

真趣开放平台运维篇：运维手册2.8.0

真趣开放平台Team

版本: V2.8.0

Updated at: 2021/09

(本文6205字，阅读约需21分钟)

1 概述

真趣开放平台以室内外定位技术为核心，是基于感知层硬件和自有通信协议研发的一款支持开发者接入的开放式位置物联网平台。

本文档是真趣开放平台的运维手册。

1.1 面向读者

本文档面向真趣开放平台研发人员、运维人员、测试人员、交付经理、运营专员及开放平台上游产品线的研发及测试人员。

本文档假定：

- 你对Linux有一定了解
- 你能基于安装文档熟练安装开放平台

1.2 文档说明

- 本文档**不是**开放平台的操作手册，**不是**开放平台的安装手册。
- 本文档内容适用于通过开放平台标准版安装包安装的环境，部分适用于定制化安装的分布式/集群化环境。
- 本文档内容适用于开放平台2.8.x；对于开放平台2.x其他版本，文档内容大部分适用。
- 针对开放平台运维过程中的常见问题，请参阅《真趣开放平台运维篇：经典15问》。
- 囿于篇幅与认知水平，错漏之处难免，如有反馈建议可发送至 wangnan@joysuch.com，内容会持续优化。

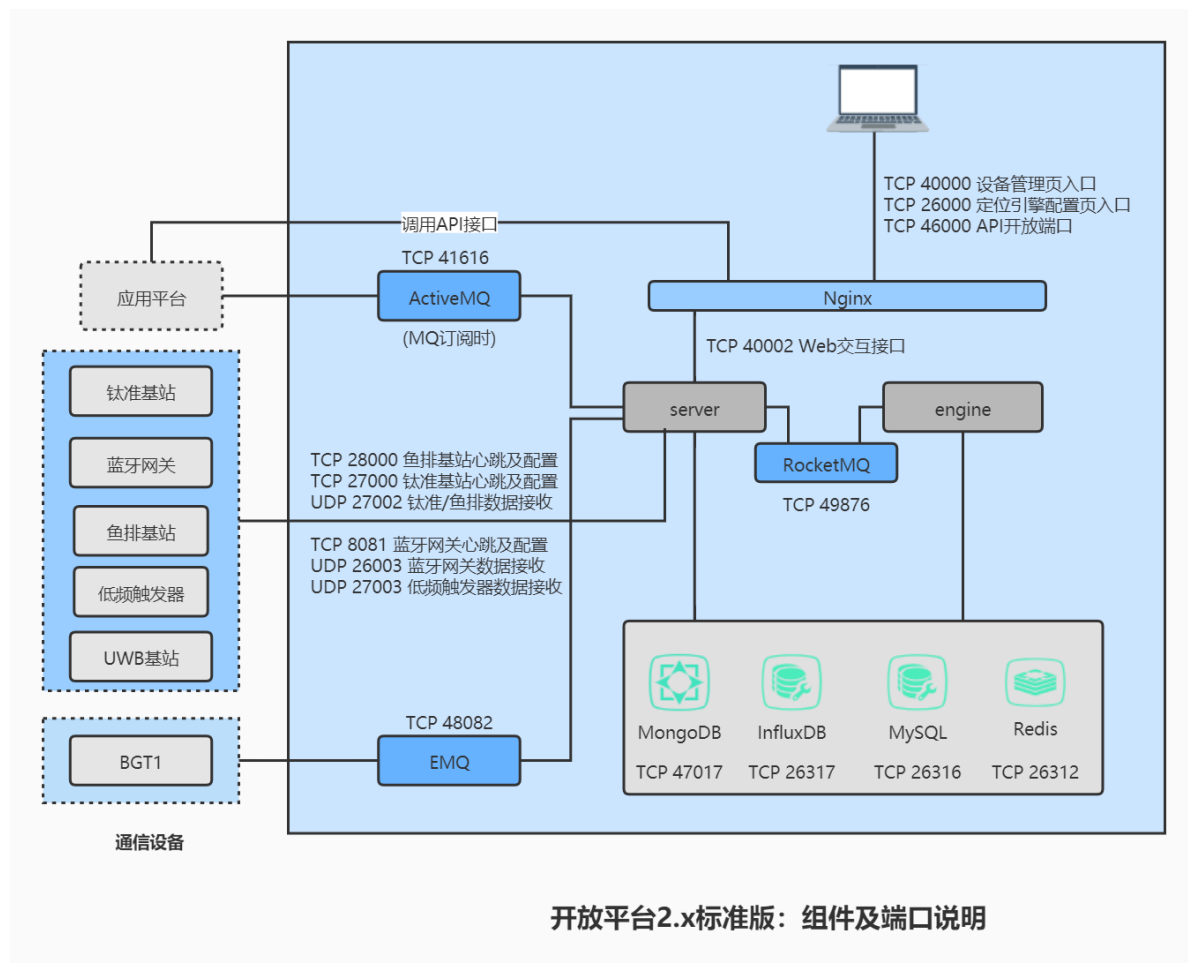
2 部署说明

2.1 真趣产品架构示意图



真趣产品架构分层示意图

2.2 开放平台部署图



开放平台2.x标准版：组件及端口说明

2.2.1 开放平台端口一览表

端口类型	端口号	端口说明
TCP	40000	平台设备管理Web页端口
TCP	26000	平台定位引擎Web页端口
TCP	46000	平台API端口
TCP	28000	鱼排定位系统基站心跳及配置端口
TCP	27000	钛准定位系统基站心跳及配置端口
UDP	27002	钛准定位系统，鱼排系统数据接收端口
TCP	8081	蓝牙定位系统网关心跳及配置端口
UDP	26003	蓝牙定位系统网关扫描数据接收端口
UDP	27003	蓝牙网关低频触发数据接收端口
UDP	27006	万能基站H3C模组数据接收端口
TCP	26316	MySQL数据库连接端口
TCP	47017	MongoDB数据库连接端口
TCP	26317	InfluxDB数据库连接端口
TCP	26312	Redis连接端口
TCP	41616	ActiveMQ broker端口
TCP	48161	ActiveMQ控制台端口
TCP	49876	RocketMQ name server端口
TCP	40911	RocketMQ broker端口
TCP	56789	RocketMQ控制台端口（使用时）
TCP	48082	EMQX MQTT连接端口
TCP	48083	EMQX控制台端口
TCP	40001	Node组件端口（仅SaaS环境）
TCP	40002	Server组件端口
TCP	41000	LoRaWANServer Web页端口（使用时）
TCP	44444	AoA定位引擎API端口（使用时）
UDP	55555	AoA定位引擎数据订阅端口（使用时）
TCP	8085	化工平台2.9 Web页端口（使用时）
TCP	8888	化工平台3.x Web页端口（使用时）
TCP	8089	化工平台3.x 服务端端口（使用时）
TCP	1935	化工平台3.x RTMP视频流（使用时）

端口类型	端口号	端口说明
TCP	9999	化工平台5.x Web页端口（使用时）
