

平 台 日 常 运 维 手 册

1.1. 编写目的	1
1.2. 声明	1
1.3. 运行环境	1
1.4. 定义	1
1.5. 符号约定	3
2. 微服务结构图	3
3. 注意事项	3
3.1. LINUX 基础教程	3
3.2. WINDOWS 下打包.TAR.GZ	4
3.3. 安装包常用指令	4
3.4. 文件上传指令	4
3.5. 开机启动项查看指令	5
3.6. LSOFF 命令	5
3.7. TCPDUMP 使用	6
3.8. IP 地址操作	6
3.9. 端口操作	6
3.10. 进程操作	6
3.11. 目录操作	8
3.12. VI 常规操作	8
3.13. 防火墙操作指令	8
3.14. SOURCE 指令	9
3.15. 路径约定	9
4. 常见问题	9
4.1. 广域网可正常登录，局域网登录提示端口和 IP 地址错误	9
4.2. 导出文件失败	10
4.3. 流媒体在线，无法播放视频	10
4.4. GB/T28181 如何获取传输过来的端口号	10
4.5. 图像系统，无法查找到图片，GA/T1400 已经正常接入	10
4.6. ARITHMETIC 微服务启动报错	11
4.7. MYSQL 启动报错	11
4.8. ACTIVEMQ 服务相关命令	11
4.9. MYSQL 服务相关命令	12
4.10. REDIS 服务相关命令	12
4.11. NGINX 服务相关命令	13
4.12. NACOS 服务相关命令	13
4.13. 转码服务相关命令	13
4.14. JAR 包服务相关命令	14
4.15. I.O.U.T 日志过大，设置日志滚动	14

引言

1.1. 编写目的

为项目系统运维管理人员提供技术支撑

1.2. 声明

- a. 软件名称：柳州智慧安防小区管理平台
- b. 使用用户：运维管理人员
- c. 本手册适用所有 linux 版本运维管理
- 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	CentOS 7.6	CentOS 7.6



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

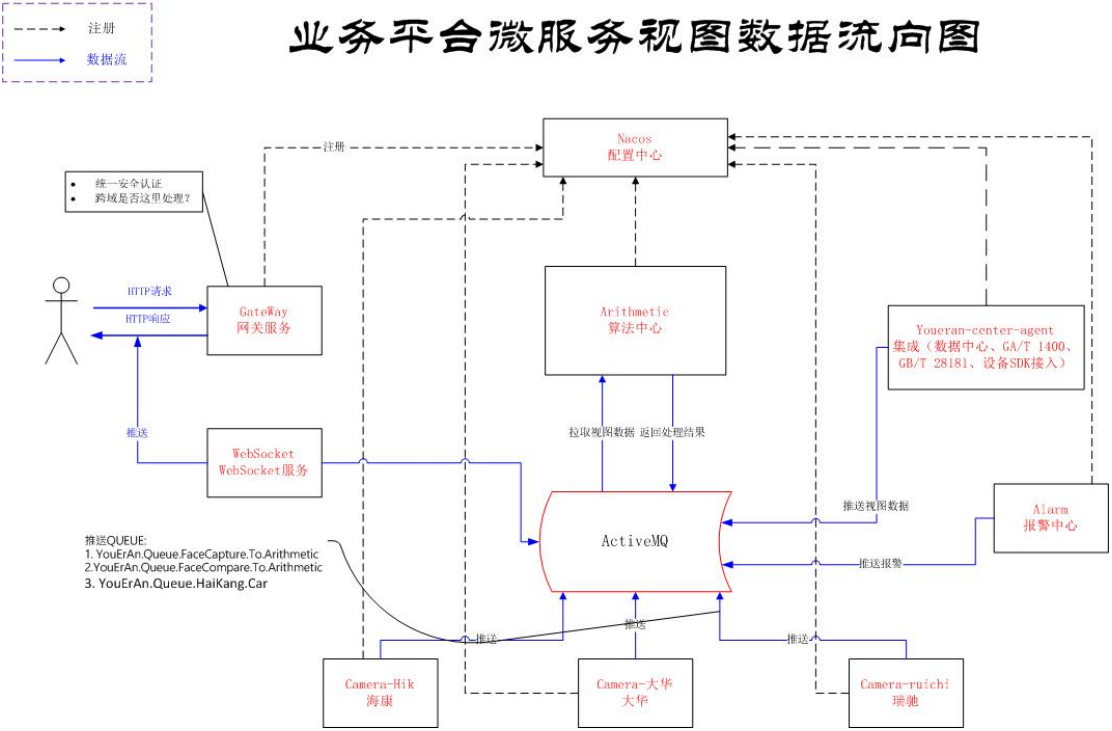
中心服务	youeran-center-agent.jar	9090	中心平台 (集成数据中心、GA/T 1400、GB/T 28181、设备 SDK 接入)
业务网关微服务	youeran-gateway.jar	8020	提供路由、权限认证
代理微服务	youeran-agent.jar	9580	统一注册服务中心, 监听每个服务是否在线
设备管理微服务	youeran-camera.jar	9501	提供设备管理服务
算法微服务	youeran-arithmetic.jar	9800	提供人、车等算法
报警处理微服务	youeran-alarm.jar	9801	告警统一服务中心
报表处理微服务	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

