- 4) `cd apache-activemq-5.15.13`
- 5) 启动命令 `./bin/activemq start`
- 6) 验证: <http://IP:8161>

默认用户名密码: admin/admin

### 6.4. 常见问题

- 解决无法通过 IP 地址访问 ActiveMQ 管理界面的问题

- 1) 修改配置文件中的 host 地址 (`vi conf/jetty.xml`)

```

1 <bean id="jettyPort" class="org.apache.activemq.web.WebConsolePort" init-method="start">
2     <!-- the default port number for the web console -->
3     <property name="host" value="127.0.0.0"/>
4     <property name="port" value="8161"/>
5 </bean>

```

修改 host 对应的 ip 地址。也可以改成 0.0.0.0（创建 docker 镜像的时候改为 0.0.0.0，否则无法从外部浏览器查看消息

队列

## 2) 开放防火墙对应端口

//8161 是管理的端口

```
sudo ufw allow 8161
```

//61616 是通信的端口

```
sudo ufw allow 61616
```

//查看是否设置成功

```
sudo ufw status: 查看已经定义的 ufw 规则
```

## 3) 重启 ActiveMQ 服务

```
./bin/activemq restart
```

# 6.5. 设置开机启动

## 1) 添加到开机启动

## 2) 参考: <https://blog.csdn.net/HuygensCheng/article/details/117995767>

## 3) 创建脚本

```
vi /etc/init.d/activemq
```

## 4) 下载地址: <http://183.56.218.31:18090/downloads/script/activemq>

## 5) 编辑脚本

```
#!/bin/bash
```

```
# chkconfig: 2345 10 90
```

```
# description: activemq ....
```

```
prog=activemq
```

```
JAVA_HOME=/usr/local/software/jdk1.8.0_172
```

```
export JAVA_HOME
```

```
MQ_HOME=/usr/local/software/apache-activemq-5.15.13
```

```
export MQ_HOME
```

```

case "$1" in
start)

    echo "Starting $prog..."

    $MQ_HOME/bin/activemq start

    ;;

stop)

    echo "Stopping $prog..."

    $MQ_HOME/bin/activemq stop

    ;;

restart)

    echo "Stopping $prog..."

    $MQ_HOME/bin/activemq stop

    sleep 2

    echo

    echo "Starting $prog..."

    $MQ_HOME/bin/activemq start

    ;;

*)

    echo "Usage: $prog {start|stop|restart}"

    ;;

esac

exit

```

#### 6) 赋予该脚本可执行权限

```

chmod +x /etc/init.d/activemq

sudo vi /etc/apt/sources.list ## 添加如下源

deb http://archive.ubuntu.com/ubuntu/ trusty main universe restricted multiverse

apt-get update 更新下

apt install sysv-rc-conf

cp /usr/sbin/sysv-rc-conf /usr/sbin/chkconfig 拷贝一个重命名为熟悉的 chkconfig:

sudo chkconfig activemq on

```

## 7. MySQL 安装

Version: mysql5.7.30

### 7.1. 下载地址

[http://183.56.218.31:18090/downloads/linux/mysql-5.7.30-linux-glibc2.12-x86\\_64.tar.gz](http://183.56.218.31:18090/downloads/linux/mysql-5.7.30-linux-glibc2.12-x86_64.tar.gz)

### 7.2. 创建用户组及用户

创建 mysql 用户组: `groupadd mysql`

创建 mysql 用户名: `useradd -g mysql mysql`

设置用户 mysql 密码: `passwd mysql` 回车, 输入新密码



注意

未执行该步骤, 以下执行安装脚本报如下错误

```
[root@MM-0-2-centos mysql]# ./bin/mysql_install_db --user=mysql --basedir=/usr/local/software/mysql/ --datadir=/usr/local/software/mysql/data/
2022-09-20 08:30:54 [WARNING] mysql_install_db is deprecated. Please consider switching to mysqld --initialize
2022-09-20 08:30:54 [ERROR] Failed to verify user id 'mysql'. Does it exist?
```

### 7.3. 安装步骤

- 1) `cd /usr/local/software`
- 2) `tar -xzf mysql-5.7.30-linux-glibc2.12-x86_64.tar.gz`
- 3) 为统一脚本文件这里移动目录到上一级: `mv mysql-5.7.30-linux-glibc2.12-x86_64 ../mysql`
- 4) `cd ..`
- 5) 调整文件夹权限

`chown -R mysql mysql/` #修改 mysql 目录用户

`chgrp -R mysql mysql/` #修改 mysql 目录用户组

`cd mysql`

`mkdir data` #创建 data 目录

`chown -R mysql:mysql data` #修改 data 目录用户及用户组

- 6) 下载地址: <http://183.56.218.31:18090/downloads/script/my.cnf>

- 7) 创建文件: `vi my.cnf`, 拷贝如下内容到文件

```
[mysql]
```

```
socket=/var/lib/mysql/mysql.sock
```



```

# set mysql client default character

default-character-set=utf8


[mysqld]

socket=/var/lib/mysql/mysql.sock

# set mysql server port

port = 3306 #默认是 3306，这里发现 3306 已经被占用，因此防止这种情况发生，可以避免使用 3306 mysql 默认端口

# set mysql install base dir

basedir=/usr/local/mysql          #这里填写实际 mysql 所在路径

# set the data store dir

datadir=/usr/local/mysql/data      #这里填写实际 mysql 所在路径

# set the number of allow max connection

max_connections=200

# set server character default encoding

character-set-server=utf8

# the storage engine

default-storage-engine=INNODB

lower_case_table_names=1

max_allowed_packet=16M

explicit_defaults_for_timestamp=true


[mysql.server]

user=mysql:q

basedir=/usr/local/mysql          #这里填写实际 mysql 所在路径

```

## 8) 执行安装脚本

```

sudo ./bin/mysqld --initialize --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data/ --user=mysql

--lc_messages_dir=/usr/local/mysql/share --lc_messages=en_US

```

```

root@ubuntu:/usr/local/mysql# ./bin/mysqld --initialize --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data/ --lc_messages_dir=/usr/local/mysql/share --lc_messages=en_US
2023-05-08T03:53:40.906206Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
2023-05-08T03:53:41.490870Z 0 [Warning] InnoDB: New log files created, LSN=45790
2023-05-08T03:53:41.632194Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
2023-05-08T03:53:41.668053Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: ec0cbe17-ed53-11ed-a4f8-eada4dc7c041.
2023-05-08T03:53:41.671076Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
2023-05-08T03:53:42.896572Z 0 [Warning] CA certificate ca.pem is self signed.
2023-05-08T03:53:42.977837Z 1 [Note] A temporary password is generated for root@localhost: DsI1=tT(,g;d

```

安装完毕之后，会给出一个临时密码，记住此密码，后面会用的到。

#### 9) 拷贝文件及设置权限

```
sudo chmod 644 my.cnf
```

```
cp ./support-files/mysql.server /etc/init.d/mysqld
```

```
sudo chmod +x /etc/init.d/mysqld
```

#### 10) 创建文件夹

```
sudo mkdir /var/lib/mysql
```

#### 11) 修改/usr/local/mysql/support-files/mysql.server

\*搜索到如图这行，添加红色部分(--user=root)

```

case "$mode" in
'start')
# Start daemon

# Safeguard (relative paths, core dumps..)
cd $basedir

echo $echo_n "Starting MySQL"
if test -x $bindir/mysqld_safe
then
# Give extra arguments to mysqld with the my.cnf file. This script
# may be overwritten at next upgrade.
$bindir/mysqld_safe --user=root --datadir=$datadir --pid-file=$mysqld_pid_file_path $other_args >/dev/null &
wait_for_pid created "$i" "$mysqld_pid_file_path"; return_value=$?

# Make lock for RedHat / SuSE
if test -w "$lockdir"
then
touch "$lock_file_path"
fi

exit $return_value
else
log_failure_msg "Couldn't find MySQL server ($bindir/mysqld_safe)"
fi

```

#### 12) 重启 mysql

/etc/init.d/mysqld restart, 重启成功如下图:

```

root@ubuntu:/usr/local/mysql# /etc/init.d/mysqld restart
Restarting mysqld (via systemctl): mysqld.service.

