目录

[初始基础环境搭建： 1](#_Toc170378552)

[Docker搭建方式： 1](#_Toc170378553)

[支持远程mysql： 2](#_Toc170378554)

[支持远程使用密码的Redis： 8](#_Toc170378555)

[总体流程： 10](#_Toc170378556)

### 初始基础环境搭建：

两台服务器分别命名为master1和master2，在master1服务器上用docker的方式搭建open-falcon的基础环境，作为server部分；在master2服务器上用docker的方式搭建agent组件环境，作为agent部分。

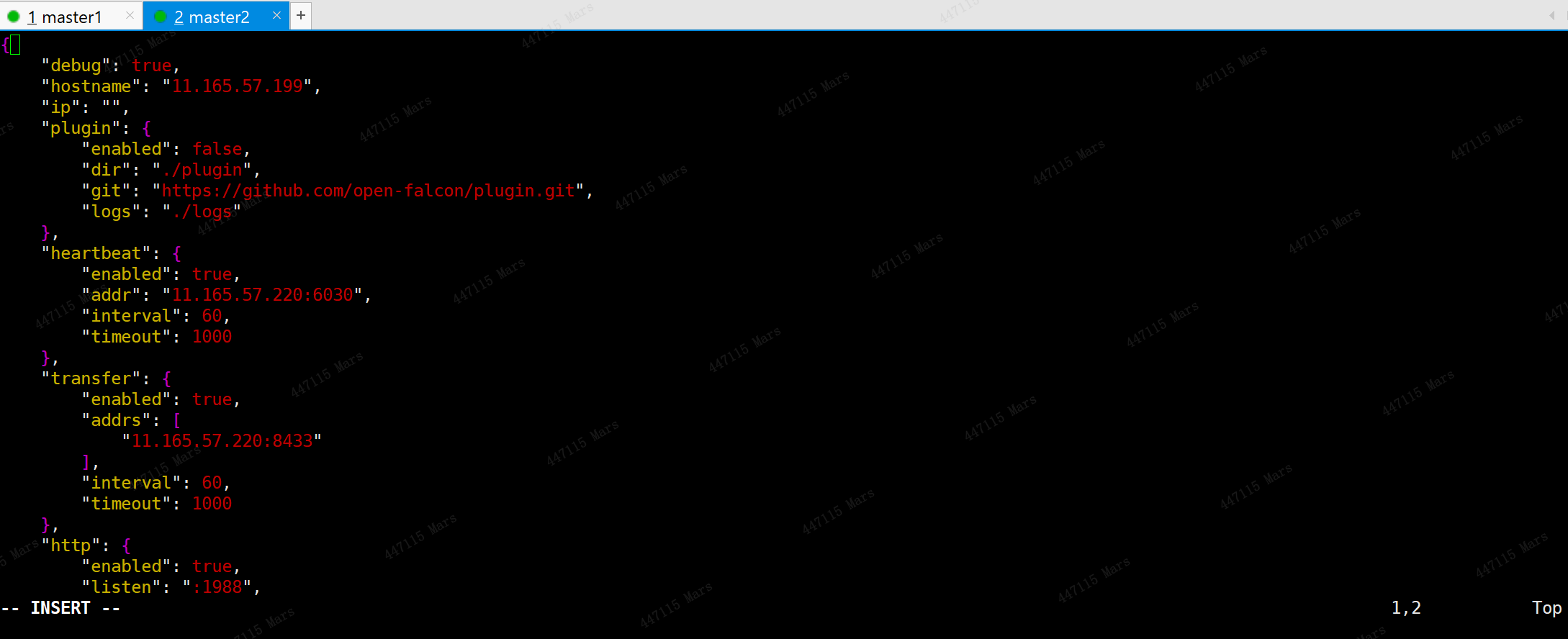
agent部分搭建好后会遇到ss工具无法正常使用的情况，需要配置agent环境，参考：

<https://aliyuque.antfin.com/alinpu_engineering/og0bia/edhumgxu0sshiv7q>

agent部分搭建后需要修改配置文件使得指向server端：

vim /open-falcon/agent/config/cfg.json

修改其中的hostname以及heartbeat和transfer中的addr部分，修改后如下：



### Docker搭建方式：

[**https://code.alibaba-inc.com/AliNPU/falcon-plus/blob/master/docker/README.md**](https://code.alibaba-inc.com/AliNPU/falcon-plus/blob/master/docker/README.md)