本章介绍部署的安装路径及 linux 相关命令介绍，运维人员可参考。

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. 安装包常用指令

- 查看系统自带的 mariadb: `rpm -qa | grep XXX`
- 卸载系统自带的 mariadb: `rpm -e --nodeps XXX`
- yum 语法

- 1) 使用 yum 查找软件包

命令: `yum search`

- 2) 列出所有可安装的软件包

命令: `yum list`

- 3) 列出所有可更新的软件包

命令: `yum list updates`

- 4) 列出所有已安装的软件包

命令: `yum list installed`

3.4. 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。

3.5. 开机启动项查看指令

```
chkconfig --list [packageName]
```

```
[root@VM-0-2-centos mysql]# chkconfig --list
Note: This output shows SysV services only and does not include native
systemd services. SysV configuration data might be overridden by native
systemd configuration.

If you want to list systemd services use 'systemctl list-unit-files'.
To see services enabled on particular target use
'systemctl list-dependencies [target]'.

activemq      0:off  1:off  2:on   3:on   4:on   5:on   6:off
mysqld        0:off  1:off  2:on   3:on   4:on   5:on   6:off
netconsole    0:off  1:off  2:off  3:off  4:off  5:off  6:off
network       0:off  1:off  2:on   3:on   4:on   5:on   6:off
[root@VM-0-2-centos mysql]#
```

3.6. lsof 命令

参考地址: <https://mp.weixin.qq.com/s/nG5991ishWryIsiEAZWH1Q>

列出打开文件(list openfiles)，在 unix 中的一切（包括网络套接字）皆为文件，那就可以做很多事了，号称神器。

```
usage: [-?abhlnoPRstUvV] [+|-c c] [+|-d s] [+D D] [+|-f[cgG]]
```

```
[-F [f]] [-g [s]] [-i [i]] [+|-L [l]] [+|-M] [-o [o]]
```

```
[-p s] [+|-r [t]] [-S [t]] [-T [t]] [-u s] [+|-w] [-x [f1]] [--] [names]
```

常用选项:

默认：没有选项，lsof 列出活跃进程的所有打开文件

组合：可以将选项组合到一起，如-abc，但要当心哪些选项需要参数

-a：结果进行“与”运算（而不是“或”）

-l：在输出显示用户 ID 而不是用户名

-h：获得帮助

-t：仅获取进程 ID

-U：获取 UNIX 套接口地址

-F：格式化输出结果，用于其它命令。可以通过多种方式格式化，如-F pcfm（用于进程 id、命令名、文件描述符、文件名，并以空终止）

语法: lsof -i[46] [protocol][@hostname|hostaddr][:service|port]

3.7. 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

3.8. IP 地址操作

- 查看 IP

```
ip addr
```

- 修改 IP 地址

方法 1: `nmtui` (图形化修改方式)

方法 2: 以 `root` 账号登录, 直接修改配置文件

```
cd /etc/sysconfig/network-scripts/
```

```
ll
```

```
vi ifcfg-XXX, xxx 为指定的网卡配置文件
```

- 重启网络服务

```
service network restart
```

- 参考网址

<https://www.php.cn/centos/462782.html>

3.9. 端口操作

查看端口: `netstat -antup / lsof -i:${port}`

3.10. 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，或是在交谈式指令列(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，将交谈式指令取消，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
```

3.11. 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
```