TCP	8090	化工平台5.x 服务端端口（使用时）
TCP	7777	化工平台5.x FLV视频流（使用时）

2.3 开放平台安装清单

2.3.1 基础软件

基础软件名	安装目录	版本	说明
JDK	/home/seekcy/softs/jdk1.8/	1.8.0	Java程序运行环境
Nginx	/home/seekcy/softs/nginx/	1.19.3	反向代理/负载均衡中间件
ActiveMQ	/home/seekcy/softs/activemq/	5.14.1	MQ中间件，用于系统间数据推送
RocketMQ	/home/seekcy/softs/rocketmq/	4.5.2	MQ中间件，用于系统内数据推送
EMQ	/home/seekcy/softs/emqx/	4.3.3	MQ中间件(since 2.6.0)，用于MQTT通信数据推送(BGT1等设备)
Redis	/home/seekcy/softs/bin/	3.0.5	缓存中间件，用于缓存围栏、订阅等数据
MySQL	/home/seekcy/database/mysql/	5.7.28	存储中间件，用于保存建筑、网络参数、数据模板、围栏/区域等基础数据
MongoDB	/home/seekcy/database/mongodb/	3.4.23	存储中间件，用于保存设备及实时状态数据
InfluxDB	/home/seekcy/database/influxdb/	1.7.9	存储中间件，用于保存扫描上报、设备心跳、定位轨迹等时序数据

2.3.2 应用软件

应用软 件名	安装目录	说明
40000页 面	/home/seekcy/js_open/webpage/	平台前端组件：设备管理Web页
26000页 面	/home/seekcy/js_open/webpage/locate/	平台前端组件：定位引擎Web页
location- server	/home/seekcy/js_open/location-server/	平台后端组件：设备一体化平台后端，用于设备接入，协议解析，数据订阅推送等
location- engine	/home/seekcy/js_open/location-engine/	平台后端组件：定位引擎，用于定位解算

3 运维指南

3.1 开机自启与watchdog

开放平台被正确安装时，安装脚本已配置平台相关基础软件及应用软件开机自启；并已配置监控脚本每3分钟检查程序运行状态，如程序不存在则尝试启动此程序。

可在 seekcy用户 下通过 crontab -l 命令查看配置情况，内容如下：

```
1 MAILTO=""
2 @reboot /home/seekcy/database/start.sh
3 @reboot /bin/sleep 30; /home/seekcy/softs/start.sh
4 @reboot /bin/sleep 60; /home/seekcy/js_open/start.sh
5 */3 * * * * /bin/sleep 600; /home/seekcy/database/watchdog.sh
6 */3 * * * * /bin/sleep 630; /home/seekcy/softs/watchdog.sh
7 */3 * * * * /bin/sleep 660; /home/seekcy/js_open/watchdog.sh
```