搭建完成后master1服务器上open-falcon连接的MySQL数据库为以docker方式创建的数据库，账号：root，密码123456。连接的Redis数据库为以docker方式创建的redis，以无密码方式登录。

### 支持远程mysql：

在环境搭建好的情况下将连接的数据库由本地数据库修改为远程数据库。

远程数据库地址和端口号rm-0jloj025h09b4jm03.mysql.rds.aliyuncs.com:3306

账号：ptgaiswai05:IBai21#v9P 密码：IBai21#v9P

**方法一：简单（推荐使用）**

新建容器，由于在官方文档中在创建容器时会直接绑定docker版本的mysql，导致在方法二中修改完open-falcon的配置文件后还需要修改容器的配置，非常麻烦。故直接修改创建容器的方法将远程MySQL配置到docker的open-falcon中。

安装**open-falcon的server部分**

**docker run -itd --name falcon-plus \**

**--link=falcon-redis:redis.falcon \**

**-p 8433:8433 \**

**-p 8080:8080 \**

**-p 18433:18433 \**

**-p 6030:6030 \**

**-e MYSQL\_PORT=ptgaiswai05:IBai21#v9P@tcp\(rm-0jloj025h09b4jm03.mysql.rds.aliyuncs.com:3306\) \**

**-e REDIS\_PORT=redis.falcon:6379 \**

**-v /home/work/open-falcon/data:/open-falcon/data \**

**-v /home/work/open-falcon/logs:/open-falcon/logs \**

**-v /etc/localtime:/etc/localtime \**

**openfalcon/falcon-plus:v0.3**

**安装open-falcon的dashboard部分**

**docker run -itd --name falcon-dashboard \**

**-p 8081:8081 \**

**--link=falcon-plus:api.falcon \**

**-e API\_ADDR=http://api.falcon:8080/api/v1 \**

**-e PORTAL\_DB\_HOST=rm-0jloj025h09b4jm03.mysql.rds.aliyuncs.com \**

**-e PORTAL\_DB\_PORT=3306 \**

**-e PORTAL\_DB\_USER=ptgaiswai05 \**

**-e PORTAL\_DB\_PASS=IBai21#v9P \**

**-e PORTAL\_DB\_NAME=falcon\_portal \**

**-e ALARM\_DB\_HOST=rm-0jloj025h09b4jm03.mysql.rds.aliyuncs.com \**

**-e ALARM\_DB\_PORT=3306 \**

**-e ALARM\_DB\_USER=ptgaiswai05 \**

**-e ALARM\_DB\_PASS=IBai21#v9P \**

**-e ALARM\_DB\_NAME=alarms \**

**-v /etc/localtime:/etc/localtime \**

**-w /open-falcon/dashboard openfalcon/falcon-dashboard:v0.2.1 \**

**'./control startfg'**

**安装open-falcon的agent部分**

**sudo docker run -d --restart always --name falcon-agent \**

**-e NUX\_ROOTFS=/rootfs \**

**-v /:/rootfs:ro \**

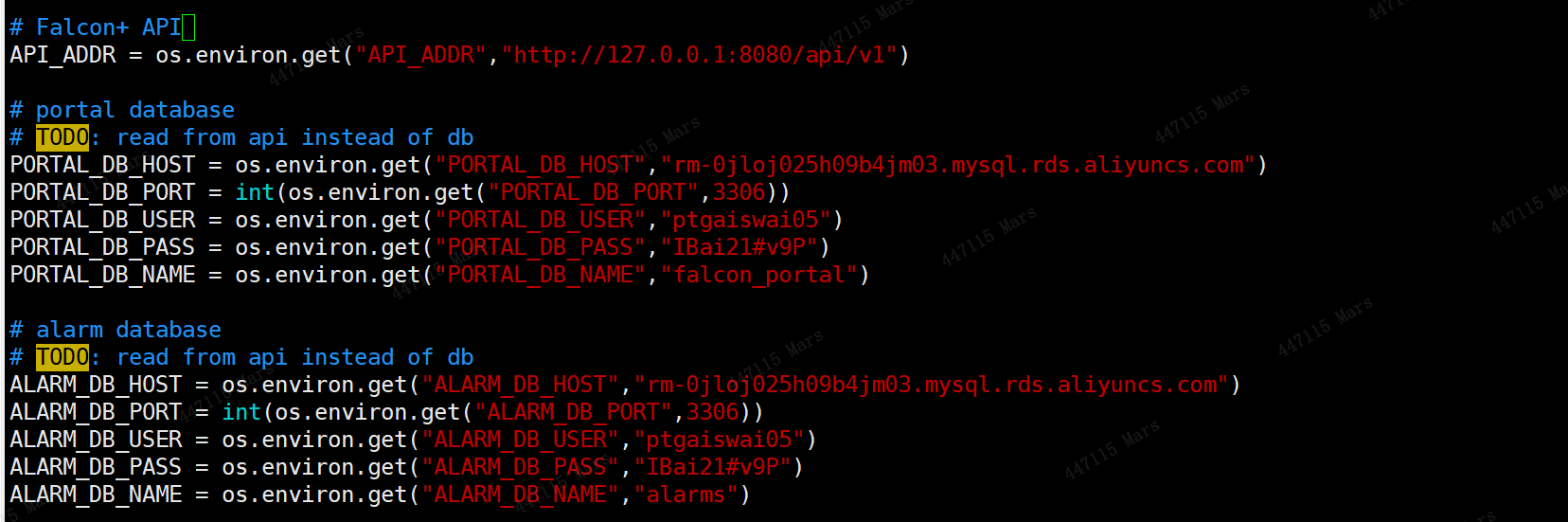