3.12. 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

3.13. 防火墙操作指令

- 安装指令: yum install firewalld firewalld-config
- 开启防火墙命令: systemctl start firewalld.service
- 重启防火墙命令: firewall-cmd --reload 或 service firewalld restart
- 禁用防火墙命令: systemctl stop firewalld
- 开启启动防火墙: systemctl enable firewalld
- 停止并禁用: systemctl disable firewalld
- 查看防火墙状态: systemctl status firewalld 或 firewall-cmd --state

- 查看端口列表: `firewall-cmd --permanent --list-port`
- 开启 TCP 端口: `firewall-cmd --zone=public --add-port=${PORT}/tcp --permanent`
- 开启 UDP 端口: `firewall-cmd --zone=public --add-port=${PORT}/udp --permanent`
- 关闭 TCP 端口: `firewall-cmd --zone=public --remove-port=${PORT}/tcp --permanent`
- 关闭 UDP 端口: `firewall-cmd --zone=public --remove-port=${PORT}/udp --permanent`
- 批量 TCP 端口: `firewall-cmd --zone=public --add-port=${PORT}-${PORT}/tcp --permanent`
- 批量 UDP 端口: `firewall-cmd --zone=public --add-port=${PORT}-${PORT}/udp --permanent`

3.14. source 指令

linux source 功能: 使当前 shell 读入路径为 filepath 的 shell 文件并依次执行文件中的所有语句, 通常用于重新执行刚修改的初始化文件, 使之立即生效, 而不必注销并重新登录。例如, 当我们修改了 /etc/profile 文件, 并想让它立刻生效, 而不用重新登录, 就可以使用 source 命令, 如 `source /etc/profile`

***如果/etc/profile 查找不到时, 可执行 `source /etc/profile`

3.15. 路径约定

- 软件拷贝目录: /usr/local/software 或 /home/local/software
- jar 包拷贝目录: /usr/local/jar 或 /home/local/jar
- 转码服务包拷贝目录: /usr/local/software/flvnode 或 /home/local/software/flvnode

可通过 Xshell 拖动等方式上传相关包至指定目录。

4. 常见问题

以下所有常见问题: 默认 jar 包的路径都为: /usr/local/jar

4.1. 广域网可正常登录, 局域网登录提示端口和 ip 地址错误

JAR 包下的服务启动之后修改 IP 地址, 则 camera 监听的服务找不到, 需要重启 jar 包下的 camera 服务。

解决方案: `cd /usr/local/jar/camera`

`./service.sh restart`

4.2. 导出文件失败

检查 nacos, excel 导出的配置信息

```

259 # 低延时设备中运行该程序
260 concurrency: 3
261 # 有多种模式
262 ack-mode: RECORD
263 #默认参数先置空, 具体看场景再调整
264 excel:
265   dev:
266     username: root
267     password: 12345
268     excelfile: \\192.168.0.33\theme\Excel
269     linux-excel-file: /usr/local/nginx/webdata/picture/Excel/
270     linux-excel-url: http://192.168.0.33:18090/picture/a/Excel/
271     excelurl: http://192.168.0.33:18090/picture/a/Excel/
272 #虹软算法
273 arcsoft:
274   dev:

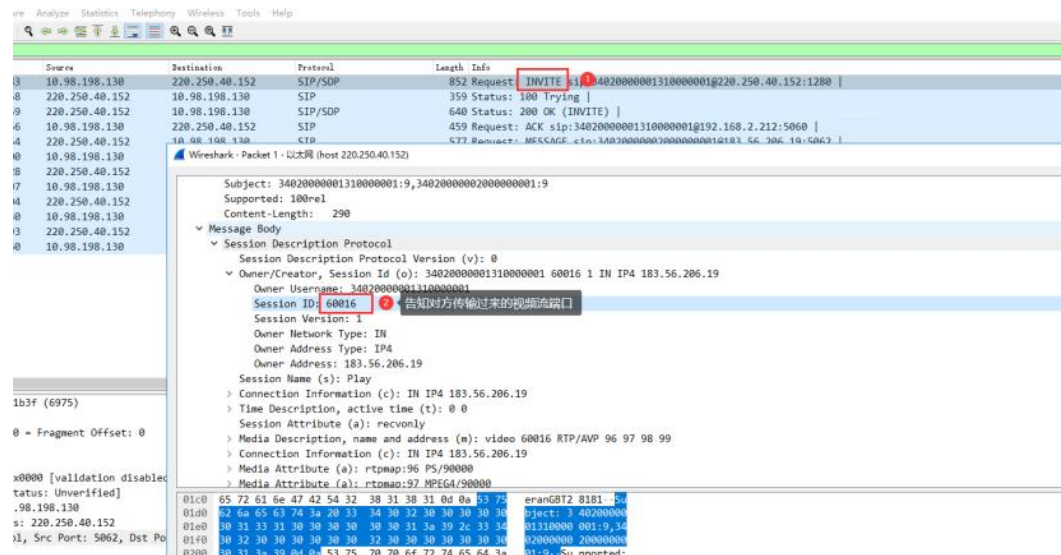
```