3.2 启动脚本说明

所在目录	脚本	作用
/home/seekcy/database/	start.sh	启动MySQL、MongoDB、InfluxDB
/home/seekcy/database/	watchdog.sh	看门狗脚本(监控MySQL、MongoDB、InfluxDB)，定时任务调用
/home/seekcy/softs/	start.sh	启动Nginx、Redis、ActiveMQ、RocketMQ、EMQ
/home/seekcy/softs/	watchdog.sh	看门狗脚本(监控Nginx、Redis、ActiveMQ、RocketMQ、EMQ)，定时任务调用
/home/seekcy/js_open/	start.sh	启动location-server、location-engine
/home/seekcy/js_open/	watchdog.sh	看门狗脚本(监控location-server、location-engine)，定时任务调用
/home/seekcy/js_open/location-server/	service-server.sh	启动/关闭/重启location-server
/home/seekcy/js_open/location-engine/	service-engine.sh	启动/关闭/重启location-engine

3.3 中间件运维

注意：本章节运维操作命令均在seekcy用户下运行。

3.3.1 Nginx

3.3.1.1 Nginx常用命令

```

1  # 切换到Nginx主目录
2  cd /home/seekcy/softs/nginx
3  # 启动Nginx
4  ./sbin/nginx -c conf/nginx.conf
5  # 修改配置后测试配置
6  ./sbin/nginx -t
7  # 修改配置后重载配置
8  ./sbin/nginx -s reload
9  # 快速关闭(不管有没有正在处理的请求都关闭)
10 ./sbin/nginx -s stop
11 # 优雅关闭(在退出前完成已经接受的连接请求)
12 ./sbin/nginx -s quit

```

3.3.1.2 Nginx运维备忘

3.3.1.2.1 超时时间配置

Nginx对上传文件大小有限制，默认是1M。若上传文件很大，要修改文件大小限制，并适当调整上传超时时间。

相关配置项在配置文件nginx.conf中修改，修改示例如下：

```
1 client_max_body_size      50m; # 限制请求体的大小，若超过所设定的大小，返回413错误，默认1m
2 client_header_timeout    1m; # 读取请求头的超时时间，若超过所设定的大小，返回408错误
3 client_body_timeout      1m; # 读取请求实体的超时时间，若超过所设定的大小，返回413错误
4 proxy_connect_timeout    60s; # http请求无法立即被容器(tomcat, netty等)处理，被放在
    nginx的待处理池中等待被处理。此参数为等待的最长时间，默认为60秒，官方推荐最长不要超过75秒
5 proxy_read_timeout       1m; # http请求被容器(tomcat, netty等)处理后，nginx会等待处理结
    果，也就是容器返回的response。此参数即为服务器响应时间，默认60秒
6 proxy_send_timeout       1m; # http请求被服务器处理完后，把数据传返回给Nginx的用时，默认60秒
```

3.3.1.2.2 某项目UDP外网映射issue

【问题描述】开放平台通过一个Nginx节点做反向代理(外网映射)，设备通过外网映射地址和端口连接服务。运行一段时间后设备离线，重启Nginx后设备在线，如此反复。

【根本原因】Nginx中，物联网模组通信UDP反向代理配置项proxy_timeout（连接保持时间，超过此时这个通信通道及端口就会被释放掉）被设值为24h，持续不断的数据上报未释放socket连接，大量占用file descriptor，超过内核配置后触发系统错误：11: Resource temporarily unavailable，致使后续UDP连接请求无法处理，同时导致服务器中所有涉及file descriptor的命令均无法执行。

【解决方案】物联网模组通信UDP反向代理连接保持时间由24h改为2m，修改后实测UDP活跃连接数保持在200左右(此项目当前7个模组、18个标签，平均每分钟UDP连接约100个)

【更改文件】Nginx配置文件nginx.conf

3.3.1.2.3 集群部署时负载均衡配置

集群部署时，HTTP负载均衡配置片段示例如下(在http{}块中配置)：

```
1 upstream locationServer {
2     server location-server.1:40002 weight=1;
3     server location-server.2:40002 weight=1;
4     server location-server.3:40002 weight=1;
5     server location-server.4:40002 weight=1;
6 }
7
8 location ^~ /heart-beat {
9     proxy_pass http://locationServer;
10 }
```

TCP负载均衡配置片段示例如下(在stream{}块中配置)：

```
1 upstream blg27000 {
2     hash $remote_addr consistent;
3     server location-server.1:27000 weight=1;
4     server location-server.2:27000 weight=1;
5     server location-server.3:27000 weight=1;
6     server location-server.4:27000 weight=1;
7 }
8
9 server {
10     listen 27000;
11     #listen 27000 so_keepalive=on; #so_keepalive=on表示启用长连接
12     proxy_pass blg27000;
13     proxy_connect_timeout 30s;
14     proxy_timeout 600s;
15 }
```