```

#### 13) 设置开机启动

```
chkconfig mysqld on
```

#### 14) 增加 mysql 到环境变量 path

sudo vi /etc/profile, 添加如下内容到文件尾

```
export PATH=$PATH:/usr/local/mysql/bin
```

如下图:

```
#java environment
export JAVA_HOME=/usr/local/software/jdk1.8.0_172
export CLASSPATH=.:$JAVA_HOME/jre/lib/rg.jar:$JAVA_HOME/lib/dt.jar:$JAVAR_HOME/lib/tools.jar
export PATH=$PATH:$JAVA_HOME/bin
#set mysql environment
export PATH=$PATH:/usr/local/mysql/bin
-- INSERT --
```

执行 `source /etc/profile` 使之生效

#### 15) 修改 mysql 密码

```
sudo dpkg --add-architecture i386
```

```
sudo apt update
```

```
sudo apt-get install libncurses5
```

**不执行以上命令，mysql -uroot -p 报错**

mysql -uroot -p 回车，如果是第一次，输入上面的初始密码即可登录，如下图：

```
[root@VM-0-2-centos mysql]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.30

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

**\*\*\*如果出现如下图，则输入指令 `ln -s /var/lib/mysql/mysql.sock /tmp`**

```
[root@localhost ~]# mysql -uroot -p
Enter password:
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/tmp/mysql.sock' (2)
```

```
set PASSWORD=PASSWORD('新密码');
```

```
flush privileges;
```

```
exit
```

#### 16) 验证 mysql 密码

```
mysql -uroot -p 回车
```

#### 17) 添加远程访问权限

```
use mysql;
```

```
update user set host="%" where user="root";
```

```
select host,user from user;
```

```
quit
```

sudo service mysqld restart (重启生效)

```
[root@M-0-2-centos mysql]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.30 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> update user set host = '%' where user = 'root';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select host, user from user
-> ;
+-----+-----+
| host | user |
+-----+-----+
| % | root |
| localhost | mysql.session |
| localhost | mysql.sys |
+-----+-----+
3 rows in set (0.00 sec)

mysql> quit;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'quit' at line 1
mysql> quit
Bye
[root@M-0-2-centos mysql]#
```

#### 18) 开启防火墙

ufw allow 3306

#### 19) 验证访问

使用 navicat.exe 工具

## 7.4. 导入数据库

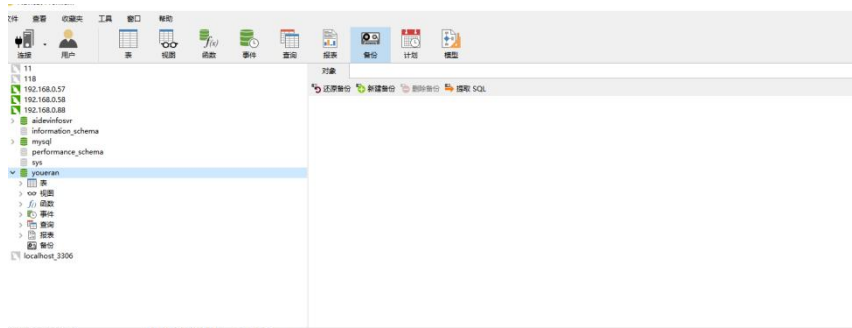
提供两种方式，SQL 脚本&PSC 文件。

下载地址(SQL 脚本): <http://183.56.218.31:18090/downloads/sql.tar.gz>

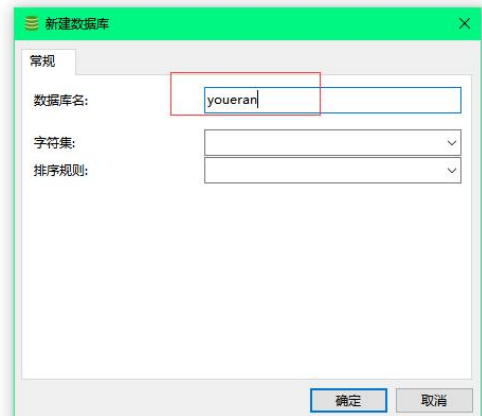
下载地址(导入 psc): [http://183.56.218.31:18090/downloads/sql\\_psc.tar.gz](http://183.56.218.31:18090/downloads/sql_psc.tar.gz)

- 通过 navicat 工具新建两个数据库，aidevinfosvr 和 youeran，按下载文件的名称分别还原数据库到 aidevinfosvr 和 youeran，数据库脚本请联系厂家技术支持。

选择对应的库，备份->还原备份



\*\*\*新建数据库的时候只需输入数据库名称即可，下方字符集及排序规则，不需要选择，如下图所示：



- sql 脚本导入:

```
mysql -uroot -p 回车
set sql_log_bin=off;
set autocommit=0;
create database youeran;
create database aidevinfosvr;
use mysql;
use youeran;
start transaction;
source /usr/local/software/youeran.sql;
commit;
use aidevinfosvr;
start transaction;
source /usr/local/software/aidevinfosvr.sql;
commit;
set sql_log_bin=on;
set autocommit=1;
exit;
```

## 7.5. 常用问题解决参考网址

- [https://blog.csdn.net/weixin\\_42272246/article/details/125534149](https://blog.csdn.net/weixin_42272246/article/details/125534149)
- <https://blog.csdn.net/zqin0/article/details/106444580/>
- <https://www.jb51.net/article/157785.htm>

## 8. Redis 安装

Version: redis-5.0.9

## 8.1. 下载地址

<http://183.56.218.31:18090/downloads/linux/redis-5.0.9.tar.gz>

## 8.2. 安装步骤

- 1) `cd /usr/local/software`
- 2) `sudo tar -xzvf redis-5.0.9.tar.gz`
- 3) `sudo mv redis-5.0.9 ../redis`
- 4) 编译安装:

```
cd ../redis
```

```
sudo make && make install
```

成功如下图:

```
Hint: It's a good idea to run 'make test' ;)

make[1]: Leaving directory '/usr/local/redis/src'
cd src && make install
make[1]: Entering directory '/usr/local/redis/src'
  CC Makefile.dep

Hint: It's a good idea to run 'make test' ;)

  INSTALL install
  INSTALL install
  INSTALL install
  INSTALL install
  INSTALL install
make[1]: Leaving directory '/usr/local/redis/src'
```

- 5) 修改配置文件 `sudo vi redis.conf`

`#bind 127.0.0.1` # 将这行代码注释, 修改为 `0.0.0.0` 监听所有的 ip 地址, 外网可以访问

```
# Examples:
#
# bind 192.168.1.100 10.0.0.1
# bind 127.0.0.1 ::1
#
# ~~~ WARNING ~~~ If the computer running Redis is directly exposed to the
# internet, binding to all the interfaces is dangerous and will expose the
# instance to everybody on the internet. So by default we uncomment the
# following bind directive, that will force Redis to listen only into
# the IPv4 loopback interface address (this means Redis will be able to
# accept connections only from clients running into the same computer it
# is running).
#
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
# ~~~~~
# bind 127.0.0.1
```

`protected-mode no` # 把 `yes` 改成 `no`, 允许外网访问

daemonize yes # 把 no 改成 yes, 后台运行

#requirepass \${newPassword} 将这行代码取消注释, \${newPassword} 改成您的密码

#### 6) 后台启动: ./src/redis-server redis.conf

```
[root@VM-0-2-centos redis]# ./src/redis-server redis.conf
18022:C 20 Sep 2022 14:53:50.863 # 000000000000 Redis is starting 000000000000
18022:C 20 Sep 2022 14:53:50.863 # Redis version=5.0.9, bits=64, commit=00000000, modified=0, pid=18022, just started
18022:C 20 Sep 2022 14:53:50.863 # Configuration loaded
[root@VM-0-2-centos redis]#
```

查看 redis 启动: ps -ef | grep redis

```
[root@VM-0-2-centos redis]# ps -ef | grep redis
root      18023      1  0 14:53 ?        00:00:00 ./src/redis-server 127.0.0.1:6379
root      18387  9669  0 14:56 pts/0    00:00:00 grep --color=auto redis
[root@VM-0-2-centos redis]#
```

## 8.3. 设置开机启动

- 1) 下载地址: <http://183.56.218.31:18090/downloads/script/redis>
- 2) 新建脚本 vi /etc/init.d/redis, 内容如下:

```
#!/bin/sh

# chkconfig:   2345 90 10

# description: Redis is a persistent key-value database

PATH=/usr/local/bin:/sbin:/usr/bin:/bin

REDISPORT=6379

EXEC=/usr/local/bin/redis-server

REDIS_CLI=/usr/local/bin/redis-cli

PIDFILE=/var/run/redis.pid

CONF="/usr/local/redis/redis.conf"    # redis.conf 所在路径

case "$1" in

    start)

        if [ -f $PIDFILE ]

        then

            echo "$PIDFILE exists, process is already running or crashed"

        else

            echo "Starting Redis server..."
```

```

        $EXEC $CONF

    fi

    if [ "$?"="0" ]

    then

        echo "Redis is running..."

    fi

    ;;

stop)

    if [ ! -f $PIDFILE ]

    then

        echo "$PIDFILE does not exist, process is not running"

    else

        PID=$(cat $PIDFILE)

        echo "Stopping ..."