4.3. 流媒体在线, 无法播放视频

1. 登录软件, F12 查看视频播放的端口, telnet 查看是否服务正常启动
2. 使用 VLC 播放视频, 看是否可正常播放
3. 查看服务器第一步里面的端口是否正常被流媒体使用

4.4. GB/T28181 如何获取传输过来的端口号

使用 wireshark 抓包工具, 通过 invite 指令, 查看端口。



4.5. 图像系统, 无法查找到图片, GA/T1400 已经正常接入

1. wireshark 抓包查找是否触发抓拍人脸、图像、人体等接口
2. ACTIVEMQ 服务是否正常启动
3. 本地是否存在图片

4. 数据库是否记录
5. 算法微服务是否正常启动

4.6. arithmetic 微服务启动报错

```
Caused by: org.springframework.messaging.converter.MessageConversionException: Cannot convert from [java.util.HashMap] to [java.lang.String] for org.springframework.messaging.handler.annotation.support.PayloadArgumentResolver.resolveArgument(PayloadArgumentResolver.java:144)
    at org.springframework.messaging.handler.invocation.HandlerMethodArgumentResolverComposite.resolveArgument(HandlerMethodArgumentResolverComposite.java:117)
    at org.springframework.messaging.handler.invocation.InvocableHandlerMethod.getMethodArgumentValues(InvocableHandlerMethod.java:148)
    at org.springframework.messaging.handler.invocation.InvocableHandlerMethod.invoke(InvocableHandlerMethod.java:116)
    at org.springframework.messaging.handler.invocation.InvocableHandlerMethod.invoke(InvocableHandlerMethod.java:114)
    ... 10 common frames omitted

2022-12-29 17:20:37.716 WARN 14108 --- [enerContainer-1] o.s.j.l.DefaultMessageListenerContainer : Execution of JMS message listener failed, and no ErrorHandler has been set.

org.springframework.jms.listener.adapter.ListenerExecutionFailedException: Listener method could not be invoked with incoming message
Endpoint handler details:
Method [public void com.youeran.arithmetic.listener.activemq.ReceiveConsumer.getHaiKangCarInfo(java.lang.String)]
Bean [com.youeran.arithmetic.listener.activemq.ReceiveConsumer@7a0b7ec0]
Nested exception is org.springframework.messaging.converter.MessageConversionException: Cannot convert from [java.util.HashMap] to [java.lang.String] for org.springframework.messaging.handler.annotation.support.PayloadArgumentResolver.resolveArgument(PayloadArgumentResolver.java:144)
    at org.springframework.messaging.handler.invocation.HandlerMethodArgumentResolverComposite.resolveArgument(HandlerMethodArgumentResolverComposite.java:117)
    at org.springframework.messaging.handler.invocation.InvocableHandlerMethod.getMethodArgumentValues(InvocableHandlerMethod.java:148)
    at org.springframework.messaging.handler.invocation.InvocableHandlerMethod.invoke(InvocableHandlerMethod.java:116)
    at org.springframework.messaging.handler.invocation.InvocableHandlerMethod.invoke(InvocableHandlerMethod.java:114)
    ... 10 common frames omitted
Caused by: org.springframework.messaging.converter.MessageConversionException: Cannot convert from [java.util.HashMap] to [java.lang.String] for org.springframework.messaging.handler.annotation.support.PayloadArgumentResolver.resolveArgument(PayloadArgumentResolver.java:144)
    at org.springframework.messaging.handler.invocation.HandlerMethodArgumentResolverComposite.resolveArgument(HandlerMethodArgumentResolverComposite.java:117)
    at org.springframework.messaging.handler.invocation.InvocableHandlerMethod.getMethodArgumentValues(InvocableHandlerMethod.java:148)
    at org.springframework.messaging.handler.invocation.InvocableHandlerMethod.invoke(InvocableHandlerMethod.java:116)
    at org.springframework.messaging.handler.invocation.InvocableHandlerMethod.invoke(InvocableHandlerMethod.java:114)
    ... 10 common frames omitted
```