3.3.1.2.4 WebSocket配置

WebSocket配置片段示例如下(在server{}块中配置):

```
1 location ^~ /websocket {
2     proxy_pass http://127.0.0.1:40002;
3     proxy_http_version 1.1;
4     proxy_set_header Upgrade $http_upgrade;
5     proxy_set_header Connection "Upgrade";
6     proxy_set_header X-Real-IP $remote_addr;
7 }
```

3.3.1.2.5 HTTPS配置

HTTPS配置片段示例如下(在http{}块中配置):

```
1 server {
2     listen 443 ssl;
3     server_name 10.10.10.10;
4     ssl on;
5     ssl_certificate /etc/pki/CA/nginx/server.crt;
6     ssl_certificate_key /etc/pki/CA/nginx/server.key;
7     ssl_session_timeout 5m;
8
9     location ^~ /test {
10        proxy_pass http://10.10.10.10:8080/test;
11        proxy_next_upstream error timeout invalid_header http_500 http_502
12        http_503;
13        proxy_set_header Host $host;
14        proxy_set_header X-Forwarded-For $remote_addr;
15        proxy_set_header X-Forwarded-Proto https;
16    }
```

3.3.2 ActiveMQ

3.3.2.1 ActiveMQ启动/关闭

```
1 # 切换到ActiveMQ的bin目录
2 cd /home/seekcy/softs/activemq/activemq-01/bin
3 #停止
4 ./activemq stop
5 #启动
6 ./activemq start > /dev/null
```

3.3.2.2 ActiveMQ控制台

控制台url: <http://服务器ip:48161/> (登录用户/密码: admin/*****)

文中密码信息用 ***** 代替, 如需要请与研发确认, 下同。

导航菜单中:

- Queues 队列方式消息
- Topics 主题方式消息
- Subscribers 消息订阅者
- Connections 客户端连接

- Network 集群环境中的Network Bridge
- Send 手动发送消息入口

控制台访问端口配置入口：

/home/seekcy/softs/activemq/activemq-01/conf/jetty.xml

控制台登录账号、密码配置入口：

/home/seekcy/softs/activemq/activemq-01/conf/jetty-realm.properties

3.3.2.3 鉴权配置

开放平台使用自定义插件 jmsplug-0.0.1.jar对ActiveMQ连接请求进行鉴权。

鉴权插件说明如下：

- 插件部署目录：/home/seekcy/softs/activemq/activemq-01/lib/
- 鉴权机制：使用open账号及用户license认证连接请求的合法性
- 鉴权插件在/home/seekcy/softs/activemq/activemq-01/conf/activemq.xml中配置
- 鉴权插件DB连接配置：/home/seekcy/softs/activemq/activemq-01/conf/db.properties

分布式部署时如ActiveMQ和数据库部署在不同节点上，需修改db连接参数

3.3.2.4 ActiveMQ运维备忘

3.3.2.4.1 hostname

主机的hostname中包含下划线“_”时，会导致ActiveMQ服务启动失败。此时通过 `hostnamectl` 命令将hostname修改为合规的字符串即可，命令示例如下（使用root用户操作）：

```
1 hostnamectl set-hostname 新主机名
```

3.3.2.4.2 服务器时间

开放平台通过ActiveMQ向应用平台推送订阅数据时，数据在ActiveMQ中默认存储30s。ActiveMQ所在机器与应用平台所在机器的时间应确保同步，否则可能导致应用平台收不到消息。可通过 `date` 命令确认服务器时间，如已确认服务器时间有误，可参考如下方式修改时间(使用root用户操作)：

(1) 当前服务器可以访问外网时，通过ntp同步时间：

```
1 # 安装ntpd
2 yum -y install ntpdate
3 # 同步时间
4 ntpdate -u pool.ntp.org
```

(2) 当前服务器不能访问外网时，手工修改时间：

```
1 # 修改为实际时间
2 date -s "2020-01-01 00:00:00"
3 # 写入时钟
4 clock -w
```

时间修改后，可再次使用date命令验证时间是否正确。

3.3.2.4.3 JVM参数调优

可在文件 `/home/seekcy/softs/activemq/activemq-01/bin/env` 中调整JVM参数，常见的修改包括：

(1) 根据消息负载情况适当调整JVM堆内存配置

(2) 使用G1GC以降低GC频率

使用默认GC，线上环境观察发现YGC 2-3s一次，FGC每天260多次；
使用G1GC后，线上环境实测一个月，发现YGC频率大大降低、FGC消失

修改示例：

```
1  ACTIVEMQ_OPTS_MEMORY="-Xms4G -Xmx4G -XX:+UseG1GC"
```

3.3.2.4.4 队列调优

可调整配置参数，定期清空不活跃的queue/topic，以节约服务器资源。

“不活跃”，指没有pending消息且所配的时间周期内没有消费者连接。

修改方式如下：

在文件 `/home/seekcy/softs/activemq/activemq-01/conf/activemq.xml` 中加入如下配置：

```
1  <broker xmlns="http://activemq.apache.org/schema/core" brokerName="localhost"
   dataDirectory="${activemq.data}" schedulePeriodForDestinationPurge="60000">
2  ....
3  <policyEntry queue=">" gcInactiveDestinations="true"
   inactiveTimeoutBeforeGC="60000"/>
```