        $REDIS_CLI -p $REDISPORT SHUTDOWN

        while [ -x ${PIDFILE} ]

        do

            echo "Waiting for Redis to shutdown ..."

            sleep 1

        done

        echo "Redis stopped"

    fi

    ;;

restart|force-reload)

    ${0} stop

    ${0} start

    ;;

*)

    echo "Usage: /etc/init.d/redis {start|stop|restart|force-reload}" >&2

    exit 1

```



```
esac
```

3) `cd /etc/init.d`

4) 修改 redis 文件权限

```
chmod 755 redis
```

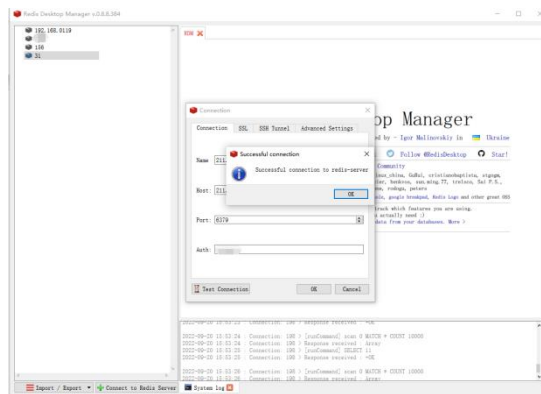
5) 增加到启动配置

```
chkconfig redis on
```

放行端口

```
ufw allow 6379
```

## 8.4. 验证是否成功



安装 redis-desktop-manager-0.8.8.384.exe 验证

下载地址: <http://183.56.218.31:18090/downloads/windows/redis-desktop-manager-0.8.8.384.exe>

## 9. nginx 安装及 WEB 页面

Version: 1.8.0

### 9.1. 下载地址

<http://183.56.218.31:18090/downloads/linux/nginx-1.18.0.tar.gz>

### 9.2. 安装步骤

1) `cd /usr/local/software`

2) `sudo tar -xzf nginx-1.18.0.tar.gz`

3) `cd nginx-1.18.0`

- 4) `sudo ./configure`
- 5) `sudo make && sudo make install`
- 6) 成功安装完成后自动生成 nginx 目录: `/usr/local/nginx`
- 7) `cd /usr/local/nginx`
- 8) `sudo mkdir webdata`      #拷贝对应的前端网页代码到该目录下面,
- 9) `cd webdata`

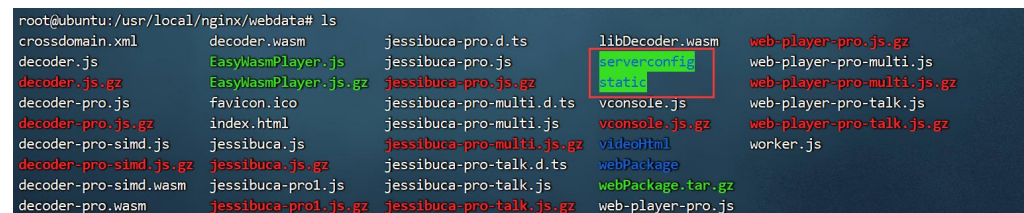
### 9.3. WEB 页面

下载地址: <http://183.56.218.31:18090/downloads/webPackage.tar.gz>

解压至 webdata 目录下 `sudo tar -xzvf webPackage.tar.gz`

解压之后, 进入 webPackage 使用 `mv` 命令, 将所有文件移动到上级目录下: `mv * ../`

webdata 下包括两部分, 前端页面及服务配置。如图:

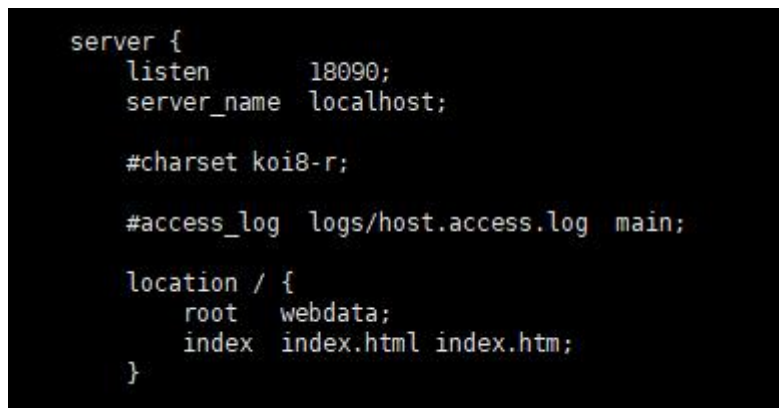


```

root@ubuntu:/usr/local/nginx/webdata# ls
crossdomain.xml  decoder.wasm      jessibuca-pro.d.ts  libDecoder.wasm  web-player-pro.js.gz
decoder.js       EasyWasmPlayer.js jessibuca-pro.js    serverconfig      web-player-pro-multi.js
decoder.js.gz    EasyWasmPlayer.js.gz jessibuca-pro.js.gz static            web-player-pro-multi.js.gz
decoder-pro.js   favicon.ico       jessibuca-pro-multi.d.ts vconsole.js      web-player-pro-talk.js
decoder-pro.js.gz index.html        jessibuca-pro-multi.js vconsole.js.gz   web-player-pro-talk.js.gz
decoder-pro-simd.js jessibuca.js     jessibuca-pro-multi.js.gz videoHtml        worker.js
decoder-pro-simd.js.gz jessibuca.js.gz  jessibuca-pro-talk.d.ts webPackage        webPackage
decoder-pro-simd.wasm jessibuca-pro1.js jessibuca-pro-talk.js webPackage.tar.gz web-player-pro.js
decoder-pro.wasm  jessibuca-pro1.js.gz jessibuca-pro-talk.js.gz web-player-pro.js
  
```

- 10) 修改 `sudo vi ../conf/nginx.conf`, 如下图:

将第一行 `user nobody` 修改为 `user root` 并取消注释;



```

server {
    listen      18090;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root    webdata;
        index   index.html index.htm;
    }
}
  
```

- 11) 修改 `sudo vi ../conf/mime.types`, 如下图:

```

application/x-shockwave-flash      swf;
application/x-stuffit               sit;
application/x-tcl                   tcl tk;
application/x-x509-ca-cert          der pem crt;
application/x-xpinstall             xpi;
application/xhtml+xml              xhtml;
application/xspf+xml                xspf;
application/zip                     zip;
application/wasm                    wasm;

```

## 9.4. 设置开机启动

1、创建开机自启脚本（nginx.service）

```
cd /etc/systemd/system
```

```
vi nginx.service
```

2、nginx.service 文件内容

[Unit]

Description=nginx service

After=network.target

[Service]

Type=forking

ExecStart=/usr/local/nginx/sbin/nginx

ExecReload=/usr/local/nginx/sbin/nginx -s reload

ExecStop=/usr/local/nginx/sbin/nginx -s quit

PrivateTmp=true

[Install]

WantedBy=multi-user.target

3、设置文件权限