算法启动访问队列报错，更新算法。

4.7. mysql 启动报错

服务器磁盘满了，清理空间，重启 msyql 即可。

```
[root@localhost ~]# systemctl status mysql
● mysqld.service - LSB: start and stop MySQL
   Loaded: loaded (/etc/rc.d/init.d/mysqld; bad; vendor preset: disabled)
   Active: failed (Result: exit-code) since Fri 2022-12-30 10:15:37 CST; 13min ago
     Docs: man:systemd-sysv-generator(8)
    Process: 1852 ExecStart=/etc/rc.d/init.d/mysqld start (code=exited, status=1/F
    AILURE)

Dec 30 10:15:36 localhost.localdomain systemd[1]: Starting LSB: start and sto...
Dec 30 10:15:37 localhost.localdomain mysqld[1852]: Starting MySQL. ERROR! Th...
Dec 30 10:15:37 localhost.localdomain systemd[1]: mysqld.service: control pro...
Dec 30 10:15:37 localhost.localdomain systemd[1]: Failed to start LSB: start ...
Dec 30 10:15:37 localhost.localdomain systemd[1]: Unit mysqld.service entered...
Dec 30 10:15:37 localhost.localdomain systemd[1]: mysqld.service failed.
Hint: Some lines were ellipsized, use -l to show in full.
```

使用命令 `du -ah --max-depth=1` 查找当前文件夹最大文件，删除无用文件。

4.8. Activemq 服务相关命令

`systemctl status activemq` 查询 Activemq 服务启动状态

```
[root@localhost center]# systemctl status activemq
● activemq.service - SYSV: activemq ....
   Loaded: loaded (/etc/rc.d/init.d/activemq; bad; vendor preset: disabled)
   Active: active (running) since 四 2023-03-16 14:49:28 CST; 1 weeks 6 days ago
     Docs: man:systemd-sysv-generator(8)
   Process: 792 ExecStart=/etc/rc.d/init.d/activemq start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/activemq.service
           └─870 /usr/local/software/jdk1.8.0_172/bin/java -Xms64M -Xmx1G -Djava.util.logging.config.file=logging.properties -Djava.security.auth.login.config=

3月 16 14:49:28 localhost.localdomain systemd[1]: Starting SYSV: activemq .....
3月 16 14:49:28 localhost.localdomain activemq[792]: Starting activemq...
3月 16 14:49:28 localhost.localdomain activemq[792]: INFO: Loading '/usr/local/software/apache-activemq-5.15.13/bin/env'
3月 16 14:49:28 localhost.localdomain activemq[792]: INFO: Using java '/usr/local/software/jdk1.8.0_172/bin/java'
3月 16 14:49:28 localhost.localdomain activemq[792]: INFO: Starting - inspect logfiles specified in logging.properties and log4j.properties to get details
3月 16 14:49:28 localhost.localdomain activemq[792]: INFO: pidfile created : '/usr/local/software/apache-activemq-5.15.13/data/activemq.pid' (pid '870')
3月 16 14:49:28 localhost.localdomain systemd[1]: Started SYSV: activemq .....
```

systemctl stop activemq 关闭 Activemq 服务

systemctl start activemq 开启 Activemq 服务

systemctl restart activemq 重启 Activemq 服务

4.9. Mysql 服务相关命令

systemctl status mysql 查询 mysql 服务启动状态

```
[root@localhost center]# systemctl status mysql
● mysqld.service - LSB: start and stop MySQL
   Loaded: loaded (/etc/rc.d/init.d/mysqld; bad; vendor preset: disabled)
   Active: active (running) since 四 2023-03-16 14:49:39 CST; 1 weeks 6 days ago
     Docs: man:systemd-sysv-generator(8)
   Process: 1334 ExecStart=/etc/rc.d/init.d/mysqld start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/mysqld.service
           └─1385 /bin/sh /usr/local/mysql/bin/mysqld_safe --datadir=/usr/local/mysql/data --pid-file=/usr/local/mysql/data/localhost.localdomain.pid
             └─1650 /usr/local/mysql/bin/mysqld --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data --plugin-dir=/usr/local/mysql/lib/plugin --user=mysql --log-error=

3月 16 14:49:34 localhost.localdomain systemd[1]: Starting LSB: start and stop MySQL...
3月 16 14:49:39 localhost.localdomain mysqld[1334]: Starting MySQL.... SUCCESS!
3月 16 14:49:39 localhost.localdomain systemd[1]: Started LSB: start and stop MySQL.
[root@localhost center]#
```