参数说明如下(参数中的时间单位为ms)：

(1) `schedulePeriodForDestinationPurge="60000"`，指broker每60s检查不活跃的queue/topic

(2) `inactiveTimeoutBeforeGC="60000"`，指清除超过60s的不活跃queue/topic

(3) `gcInactiveDestinations="true"`，指清除不活跃queue/topic功能开启

3.3.3 RocketMQ

3.3.3.1 RocketMQ常用命令

```
1  # 切换到RocketMQ的bin目录
2  cd /home/seekcy/softs/rocketmq/bin
3  # 启动namesrv
4  nohup sh mqnamesrv -c /home/seekcy/softs/rocketmq/conf/namesrv.properties &
5  # 启动broker
6  nohup sh mqbroker -c /home/seekcy/softs/rocketmq/conf/broker.conf &
7  # 关闭broker服务
8  ./mqshutdown broker
9  # 关闭namesrv服务
10 ./mqshutdown namesrv
11
12 # 删除Topic
13 ./mqadmin deleteTopic -n localhost:49876 -c DefaultCluster -t openV2--engine-
   initiative-location-result
14 # 创建Topic
```

```

15  ./mqadmin updateTopic -n localhost:49876 -c DefaultCluster -t openV2--engine-
    initiative-location-result
16  # 查看所有topic
17  ./mqadmin topicList -n localhost:49876
18  # 查看所有topic的吞吐TPS统计
19  ./mqadmin statsAll -n localhost:49876
20  # 查看特定topic的吞吐TPS统计
21  ./mqadmin statsAll -t openV2--engine-location-result -n localhost:49876
22  # 查看定位结果数据(topic: openV2--engine-location-result)积压情况
23  ./mqadmin consumerProgress -g 'open-location-result-v1-lister' -n localhost:49876
    -s true | grep 'openV2--engine-location-result' | grep -v 'RETRY' | sed 's/ \+/
    /g' | cut -d ' ' -f 7 | sort -r | head -n 1
24
25  # 查看mqadmin支持哪些命令
26  ./mqadmin
27  # 查看具体命令的使用
28  ./mqadmin help 命令名称

```

建议启动时先启动namesrv再启动broker；关闭时先关闭broker再关闭namesrv。

3.3.3.2 RocketMQ控制台

RocketMQ Console是RocketMQ项目的扩展插件，是一个图形化管理控制台，提供Broker集群状态查看，Topic管理，Producer、Consumer状态展示，消息查询等常用功能，这个功能在安装好RocketMQ后需要额外单独安装、运行。标准版安装包暂未提供RocketMQ控制台（rocketmq-console），如有需要可联系研发部署。

控制台url: <http://服务器ip:56789/>

3.3.3.3 RocketMQ运维备忘

3.3.3.3.1 JVM参数调优

可根据消息负载情况，在文件 `/home/seekcy/softs/rocketmq/bin/runbroker.sh` 中调整JVM堆内存配置，示例如下：

```

1  JAVA_OPT="{JAVA_OPT} -server -Xms8G -Xmx8G -Xmn4G"

```

3.3.3.3.2 RocketMQ消息体大小设置

以将支持的消息体大小最大为8M为例，配置如下：

(1) RocketMQ broker配置

在文件 `/home/seekcy/softs/rocketmq/conf/broker.conf` 中设置maxMessageSize：

```

1  maxMessageSize=8388608

```

(2) Spring Boot应用配置

在应用的配置文件 `application.properties` 中设置生产者/消费者的maxMessageSize：

```

1  rocketmq.producer.maxMessageSize=8388608
2  rocketmq.consumer.maxMessageSize=8388608

```

3.3.3.3.3 TPS调优实践

暂无

3.3.3.3.4 RocketMQ日志

说明	目录	文件
中间件日志	/home/seekcy/softs/rocketmq/logs/	namesrv.log broker.log stats.log store.log ...
location-server与RocketMQ交互日志	/home/seekcy/js_open/location-server/logs/	rocketmq_client.log rocketmq_client.log.1 ...
location-engine与RocketMQ交互日志	/home/seekcy/logs/rocketmqlogs/	rocketmq_client.log rocketmq_client.log.1 ...