```
chmod 755 nginx.service
```

4、设置开机自启动

```
systemctl daemon-reload
```

```
systemctl enable nginx
```

5、先用此命令启停一下 看配置是否正常

```
systemctl start nginx # nginx 启动  
systemctl stop nginx # nginx 服务启动  
./sbin/nginx -c /usr/local/nginx/conf/nginx.conf #启动  
./sbin/nginx -s stop #停止  
./sbin/nginx -s reload #重启
```

## 9.5. 开启防火墙

开放 18090 端口

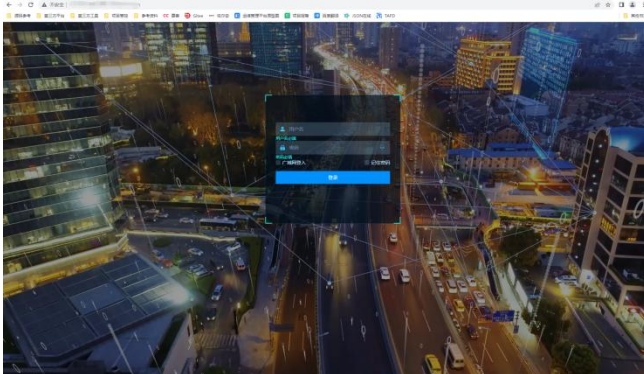
```
Ufw allow 18090
```

## 9.6. 验证是否成功及 WEB 页地址

输入地址：

服务配置页面：<http://nginxIP:18090/serverconfig/>

前端页面：<http://nginxIP:18090>



\*默认用户名/密码：admin/12345,由于还没有启动 jar 包，因此登录不上，可正常打开界面即可。

## 9.7. 使用域名访问平台

### 1) 安装和配置 ssl 证书

- ① 获得 SSL 证书：从可信任的证书颁发机构（CA）或使用自签名证书创建 SSL 证书。
- ② 查看 `/usr/local/nginx/sbin/nginx -V`，如果出现 `(configure arguments: --with-http_ssl_module)`，则已安装（下面的步骤可以跳过，直接进行第 7 步）

```
[root@ecs564b sbin]# /usr/local/nginx/sbin/nginx -V
nginx version: nginx/1.18.0
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-16) (GCC)
built with OpenSSL 1.0.2k-fips 26 Jan 2017
TLS SNI support enabled
configure arguments: --with-http_stub_status_module --with-http_ssl_module
```

③ 一般情况下都是不存在 ssl 模块的，cd /usr/local/software/nginx-1.18.0/。执行：`./configure --prefix=/usr/local/nginx --with-http_stub_status_module --with-http_ssl_module`

```
1 | ./configure --prefix=/usr/local/nginx --with-http_stub_status_module --with-http_ssl_module
```

```
./configure: error: SSL modules require the OpenSSL library.
You can either do not enable the modules, or install the OpenSSL library
into the system, or build the OpenSSL library statically from the source
```

如果提示上面的报错，从官网下载一个 openssl 源码包

- (1) `cd ../software`
- (2) `wget https://www.openssl.org/source/openssl-1.1.1f.tar.gz --no-check-certificate`
- (3) `tar -zxvf openssl-1.1.1f.tar.gz`
- (4) `./configure --prefix=/usr/local/nginx --with-http_stub_status_module --with-http_ssl_module --with-openssl=/usr/local/software/openssl-1.1.1f`

④ 执行 make 进行安装

\*\*\*不要执行 make install，否则会重新安装 nginx

⑤ 进入 nginx 目录停止 nginx 服务，`./nginx -s stop`

⑥ 替换 /sbin/ 目录下生成的 `nginx:cp /usr/local/software/nginx-1.18.0/objs/nginx /usr/local/nginx/sbin`

⑦ 创建证书文件目录：`mkdir https`

⑧ 进入 https：`cd https`

⑨ 将后缀名称为 pem 和 key 的 ssl 证书放在 https 下面

⑩ 编辑 nginx 配置文件：`vi ../conf/nginx.conf`

```
server {
    listen      18090 ssl;

    server_name  gyss.jcyai.com;

    add_header "Access-Control-Allow-Origin" $http_origin always;

    add_header 'Access-Control-Allow-Credentials' 'true' always;

    add_header 'Access-Control-Allow-Methods' 'OPTIONS, POST, GET' always;

    add_header  'Access-Control-Allow-Headers'      'Origin, X-Requested-With,
Content-Type, Accept' always;

    ssl_certificate      /data/local/nginx/https/gyss.jcyai.com_bundle.pem;

    ssl_certificate_key  /data/local/nginx/https/gyss.jcyai.com.key;
```

```

ssl_session_cache    shared:SSL:1m;

ssl_session_timeout  5m;

error_page 497 https://$host:18090$request_uri;
# return 301 https://$host:18090$request_uri;

#charset koi8-r;

#access_log  logs/host.access.log  main;

location / {
    root    webdata;

    index  index.html index.htm;
}

location /api/ {
    proxy_set_header    Host $host;

    proxy_set_header    X-Real-IP $remote_addr;

    proxy_pass http://127.0.0.1:8020/;
}

location /ws/ {
    proxy_set_header    Host $host;

    proxy_set_header    X-Real-IP $remote_addr;

    proxy_pass http://127.0.0.1:9299/;
}

# location /video/ {
#     proxy_set_header    Host $host;
#     proxy_set_header    X-Real-IP $remote_addr;
#     proxy_pass http://127.0.0.1:8888/;
# }

#error_page 404            /404.html;

```

```
# redirect server error pages to the static page /50x.html

#

error_page 500 502 503 504 /50x.html;

location = /50x.html {

    root html;

}
```

⑪ 重启 nginx

```
cd ../
```

```
./sbin/nginx -s reload
```

## 10. nacos 安装

Version: 1.1.3

### 10.1. 下载地址

<http://183.56.218.31:18090/downloads/linux/nacos-server-1.1.3.tar.gz>

### 10.2. 安装步骤

- cd /usr/local/software
- sudo tar -xzf nacos-server-1.1.3.tar.gz
- sudo chown -R \${username} nacos

### 10.3. 设置开机启动

- 1) 下载地址: <http://183.56.218.31:18090/downloads/script/nacos>
- 2) 新建脚本 vi /etc/init.d/nacos, 内容如下:

```
#!/bin/bash

#

# /etc/init.d/nacos

# chkconfig: 2345 63 37

# description: nacos servlet container

# processname: nacos 1.1.3
```

```
# Source function library.

#. /etc/init.d/functions

# source networking configuration.

#. /etc/sysconfig/network


export JAVA_HOME=/usr/local/software/jdk1.8.0_172

export NACOS=/usr/local/software/nacos

case $1 in

    start)

        bash $NACOS/bin/startup.sh -m standalone

        ;;

    stop)

        bash $NACOS/bin/shutdown.sh

        ;;

    restart)

        bash $NACOS/bin/shutdown.sh

        sleep 1

        bash $NACOS/bin/startup.sh

        ;;

esac

exit 0
```

3) cd /etc/init.d

4) 修改 nacos 文件权限

```
chmod 755 nacos
```

5) 增加到启动配置

```
sudo chkconfig nacos on
```



## 10.4. 启动指令

```
bash /usr/local/software/nacos/bin/startup.sh -m standalone
```

\*\*\*startup.sh 开头必须将#!/bin/sh 改成#!/bin/bash

\*\*\*startup.sh 文件中

```
[ ! -e "$JAVA_HOME/bin/java" ] && JAVA_HOME=/usr/java 改为[ ! -e "$JAVA_HOME/bin/java" ]  
&& JAVA_HOME=/usr/local/software/jdk1.8.0_172
```

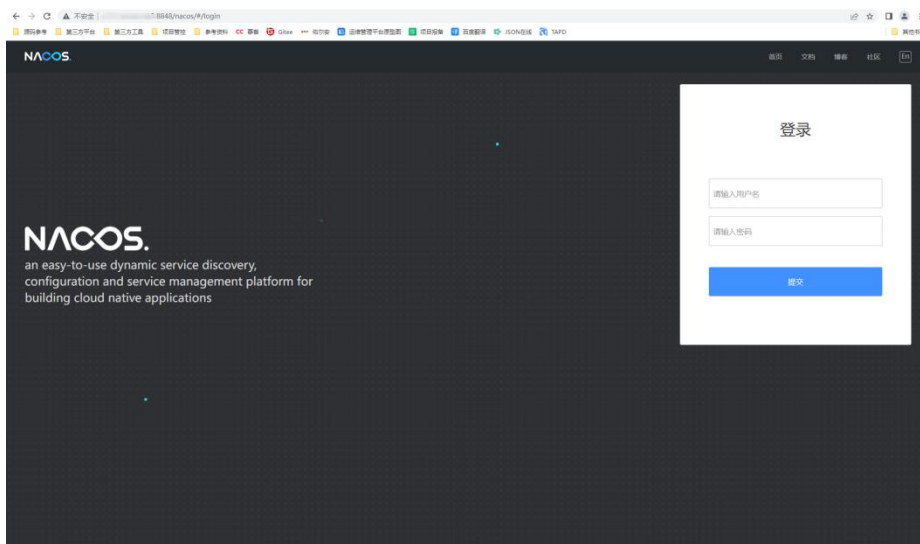
## 10.5. 防火墙开放

放行端口