systemctl stop mysql 关闭 mysql 服务

systemctl start mysql 开启 mysql 服务

systemctl restart mysql 重启 mysql 服务

4.10. Redis 服务相关命令

后台启动:

```
cd /usr/local/redis/
```

```
./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 # 0000000000000000 Redis is starting 0000000000000000
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@localhost ~]# ps -ef | grep redis
root      2236      1   0  3月20 ?        00:30:13 /usr/local/bin/redis-server 0.0.0.0:6379
root      8235    7956   0  20:09 pts/2    00:00:00 grep --color=auto redis
[root@localhost ~]#
```


4.11. nginx 服务相关命令

systemctl status nginx 查询 nginx 服务启动状态

```
[root@localhost ~]# systemctl status nginx
● nginx.service - nginx service
   Loaded: loaded (/etc/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since — 2023-03-20 15:00:01 CST; 1 weeks 2 days ago
   Process: 1561 ExecStart=/usr/local/nginx/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 1573 (nginx)
   CGroup: /system.slice/nginx.service
           └─1573 nginx: master process /usr/local/nginx/sbin/nginx
             └─1578 nginx: worker process

3月 20 15:00:01 localhost.localdomain systemd[1]: Starting nginx service...
3月 20 15:00:01 localhost.localdomain systemd[1]: Started nginx service.
```

systemctl stop nginx 关闭 nginx 服务

systemctl start nginx 开启 nginx 服务

systemctl restart nginx 重启 nginx 服务

4.12. nacos 服务相关命令

查看 nacos 启动: ps -ef | grep nacos

```
[root@localhost ~]# ps -ef | grep nacos
root      963      1  9 3月20 ?        20:39:48 java -jar -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-camera.jar --server.port=9501 --spring.cloud.nacos
ry.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root     1018      1  2 3月20 ?        05:21:32 java -Xms256M -Xmx512M -XX:PermSize=256M -XX:MaxPermSize=512M -jar youeran-gateway.jar --spring.cloud.nacos.discovery.server-
0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848 --server.port=8020
root     1045      1  8 3月20 ?        19:22:54 java -jar -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-log.jar --server.port=9581 --spring.cloud.nacos.d
server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root     1071      1  5 3月20 ?        12:48:37 java -jar -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-websocket.jar --server.port=9299 --spring.cloud.n
overy.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root     1132      1  8 3月20 ?        19:37:11 java -jar -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-mengin.jar --server.port=9999 --spring.cloud.nacos
ry.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root     1202      1  9 3月20 ?        20:08:21 java -jar -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-arithmetic.jar --server.port=9880 --spring.cloud.
covery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root     1493      1  5 3月27 ?        02:54:08 java -jar -Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=0.0.0.0:9608 -Xms512M -Xmx512M -XX:PermSize=512M -X
mSize=1024M youeran-stream-media-server.jar --server.port=8998 --spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:88
,server.id={892df55f-f481-4891-9390-9ea6f0e3ffcc}
root     1644      1  3 3月20 ?        07:20:42 /usr/local/software/jdk1.8.0_172/bin/java -Xms512m -Xmx512m -Xm256m -Dnacos.standalone=true -Djava.ext.dirs=/usr/local/softwa
8.0_172/jre/lib/ext:/usr/local/software/jdk1.8.0_172/lib/ext:/usr/local/software/nacos/plugins/cmdb:/usr/local/software/nacos/plugins/mysql -Xloggc:/usr/local/software/nacos
coe_gc.log -verbose:gc -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=100M -Dnacos.i
/local/software/nacos -Dloader.path=/usr/local/software/nacos/plugins/health -jar /usr/local/software/nacos/target/nacos-server.jar --spring.config.location=classpath:/clas
onfig/,file:/usr/local/software/nacos/conf/ --logging.config=/usr/local/software/nacos/conf/nacos-logback.xml --server.max-http-header-size=524288 na
root     9407      1  9 3月27 ?        04:29:11 java -jar -Xms256M -Xmx512M -XX:PermSize=256M -XX:MaxPermSize=512M youeran-alarm.jar --server.port=9801 --spring.cloud.nacos.d
server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
```