3.3.4 EMQ

3.3.4.1 EMQ启动/关闭

```
1  # 切换到EMQ目录
2  cd /home/seekcy/softs/emqx/bin
3  # 启动EMQ
4  ./emqx start
5  # 关闭EMQ
6  ./emqx stop
7  # 重启EMQ
8  ./emqx restart
```

3.3.4.2 EMQ控制台

可通过 Web 控制台查看服务器运行状态、统计数据、客户端、会话、主题、订阅、插件信息等。

控制台url: <http://服务器ip:48083/> (登录用户/密码: admin/ *****)

控制台登录账号、密码及访问端口配置入口:

/home/seekcy/softs/emqx/etc/plugins/emqx_dashboard.conf

3.3.5 Redis

3.3.5.1 Redis启动/关闭

```
1  # 切换到Redis目录
2  cd /home/seekcy/softs/bin
3  # 启动Redis server
4  redis-server ../redis_26312.conf > /dev/null 2>&1 &
5  # 关闭Redis server
6  ps aux | grep 'redis' | grep -v grep | awk '{print $2}' | xargs kill -9
```

3.3.5.2 Redis命令行界面

输入如下命令进入Redis命令行界面：

```
1 redis-cli -p 26312 -a *****
```

在命令行界面输入 exit 退出；输入 shutdown 可关闭 redis server。

3.3.5.3 Redis常用命令

```
1 # 查看钦准基站当前下行topic
2 hgetall tran_blg
3 # 删除指定key
4 del BLT_ON_OFFLINE_1918FF00D8B1
```

shell界面批量删除数据命令示例：

```
1 ./redis-cli -p 26312 -a ***** keys "region_stat_*" |xargs ./redis-cli -p 26312 -a
***** del
```

3.3.6 MySQL

3.3.6.1 MySQL启动/关闭

```
1 # 切换到MySQL目录
2 cd home/seekcy/database/mysql/support-files
3 # 启动MySQL
4 ./mysql.server start
5 # 关闭MySQL
6 ./mysql.server stop
7 # 重启MySQL
8 ./mysql.server restart
```

3.3.6.2 MySQL命令行界面

输入如下命令进入MySQL命令行界面（此处显式指定sock文件）：

```
1 mysql -u root -p --socket=/home/seekcy/database/mysql/mysql.sock
```

按提示输入密码后进入命令行界面；

在命令行界面输入 exit 退出。

3.3.6.3 MySQL常用语句

```
1 -- 允许远程访问
2 GRANT ALL PRIVILEGES ON *.* TO root@'%' IDENTIFIED BY 'Password2017';
3 FLUSH PRIVILEGES;
4
5 -- 切换到js_open_location_server库
6 use js_open_location_server;
7
8 -- 查看HTTP订阅数据
9 select * from subscribe;
10
11 -- 查看系统配置
12 select * from sys_config;
13
```

```

14  -- 查看围栏数据
15  select * from rail;
16
17  -- 查看围栏点位数据
18  select * from rail_point;
19
20  -- 显示当前连接数
21  show status like 'Threads_connected';
22
23  -- 显示当前正在执行的连接
24  show processlist;

```

3.3.6.4 MySQL运维备忘

3.3.6.4.1 启动MySQL报错: the server quit without updating pid file

启动MySQL时报错: the server quit without updating pid file

问题排查: 查看错误日志, 发现信息: another process with pid xxx is using unix socket file

解决方法: 删除或重命名文件 mysql.sock.lock 后, 再次启动正常

3.3.6.4.2 重启MySQL报错: World-writable config file '... my.cnf' is ignored

重启MySQL时提示, Warning: World-writable config file '/home/seekcy/database/mysql/my.cnf' is ignored

问题排查: 配置文件my.cnf操作权限为777, 此时表示任何一个用户都可以操作my.cnf, MySQL认为存在安全隐患, 故忽略此配置文件。

解决方法: 执行如下命令将配置文件my.cnf操作权限调整为644, 修改后再重启MySQL即可。

```

1  chmod 644 /home/seekcy/database/mysql/my.cnf

```

3.3.6.4.3 切换数据库时报错: get a quicker startup with -A

切换数据库时提示:

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

问题排查: 打开数据库即 use dbname 时, 要预读数据库信息, 当数据库比较庞大的时候, 系统会出现卡顿问题

解决方法: 连接数据库时使用 `-A` 参数, 就不预读数据库信息

```

1  mysql -u root -p -A

```

3.3.7 MongoDB

3.3.7.1 MongoDB启动/关闭

```

1  # 切换到MongoDB目录
2  cd /home/seekcy/database/mongodb/bin
3  # 启动MongoDB
4  ./mongod -f ../mongodb.conf
5  # 关闭MongoDB
6  ./mongod --dbpath=/home/seekcy/database/mongodb/data/db --shutdown

```

3.3.7.2 MongoDB命令行界面

输入如下命令进入MongoDB命令行界面（指定用户、密码、数据库）：

```
1 mongo mongodb://jsopen:*****@localhost:47017/js_open_location_server
```