```
ufw allow 8848
```

## 10.6. 验证是否成功

<http://IP:8848/nacos>



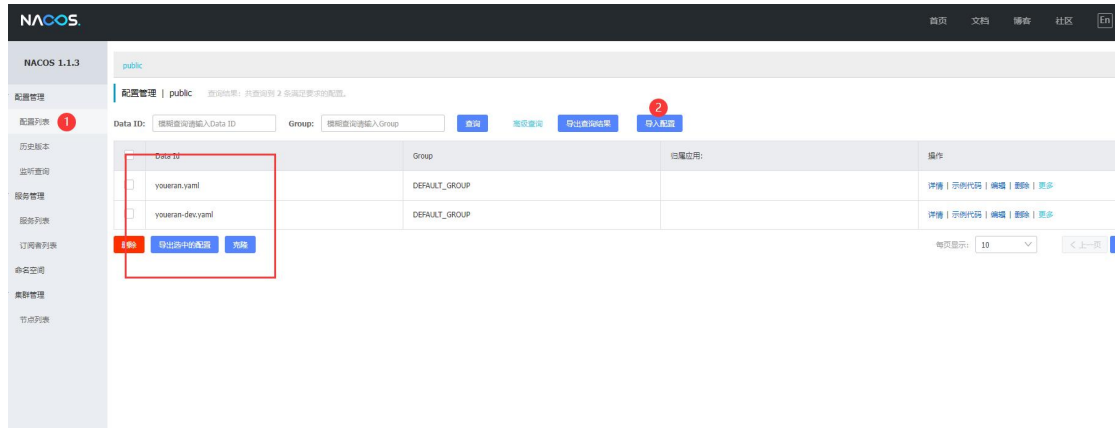
默认用户名: nacos 密码: nacos

## 10.7. 参数配置

请咨询技术工程师获取参数配置包，通过 nacos 配置界面导入，根据项目实际情况做微调即可。//需要调整端口密码 ip 等内容

下载地址：<http://183.56.218.31:18090/downloads/nacosConfig.zip>

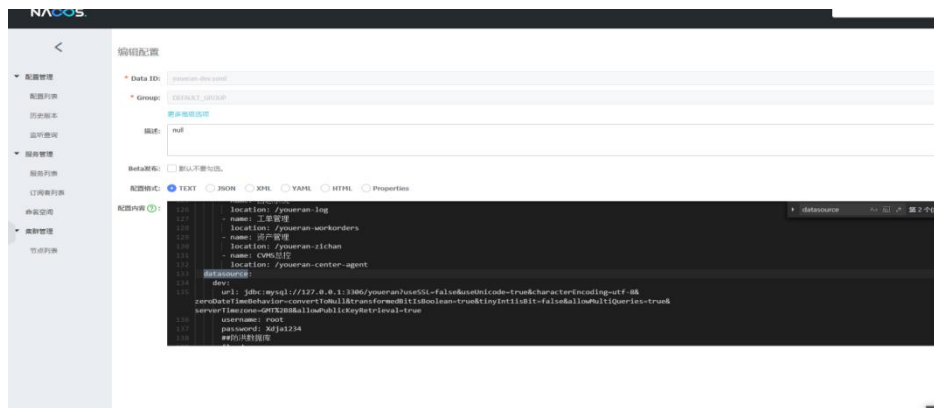
配置管理->导入配置



导入后

配置列表-> youeran-dev.yaml 修改 datasource 下的相关信息：

数据库用户：root，密码：mysql 设置的密码



## 11. ELK 安装（可选择是否部署）

参考链接:

<http://note.youdao.com/noteshare?id=f8140f4220f0f1656e3970b85421c382&sub=A98B018AEE1D4A23981A0F5AE7BBCCD5>

将下载好的 elkconfig 安装包解压之后上传至/usr/local/software, 为了方便之后的安装方便将/usr/local/software/elkconfig/repo 下的四个 repo 文件复制到 /etc/yum.repos.d/之下:

```
cp /usr/local/software/elkconfig/repo *repo /etc/yum.repos.d/
```

\*\*\*如果没有 yum.repos.d 文件夹, 执行: `mkdir /etc/yum.repos.d`

## 11.1. Elasticsearch 安装

1) rpm 设置 (此处执行过一次之后, 后续软件安装无需执行此步骤)

```
rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

2) 进入/usr/local/software/elkconfig/centos7\_64\_rpm 执行以下命令进行安装:

```
yum install elasticsearch-7.8.0-x86_64.rpm
```

3) 将/usr/local/software/elkconfig 下的 elasticsearch.yml 文件替换 /etc/elasticsearch/下的 elasticsearch.yml 或者对照着修改。

4) 设置开机启动 elasticsearch

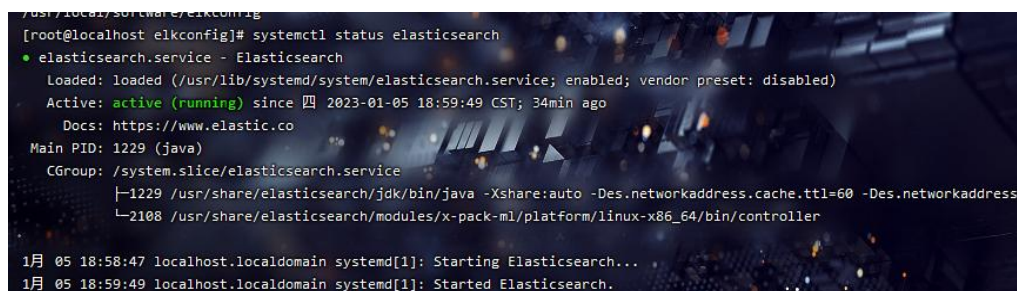
```
systemctl enable elasticsearch.service
```

5) 启动 elasticsearch

```
service elasticsearch start
```

6) 查看 elasticsearch 启动状态:

```
systemctl status elasticsearch
```



```

/usr/local/software/elkconfig
[root@localhost elkconfig]# systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor preset: disabled)
   Active: active (running) since 四 2023-01-05 18:59:49 CST; 34min ago
     Docs: https://www.elastic.co
    Main PID: 1229 (java)
    CGroup: /system.slice/elasticsearch.service
            └─1229 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.ttl=60 -Des.networkaddress
                └─2108 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

1月 05 18:58:47 localhost.localdomain systemd[1]: Starting Elasticsearch...
1月 05 18:59:49 localhost.localdomain systemd[1]: Started Elasticsearch.
  
```

7) 防火墙放开 9200 端口:

```
firewall-cmd --zone=public --add-port=9200/tcp --permanent
```

## 11.2. kibana 安装

1) 进入/usr/local/software/elkconfig/centos7\_64\_rpm 执行以下命令进行安装:

```
yum install kibana-7.8.0-x86_64.rpm
```

2) 将/usr/local/software/elkconfig 下的 kibana.yml 文件替换/etc/kibana 下的 kibana.yml 或者对照着修改。

3) 设置开机启动 kibana.yml

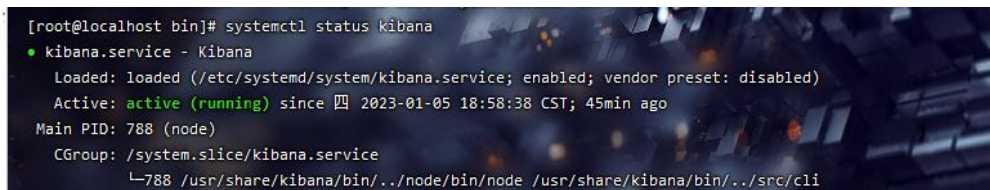
```
systemctl enable kibana.service
```

4) 启动 kibana

```
service kibana start
```

5) 查看 kibana 启动状态:

```
systemctl status kibana
```



```
[root@localhost bin]# systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: disabled)
   Active: active (running) since 四 2023-01-05 18:58:38 CST; 45min ago
     Main PID: 788 (node)
    CGroup: /system.slice/kibana.service
            └─788 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli
```

6) 防火墙放开 5601 端口

```
firewall-cmd --zone=public --add-port=5601/tcp --permanent
```