启动命令: sh /usr/local/software/nacos/bin/startup.sh -m standalone

4.13. 转码服务相关命令

pm2 list #查看进程列表

pm2 kill #结束进程

pm2 restart all #重启所有进程

pm2 logs #查看日志

启动转码服务:

cd /usr/local/software/flvnode

./service.sh start

4.14. Jar 包服务相关命令

查看 jar 包启动情况：ps -ef | grep youeran 以此来查看未启动起来的 jar 包。

```
[root@localhost ~]# ps -ef | grep youeran
root      963      1  9 3月20 ?        20:40:13 java -jar -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-camera.jar --server.port=9501 --spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root     1018      1  2 3月20 ?          05:21:39 java -Xms256M -Xmx512M -XX:PermSize=256M -XX:MaxPermSize=512M -jar youeran-gateway.jar --spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848 --server.port=8020
root     1045      1  8 3月20 ?          19:23:18 java -jar -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-log.jar --server.port=9581 --spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root     1071      1  5 3月20 ?          12:49:01 java -jar -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-websocket.jar --server.port=9299 --spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root     1132      1  8 3月20 ?          19:37:35 java -jar -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-menjin.jar --server.port=9999 --spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root     1202      1  9 3月20 ?          20:08:45 java -jar -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-arithmetic.jar --server.port=9800 --spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root     1493      1  5 3月27 ?          02:54:23 java -jar -Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=0.0.0.0:9608 -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-stream-media-server.jar --server.port=8998 --spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848 --msa.server-id={892df55f-f481-4891-9390-9ea6f0e3ffcc}
root     9407      1  9 3月27 ?          04:29:39 java -jar -Xms256M -Xmx512M -XX:PermSize=256M -XX:MaxPermSize=512M youeran-alarm.jar --server.port=9801 --spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root    13007      1  9 3月20 ?          19:58:31 java -jar -Xms256M -Xmx512M -XX:PermSize=256M -XX:MaxPermSize=512M youeran-report.jar --server.port=9700 --spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
root    19547      1 10 3月27 ?          05:03:25 java -jar -Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=0.0.0.0:9600 -Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=1024M youeran-wisdom.jar --server.port=9908 --spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848 --spring.cloud.nacos.config.server-addr=127.0.0.1:8848
```

一下以 camera 服务作为例子描述 jar 包相关命令：

1. 启动 camera 服务：
cd /usr/local/jar/camera
./service.sh start
2. 重启 camera 服务：
cd /usr/local/jar/camera
./service.sh restart
3. 停止 camera 服务：
cd /usr/local/jar/camera
./service.sh stop

4.15. 1.out 日志过大，设置日志滚动

使用 centos 自带 logrotate 管理日志。以 center 和 camera 为例，可根据实际情况进行日志管理。

cd /etc/logrotate.d/创建 1.out.log 文件，如下图所示：


```

/usr/local/jar/center/*.out
/usr/local/jar/camera/*.out {
    size 100K
    rotate 5
    missingok
    notifempty
    copytruncate
}

```

/usr/local/jar/center/*.out 为 center 记录日志的目录
 /usr/local/jar/camera/*.out 为 camera 记录日志的目录
 size 100K 为截断日志的大小，单位可以设置为 K 和 M
 rotate 5 为滚动保留日志的个数
 notifempty 当日志为空时，不进行轮转
 copytruncate 将当前日志备份并截断。

具体脚本如下：

```

/usr/local/jar/center/*.out
/usr/local/jar/camera/*.out {
    size 100K
    rotate 5
    missingok
    notifempty
    copytruncate
}

```

加入定时任务，每一分钟执行一次 `crontab -e`：

```
*/1 * * * * /usr/sbin/logrotate /etc/logrotate.d/1.out.log
```

```

[root@localhost center]# ll -h
总用量 132M
-rw-r--r-- 1 root root 44K 6月 19 10:02 1.out
-rw-r--r-- 1 root root 171K 6月 19 10:01 1.out.1
-rw-r--r-- 1 root root 217K 6月 19 10:00 1.out.2

```