在命令行界面输入 exit 退出。

3.3.7.3 MongoDB常用语句

```
1  查询终端总数量
2  db.blt.find({}).count()
3
4  查询各建筑下终端总数量(按总数量倒序排序)
5  db.blt.aggregate([
6      {"$group" : {_id:"$buildId", count:{$sum:1}}},
7      {$sort: {"count":-1}}
8  ])
9
10 按终端硬件版本/固件版本统计计数(按总数量倒序排序)
11 db.blt.aggregate([
12     {$group: { _id: { hardwareVersion: "$hardwareVersion", firmwareVersion:
13         "$firmwareVersion" }, count:{$sum:1} } },
14     {$sort: {"count":-1}}
15 ])
16 查看指定建筑下（以建筑id为123456为例）所有beacon的mac, major, minor信息
17 db.ibeacon.find({"buildId":"123456"}, {_id:0, mac:1, major:1, minor:1})
18
19 查看指定建筑下（以建筑id为123456为例）最后通讯时间在2021年1月1日之前(默认是UTC时间)的终端计数
20 db.getCollection("blt").find({"buildId":"123456", "lastCommunicationTime":
21     {$lt:new Date("2021-01-01T00:00:00Z")}}).count();
22
23 查看指定建筑下（以建筑id为123456为例）通讯时间为空的终端计数
24 db.getCollection("blt").find({"buildId":"123456", "lastCommunicationTime":
25     {$eq:null}}).count();
26
27 查看指定建筑下（以建筑id为123456为例）通讯时间为空的终端mac字段
28 db.getCollection("blt").find({"buildId":"123456", "lastCommunicationTime":
29     {$eq:null}}, {_id:0, mac:1});
30
31 数据更新：电量为1的beacon，电量更新为100（需指定multi:true，否则只更新符合条件数据中的1条）
32 db.ibeacon.update({'battery':1},{ $set: {'battery':100}}, {multi:true})
33
34 数据删除：删除指定建筑下（以建筑id为123456为例）的ibeacon
35 db.ibeacon.remove({'buildId':"123456"})
36
37 数据删除：删除指定建筑下（以建筑id为123456为例）的ibeacon(只删除1条)
38 db.ibeacon.remove({'buildId':"123456"}, 1)
```

3.3.8 InfluxDB

3.3.8.1 InfluxDB启动/关闭

```

1  # 切换到InfluxDB目录
2  cd /home/seekcy/database/influxdb/bin
3  # 启动InfluxDB
4  nohup ./influxd -config ../influxdb.conf &
5  # 关闭InfluxDB
6  ps aux | grep 'influx' | grep -v grep | awk '{print $2}' | xargs kill -9

```

3.3.8.2 InfluxDB命令行界面

输入如下命令进入InfluxDB命令行界面：

```

1  influx -username admin -password ***** -database 'js_open_location' -host
    '127.0.0.1' -port '26317'

```

在命令行界面输入 exit 退出。

3.3.8.3 InfluxDB常用语句

```

1  切换到 js_open_location 数据库
2  use js_open_location;
3
4  显示当前数据库中所有的表
5  show measurements;
6
7  查看已有的保留策略
8  show retention policies;
9
10 查看指定库的保留策略
11 show retention policies on js_open_location;
12
13 给test库新建保留策略并设为默认
14 create retention policy "2_hours" on "test" duration 2h replication 1 default;
15
16 查询指定终端指定时间段内的围栏/区域进出信息
17 SELECT * FROM "rail_region_inout_history" WHERE bltMac = '111111111111' and time
    >= '2021-06-10T00:00:00Z' and time < '2021-06-10T00:30:00Z' limit 100
    TZ('Asia/Shanghai')
18
19 查询指定终端(MAC前4位需替换为0000)指定时间段内的上报数据
20 SELECT * FROM "blt_beacon_influ"
21 WHERE time >= '2021-02-03T01:00:00Z' and time < '2021-02-03T02:00:00Z'
22 and buildId = '123456' and bltMac = '000011111111' TZ('Asia/Shanghai')
23
24 查询指定终端指定时间段内的定位数据
25 SELECT * FROM "js_datacenter_location"
26 WHERE time >= '2021-02-03T01:00:00Z' and time < '2021-02-03T02:00:00Z'
27 and buildId = '123456' and userId = '111111111111' TZ('Asia/Shanghai')
28
29 统计计数(当前不存在field或tag名为count时方可用)
30 SELECT SUM(count) FROM (SELECT *,count::INTEGER FROM MyMeasurement GROUP BY count
    FILL(1))
31
32 统计计数(根据field_key)
33 SELECT count(floorId) FROM "js_datacenter_location"
34
35 删除数据(性能上会有影响,一般不建议使用,而是用RP自动删除旧数据)
36 delete from rail_region_inout_history where time < now()-30d
37