## 11.3. Logstash 安装

1) 进入/usr/local/software/elkconfig/centos7\_64\_rpm 执行以下命令进行安装:

```
yum install logstash-7.8.0.rpm
```

2) 将/usr/local/software/elkconfig 下的 logstash.yml 文件替换/etc/logstash 下的 logstash.yml 或者对照着修改。

3) 设置开机启动 logstash.yml

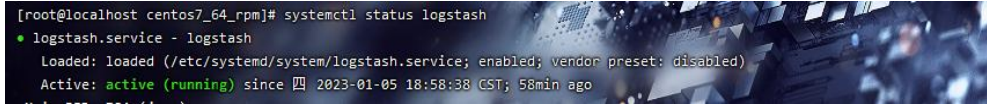
```
systemctl enable logstash.service
```

4) 启动 logstash

```
service logstash start
```

5) 查看 logstash 启动状态:

```
systemctl status logstash
```



```
[root@localhost centos7_64_rpm]# systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: disabled)
   Active: active (running) since 四 2023-01-05 18:58:38 CST; 58min ago
```

## 11.4. filebeat 安装

1) 进入/usr/local/software/elkconfig/centos7\_64\_rpm 执行以下命令进行安装:

```
yum install filebeat-7.8.0-x86_64.rpm
```

2) 将/usr/local/software/elkconfig 下的 filebeat.yml 文件替换/etc/filebeat 下的 filebeat.yml 或者对照着修改。

3) 设置开机启动 filebeat.yml

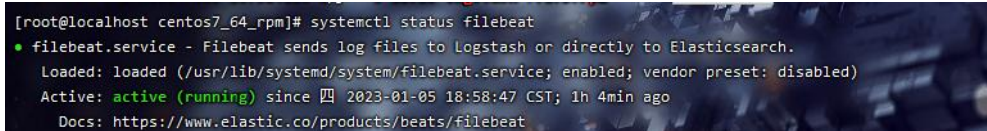
```
systemctl enable filebeat.service
```

4) 启动 filebeat

```
service filebeat start
```

5) 查看 filebeat 启动状态:

```
systemctl status filebeat
```



```
[root@localhost centos7_64_rpm]# systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
   Loaded: loaded (/usr/lib/systemd/system/filebeat.service; enabled; vendor preset: disabled)
   Active: active (running) since 四 2023-01-05 18:58:47 CST; 1h 4min ago
   Docs: https://www.elastic.co/products/beats/filebeat
```

6) 防火墙放开 9000、5044 端口

```
firewall-cmd --zone=public --add-port=5044/tcp --permanent
```

```
firewall-cmd --zone=public --add-port=9000/tcp --permanent
```

## 12. jar 包部署

### 12.1. 下载地址

核心 jar 包: <http://183.56.218.31:18090/downloads/jar/jar.tar.gz>

### 12.2. 路径约定

所有 jar 包拷贝到/usr/local/jar 文件夹, 按照 jar 包名称在此目录下建立子文件夹, 然后依次将指定.jar 及依赖文件拷贝到该子文件夹下, 如下图:

```
[root@VM-0-2-centos local]# cd jar/
[root@VM-0-2-centos jar]# ll
total 32
drwxr-xr-x 2 root root 4096 Sep 20 17:02 alarm
drwxr-xr-x 2 root root 4096 Sep 20 17:03 arithmetic
drwxr-xr-x 2 root root 4096 Sep 20 17:04 camera
drwxr-xr-x 2 root root 4096 Sep 20 17:02 center
drwxr-xr-x 2 root root 4096 Sep 20 17:04 log
drwxr-xr-x 2 root root 4096 Sep 20 17:05 report
drwxr-xr-x 2 root root 4096 Sep 20 17:06 websocket
drwxr-xr-x 2 root root 4096 Sep 20 17:06 wisdom
[root@VM-0-2-centos jar]#
```

## 12.3. jar 通用启动脚本格式

\*相关脚本在对应微服务已经生成了脚本文件

```
java -jar -Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=19696 -Xms512M -Xmx2048M
```

```
-XX:PermSize=512M -XX:MaxPermSize=1024M -Dfile.encoding=utf-8 jarPackageName.jar --server.port=8120
```

```
--spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848
```

```
--spring.cloud.nacos.config.server-addr=127.0.0.1:8848 --youeran.log.elk.destination=127.0.0.1:9000
```

- 1) 增加远程调试选项, 19696: 调试监听端口 suspend=n 表示自动运行, suspend=y 表示启动监听, 等待客户端远程连接
- 2) 是否开启日志记录到 elk, 这里需要部署 elk, 默认不开启, 此选项可删除
- 3) 指定 JVM 内存参数, 如果不需要可关闭
- 4) 服务监听端口
- 5) 指向注册中心 nacos

## 12.4. jar 启动

每一个子文件夹下有一个启动脚本, 如微服务 alarm:

```
cd alarm
```

```
[root@VM-0-2-centos jar]# cd alarm
[root@VM-0-2-centos alarm]# ll
total 95072
-rw-r--r-- 1 root root 1114 Sep 20 17:09 service.sh
-rw-r--r-- 1 root root 97349164 Sep 20 17:03 youeran-alarm.jar
[root@VM-0-2-centos alarm]#
```

增加脚本执行权限: `chmod +x service.sh`

运行脚本: `./service.sh start`

查看日志: `tail -f l.out`

退出日志: `ctrl + c`

//显示端口开放表示开启成功, 进入 nacos 界面刷新查看,

然后点服务管理，服务列表，刷新时候查看 data id 有没有启动的服务，

如下图：

```
[root@VM-0-2-centos alarm]# chmod +x service.sh
[root@VM-0-2-centos alarm]# ll
total 95072
-rwxr-xr-x 1 root root 1114 Sep 20 17:09 service.sh
-rw-r--r-- 1 root root 97349164 Sep 20 17:03 youeran-alarm.jar
[root@VM-0-2-centos alarm]# ./service.sh start
[root@VM-0-2-centos alarm]# tail -f l.out
----启动中，读取到的环境变量:[dev], jar地址:[file:/usr/local/jar/alarm/youeran-alarm.jar]----
17:15:36.272 [main] INFO com.youeran.common.launch.LauncherServiceImpl - spring.cloud.nacos.discovery.server-addr is not null
::
:: <-->
::
::
:: youeran :: youeran-alarm :: Running SpringBoot 2.1.11.RELEASE ::
```

其它微服务同理，只需启动脚本即可

配置视频动态链接库：

下载地址：<http://183.56.218.31:18090/downloads/linux/lib.tar.gz>

将下载好的 lib 包解压放在/usr/local/jar 文件夹下

```
sudo vi /etc/ld.so.conf.d/youeran.conf
```

```
/usr/local/jar/lib/youeran
```

```
/usr/local/jar/lib/hik
```

```
/usr/local/jar/lib/isup
```

```
/usr/local/jar/lib/dahua
```

```
/usr/local/jar/lib/uniview
```

```
/usr/local/jar/lib/tiandy
```

```
/usr/local/jar/lib/arcssoft
```

## 12.5. 数据中心配置

youeran-center-agent.jar 是整个平台的数据中心，集成了设备/用户认证及 GA/T 1400、GB/T 28181 及设备 SDK 接入，做为分布式部署方案，GA/T 1400、GB/T 28181、设备 SDK 接入可以做为独立微服务运行，具体另行提供独立微服务。

该微服务运行前需先生成配置参数文件

- 下载地址：<http://183.56.218.31:18090/downloads/script/javaConfig.ini>
- 准备工作

新建配置文件 vi /usr/local/jar/center/lib/config/javaConfig.ini



内容如下：

```
[MYSQL]

IP = 127.0.0.1          # mysql 地址


PORT = 3306            # mysql 端口

USERNAME = root         # mysql 用户名

PASSWORD = 123456      # mysql 密码

ENABLED = true         # 是否启动 mysql

URL =                  # 保留
```

	注意	# 该文件不支持注释，文件里应该删除注释相关文字
---	----	--------------------------

- jar 启动  
见前一节

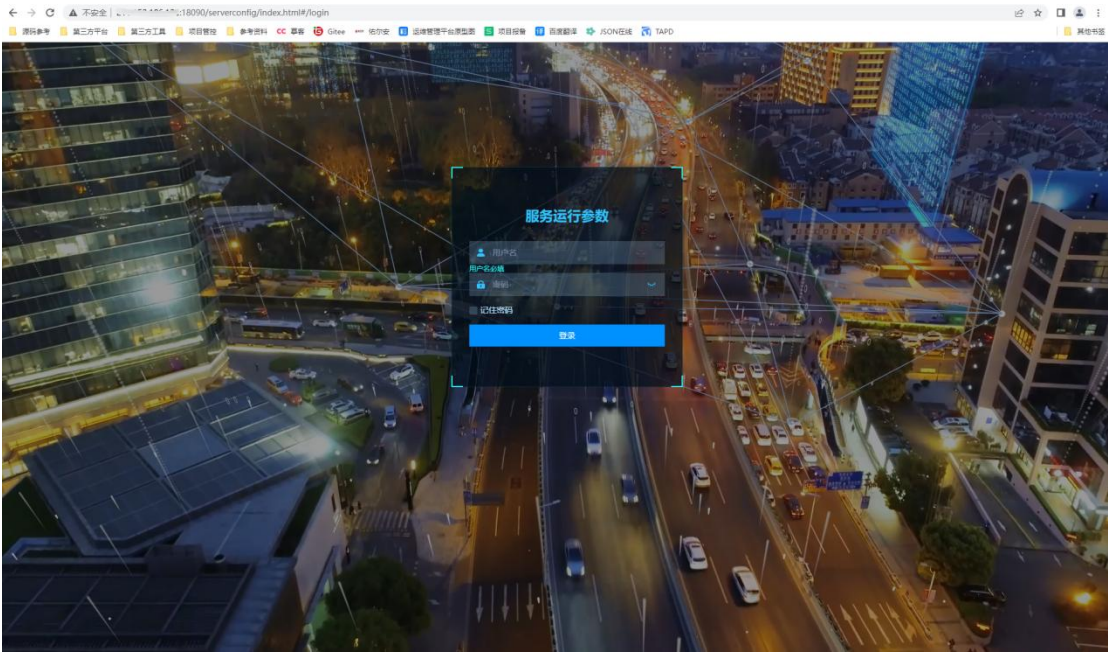
20) 开启防火墙

```
ufw allow 8020

ufw allow 9090
```

- 验证是否成功

输入地址：<http://IP:18090/serverconfig/index.html>

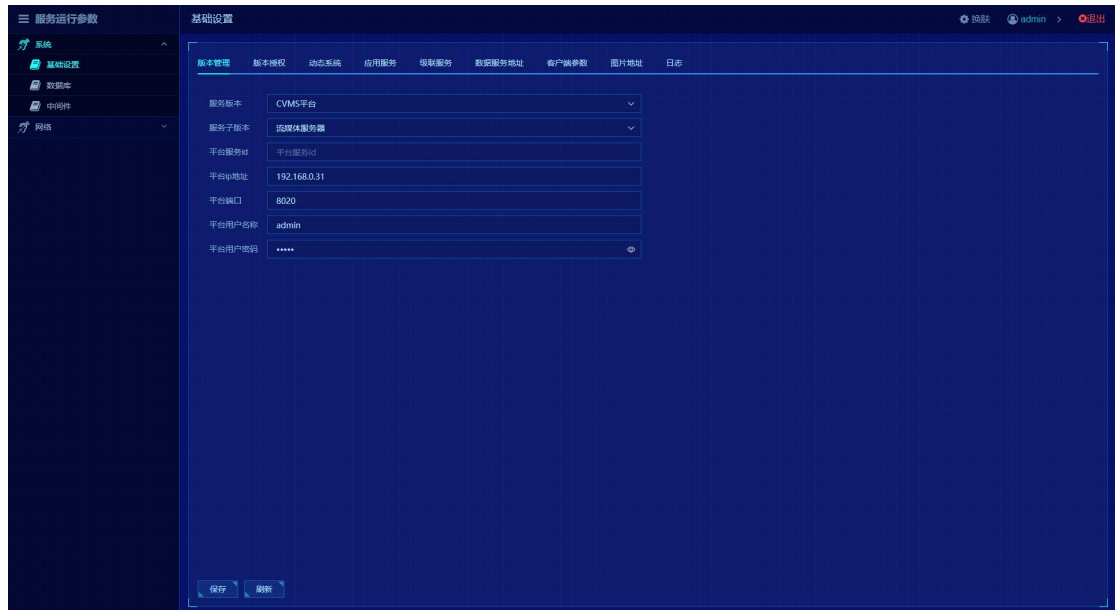


默认用户名：admin 密码：12345



## ● 详细参数配置

点击登录后，如下图，可对具体参数进行配置



## 12.6. 验证 JAR 微服务是否运行

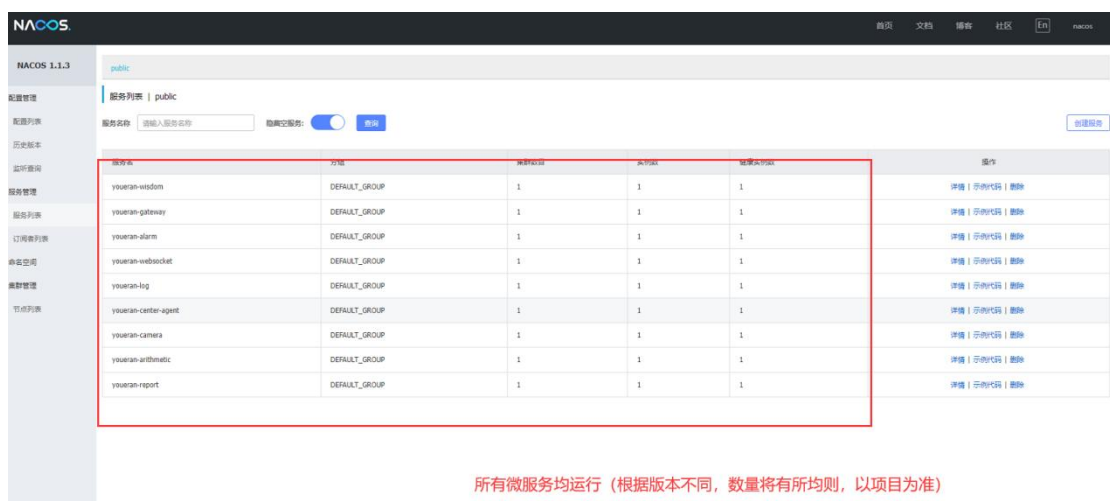
通过 nacos 页面验证

地址：<http://IP:8848/nacos>

默认用户名/密码：`nacos/nacos`

服务管理->服务列表，查看微服务的状态。

每个 jar 包对应 1 个微服务，开启后，nacos 应该可检测到其状态。如图：



## 13. 部署 node.js 环境

### 13.1. 下载地址

<http://183.56.218.31:18090/downloads/linux/nodejs.tar.gz>

### 13.2. 安装步骤

1) `cd /usr/local/software`

2) `tar -xvf nodejs.tar.gz`

3) `cd /usr/local/software/nodejs/bin`

4) 创建软连接:

```
ln -s ../lib/node_modules/corepack/dist/corepack.js corepack
ln -s ../lib/node_modules/koa-generator/bin/koa
ln -s ../lib/node_modules/koa-generator/bin/koa2
ln -s ../lib/node_modules/npm/bin/npm-cli.js npma
ln -s ../lib/node_modules/npm/bin/npm-cli.js npx
ln -s ../lib/node_modules/pm2/bin/pm2
ln -s ../lib/node_modules/pm2/bin/pm2-dev
ln -s ../lib/node_modules/pm2/bin/pm2-docker
ln -s ../lib/node_modules/pm2/bin/pm2-runtime
ln -s ../lib/node_modules/yarn/bin/yarn.js yarn
ln -s ../lib/node_modules/yarn/bin/yarn.js yarnpkg
ln -s /usr/local/software/nodejs/bin/npm /usr/local/bin/npm
ln -s /usr/local/software/nodejs/bin/node /usr/local/bin/node
ln -s /usr/local/software/nodejs/bin/npx /usr/local/bin/npx
ln -s /usr/local/software/nodejs/bin/pm2 /usr/local/bin/pm2
ln -s /usr/local/software/nodejs/bin/yarn /usr/local/bin/yarn
```

5) `node -v, yarn -v, npm -v, yarn -v, npx -v, pm2 -v` 查看各个服务的版本:

```
[root@localhost bin]# node -v
v16.13.0
[root@localhost bin]# yarn -v
1.22.19
[root@localhost bin]# npm -v
9.2.0
[root@localhost bin]# yarn -v
1.22.19
[root@localhost bin]# npx -v
9.2.0
```