```

```
38 统计指定库series数量
39 select * from _internal.."database" where "database"='js_open_location' order by
    time desc limit 1;
40
41 从单个measurement删除所有series
42 DROP SERIES FROM "h2o_feet"
43
44 从单个measurement删除指定tag的series
45 DROP SERIES FROM "h2o_feet" WHERE "location" = 'santa_monica'
46
47 从数据库删除有指定tag的所有measurement中的所有数据
48 DROP SERIES WHERE "location" = 'santa_monica'
```

3.3.8.4 InfluxDB运维备忘

3.3.8.4.1 max-series-per-database 限制

【问题描述】应用程序往InfluxDB中写入数据时报错：org.influxdb.InfluxDBException: partial write: max-series-per-database limit exceeded: (1000000) dropped=1

【根本原因】InfluxDB中，默认每一个数据库中的 series 的数量不能超过 100 万。限制的原因在于，Influxdb 为每个 series 建立了索引并且常驻内存，如果过大则可能将内存耗尽。一旦达到上限，再写入新的 series 则会报错。

series 的定义：使用同一个 measurement，retention policy，tag set 组合的points的集合

【解决方案】可临时将InfluxDB配置项max-series-per-database设置为0（设置为 0 表示没有限制），更改后重启InfluxDB。

建议优化measurement的tag key定义，避免出现大量的series。

【更改文件】InfluxDB配置文件influxdb.conf

3.4 应用运维

3.4.1 location-server

3.4.1.1 location-server启动/关闭/重启

```
1  # 切换到location-server目录
2  cd /home/seekcy/js_open/location-server
3  # 重启location-server(方式一)
4  ./service-server.sh
5  # 重启location-server(方式二)
6  ./service-server.sh restart
7  # 启动location-server
8  ./service-server.sh start
9  # 关闭location-server
10 ./service-server.sh stop
```

3.4.1.2 关键日志

3.4.1.2.1 基站日志

通过如下命令在实时日志中查看指定基站（假定MAC为111111111111）日志：

```
1 tailf /home/seekcy/js_open/location-server/logs/info.log | grep '111111111111'
```

如需查看历史日志，则在/home/seekcy/js_open/location-server/logs/目录下先找到某时间段下对应的log文件，再通过grep命令查找，实例如下：

```
1 # 从info.log中查看指定基站（假定MAC为111111111111）日志
2 grep '111111111111' info.log
3 # 从info-2021-09-01_001.log和info.log中查找指定基站（假定MAC为111111111111）日志，结果写入1.log
4 grep '111111111111' info-2021-09-01_001.log info.log > 1.log
```

后续章节不再赘述查看历史日志的方式。

3.4.1.2.2 订阅日志

修改配置文件/home/seekcy/js_open/location-server/config/logback-spring.xml,

(1) 开启HTTP订阅日志

```
1 <logger name="com.joysuch.location.server.service.location.subscribe.handler"
  level="debug" />
```

通过如下命令在实时日志中查看订阅推送信息：

```
1 # 查看订阅推送日志
2 tailf /home/seekcy/js_open/location-server/logs/info.log | grep 'send subscribe
  complete'
3
4 # 查看指定终端（假定MAC为222222222222）订阅推送
5 # 推送成功时
6 tailf /home/seekcy/js_open/location-server/logs/info.log | grep '222222222222' |
  grep 'send subscribe complete'
7 # 推送失败时(回调订阅接口时出现异常，往往是因为回调url不通)
8 tailf /home/seekcy/js_open/location-server/logs/info.log | grep '222222222222' |
  grep '推送失败'
```

(2) 开启MQ订阅日志

```
1 <logger name="com.joysuch.location.server.service.helper.SubScribeMqHelper"
  level="debug" />
```

通过如下命令在实时日志中查看订阅推送信息：

```
1 # 查看订阅推送
2 tailf /home/seekcy/js_open/location-server/logs/info.log | grep 'send mq content'
3 # 查看指定终端（假定MAC为222222222222）订阅推送
4 tailf /home/seekcy/js_open/location-server/logs/info.log | grep 'send mq content'
  | grep '222222222222'
```

3.4.2 location-engine

3.4.2.1 location-engine启动/关闭/重启

```
1  # 切换到location-engine目录
2  cd /home/seekcy/js_open/location-engine
3  # 重启location-engine(方式一)
4  ./service-engine.sh
5  # 重启location-engine(方式二)
6  ./service-engine.sh restart
7  # 启动location-engine
8  ./service-engine.sh start
9  # 关闭location-engine
10 ./service-engine.sh stop
```

3.4.2.2 关键日志

3.4.2.2.1 定位日志

修改配置文件/home/seekcy/js_open/location-engine/config/logback-spring.xml,

开启engine定位解算debug日志

```
1  <logger name="com.joysuch.location.server.engine.inner" level="debug"/>
```

通过如下命令在实时日志中查看指定终端（假定MAC为222222222222）定位日志：

```
1  tailf /home/seekcy/js_open/location-engine/logs/info.log | grep 'doLocate' | grep
'222222222222'
```