[illegible]

## 14. ffmpeg 完整安装

### 14.1. 下载地址

<http://183.56.218.31:18090/downloads/linux/pkg-config-0.29.2.tar.gz>

<http://183.56.218.31:18090/downloads/linux/yasm-1.3.0.tar.gz>

<http://183.56.218.31:18090/downloads/linux/x264.tar.gz>

<http://183.56.218.31:18090/downloads/linux/ffmpeg-4.1.tar.gz>

## 14.2. 安装步骤

- ```
1) cd /usr/local/software
2) tar -xvf pkg-config-0.29.2.tar.gz
3) cd pkg-config-0.29.2
```

4) `./configure --with-internal-glib --enable-shared`

5) `make && make install`

6) `cd ../`

7) `tar -xvf yasm-1.3.0.tar.gz`

8) `cd yasm-1.3.0`

9) `./configure`

10) `make && make install`

11) `yasm --version`

```
[root@localhost yasm-1.3.0]# yasm --version
yasm 1.3.0
Compiled on Mar 16 2023.
Copyright (c) 2001-2014 Peter Johnson and other Yasm developers.
Run yasm --license for licensing overview and summary.
```

12) `cd ../`

13) `tar -xvf x264.tar.gz`

14) `cd x264`

15) `./configure --prefix=/usr/local/x264/ --includedir=/usr/local/include  
--libdir=/usr/local/lib --enable-shared --disable-asm`

16) `make && make install`

17) `cd ../`

18) `tar -xvf ffmpeg-4.1.tar.gz`

19) `cd ffmpeg-4.1/`

20) `./configure --prefix=/usr/local/ffmpeg --enable-gpl --enable-shared --enable-libx264`

21) `make && make install`

22) `cd yasm-1.3.0`

23) `./configure`

24) `make && make install`

25) 配置环境变量: `vi /etc/profile`

`export FFMPEG_PATH=/usr/local/ffmpeg`

`export CLASSPATH=$FFMPEG_PATH/lib:$CLASSPATH`

`PATh=$FFMPEG_PATH/bin:$PATh`

```
unset i
unset -f pathmunge
#java environment
export JAVA_HOME=/usr/local/software/jdk1.8.0_172
export CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export PATH=$PATH:$JAVA_HOME/bin
export PATH=$PATH:/usr/local/mysql/bin
export FFmpeg_PATH=/usr/local/ffmpeg
export CLASSPATH=$FFmpeg_PATH/lib:$CLASSPATH
PATH=$FFmpeg_PATH/bin:$PATH
-- INSERT --
```

执行 `source /etc/profile`，使环境变量生效。

26) 配置 ffmpeg 启动目录: `vi /etc/ld.so.conf`

`/usr/local/ffmpeg/lib`

`/usr/local/lib`

27) 重新加载配置: `ldconfig`

28) 查看版本: `ffmpeg -v`

```
[root@localhost bin]# ffmpeg -v
ffmpeg version 4.1 Copyright (c) 2000-2018 the FFmpeg developers
built with gcc 4.8.5 (GCC) 20150623 (Red Hat 4.8.5-16)
configuration: --prefix=/usr/local/ffmpeg --enable-gpl --enable-shared --enable-libx264
libavutil      56. 22.100 / 56. 22.100
libavcodec     58. 35.100 / 58. 35.100
libavformat    58. 20.100 / 58. 20.100
libavdevice    58.  5.100 / 58.  5.100
libavfilter    7. 40.101 / 7. 40.101
libswscale     5.  3.100 / 5.  3.100
libswresample  3.  3.100 / 3.  3.100
libpostproc   55.  3.100 / 55.  3.100
Missing argument for option 'v'.
Error: splitting the argument list: Invalid argument
```

## 15. 启动转码服务

### 15.1. 下载地址

<http://183.56.218.31:18090/downloads/linux/flvnode.tar.gz>

### 15.2. 安装步骤

1) `cd /usr/local/software`

2) `tar -xvf flvnode.tar.gz`

3) `cd flvnode/app/config`

4) 依据实际情况，填写对应的端口及流媒体 IP 等信息，参考下图：

```
// 这里的参数改变需要重启pm2
const config = {
  wsPort: 8888, //rtspwebspcket端口
  mediaServerIp: "192.168.0.201", //流媒体ip
  mediaServerMapIp: "192.168.0.201", //流媒体映射ip
  mediaServerPort: 3027, //RTMP流媒体端口
  httpPort: 8886, //http端口,对应配置界面转码端口
  mediaHttpPort: 8887, //FLV流媒体http端口
  gopCache: true, //是否缓存GOP
  fps: 25, //输入帧数
};
module.exports = config;
```

5) 防火墙开放对应的端口号，如下图所示：

```
ufw allow 8886
```

```
ufw allow 8887
```

```
Ufw allow 8888
```

```
Ufw 3027
```

## 16. FAQ

### 16.1. GA/T 1400 检查参数配置

#### 16.1.1. 数据库配置



#### 16.1.2. 图片保存路径设置

设置图片本地保存及远程访问路径

本地图片保存路径

</usr/local/nginx/webdata/picture/>

图片访问地址内网

<http://192.168.0.XX:18090/picture/>

图片访问地址外网

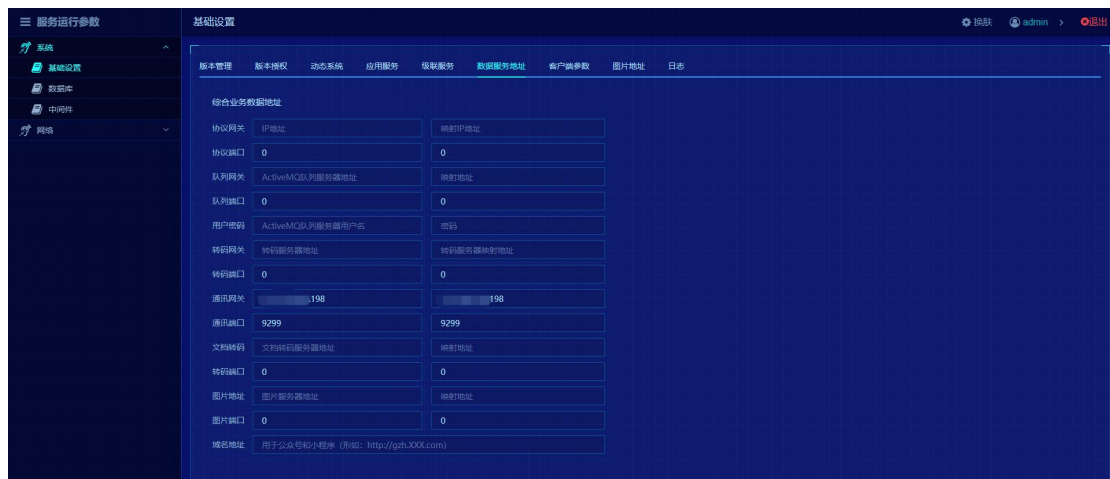
<http://192.168.0.XX:18090/picture/>



### 16.1.3. WebSocket 通讯地址

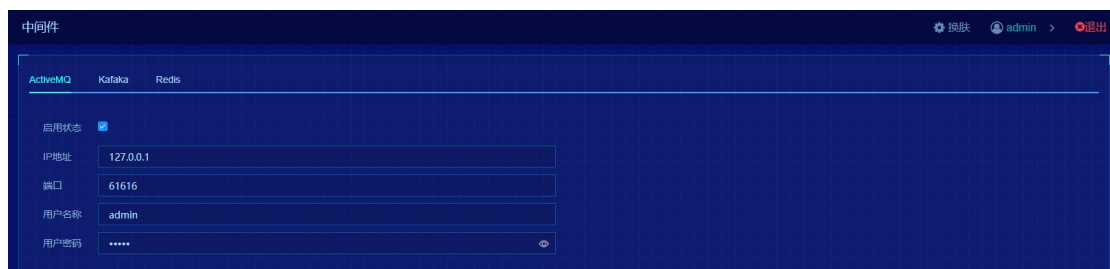
未配置【通讯网关】地址，浏览器将无法实现 websocket 交互，同样也获取不到

Ufw allow 9299



### 16.1.4. ActiveMQ 参数配置

GA/T 1400 写入队列配置信息





## 16.1.5. nacos 参数配置