**-v /etc/localtime:/etc/localtime \**

**--network=host \**

**openfalcon/falcon-plus:v0.3 \**

**./agent/bin/falcon-agent -c /open-falcon/agent/config/cfg.json**

**修改dashboard配置文件中的DB部分 vim /open-falcon/dashboard/rrd/config.py**

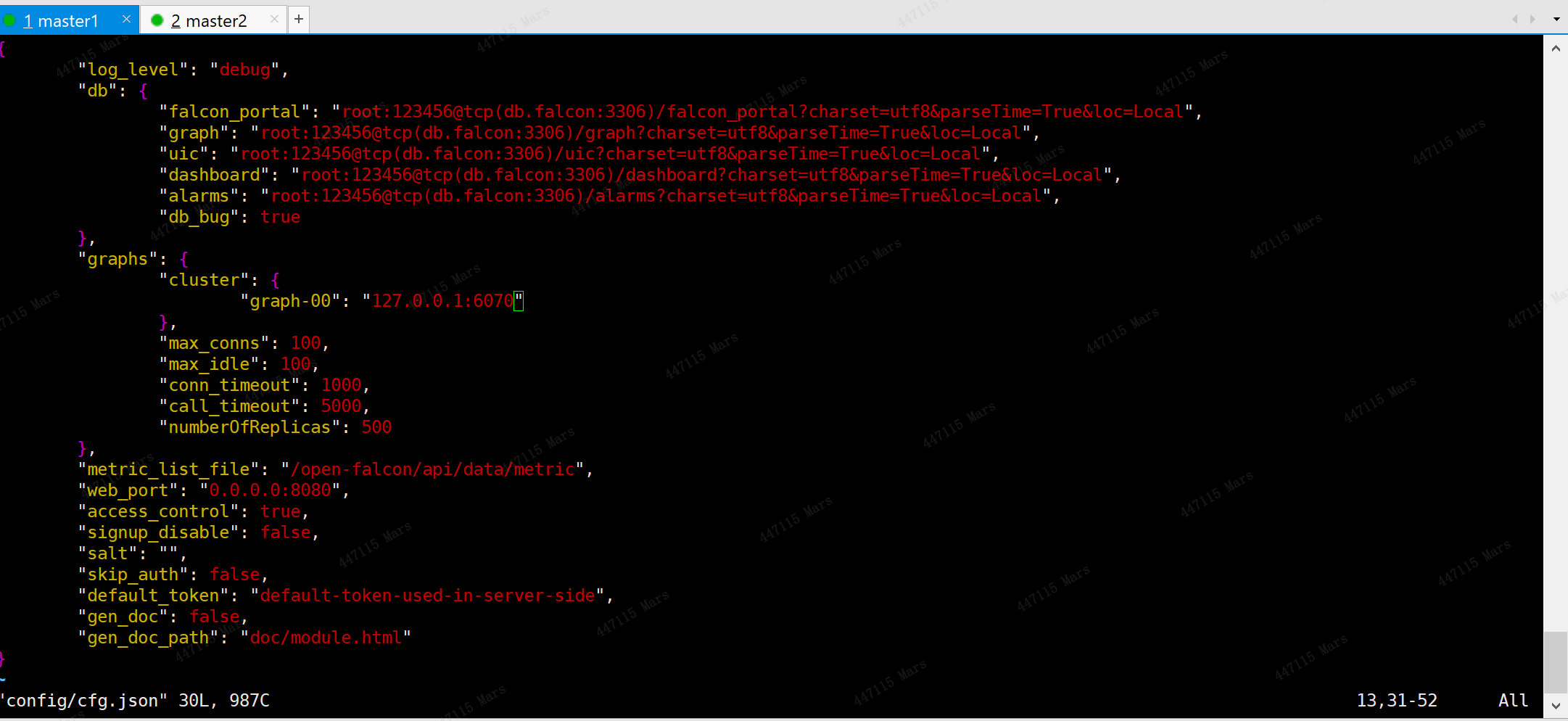


此方法得到的效果与方法二一致，即open-falcon的各个组件部分的配置文件中关于数据库部分全部修改为远程数据库并且不需要单独修改容器的配置。

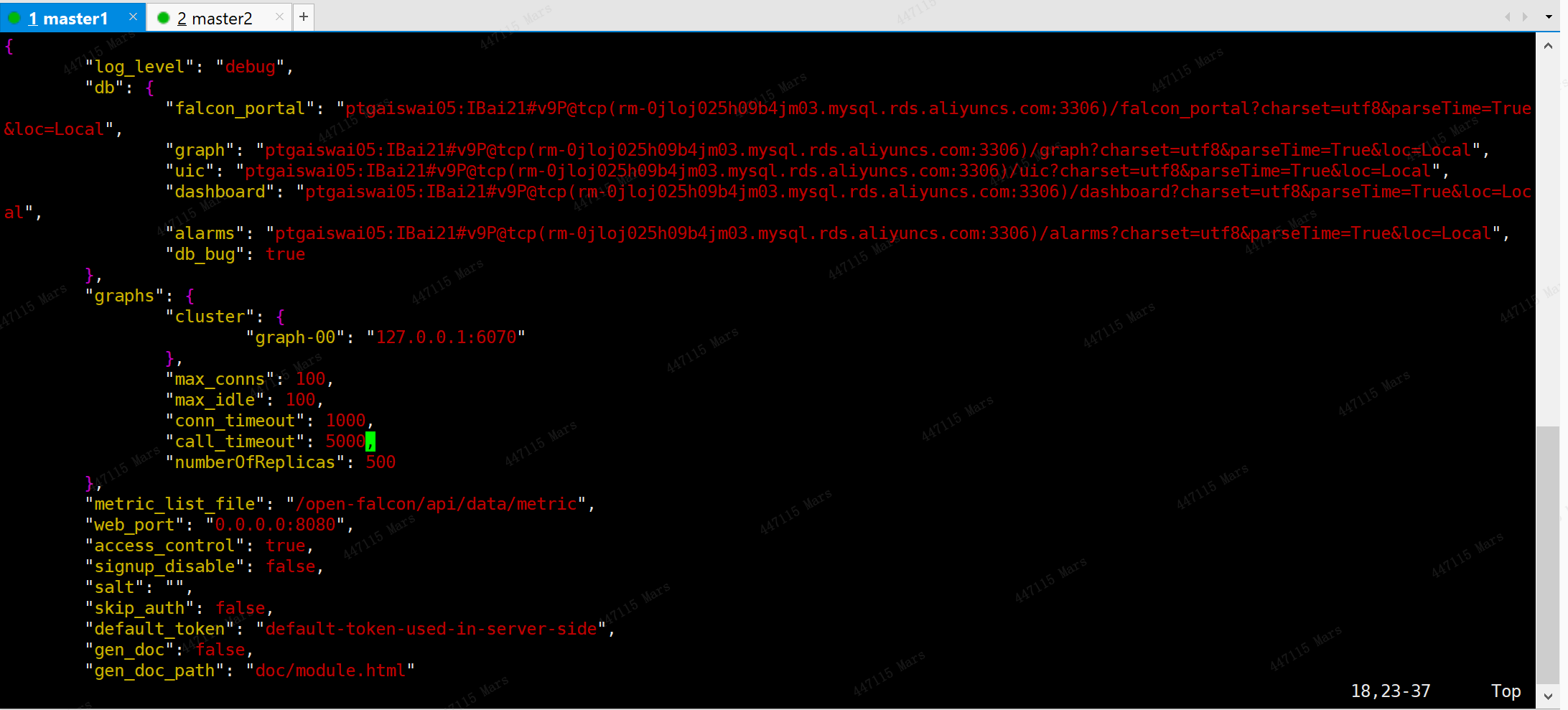
**方法二：复杂，不推荐使用，可能会出现失败的情况**

1. 修改api组件的配置文件中数据库部分 vim /open-falcon/api/config/cfg.json

修改前：



修改后：



1. 修改graph组件的配置文件中数据库部分 vim /open-falcon/graph/config/cfg.json

修改前：

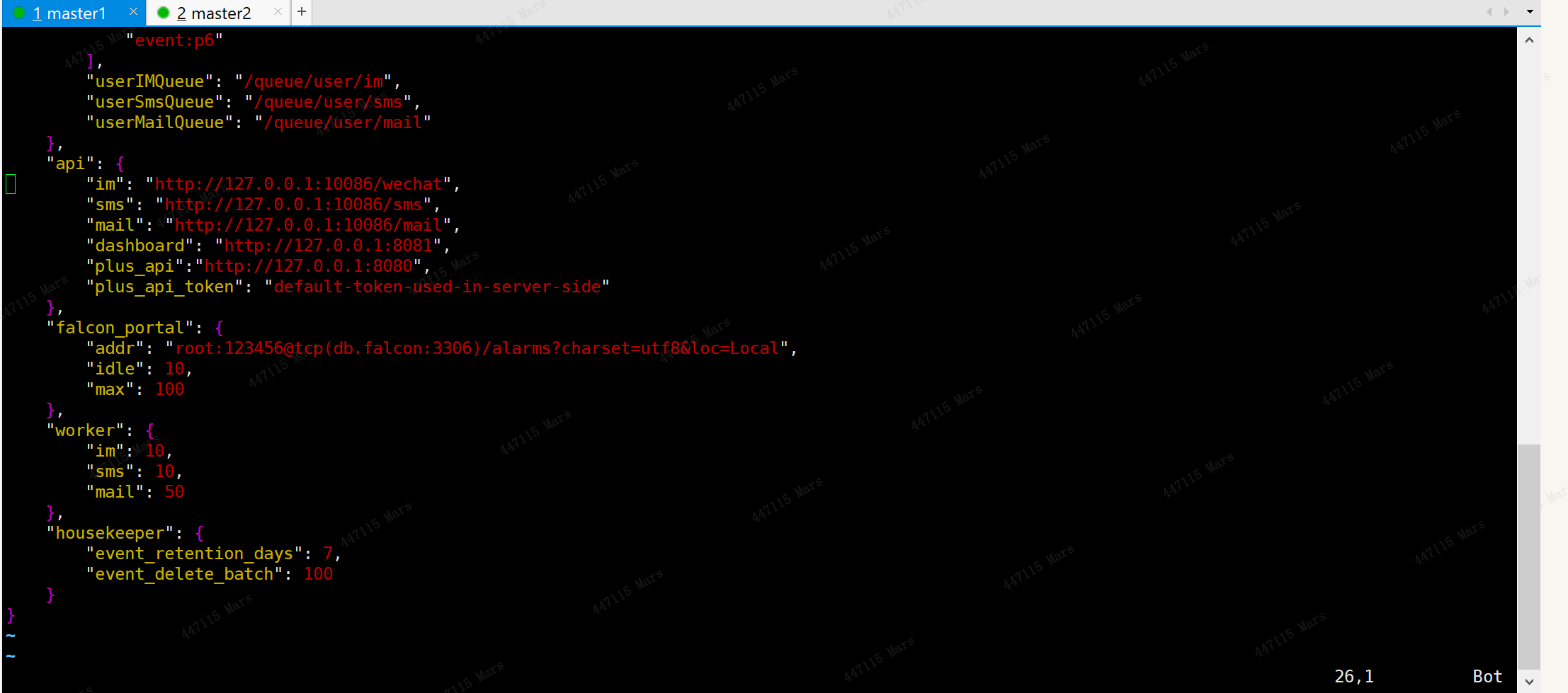


修改后：

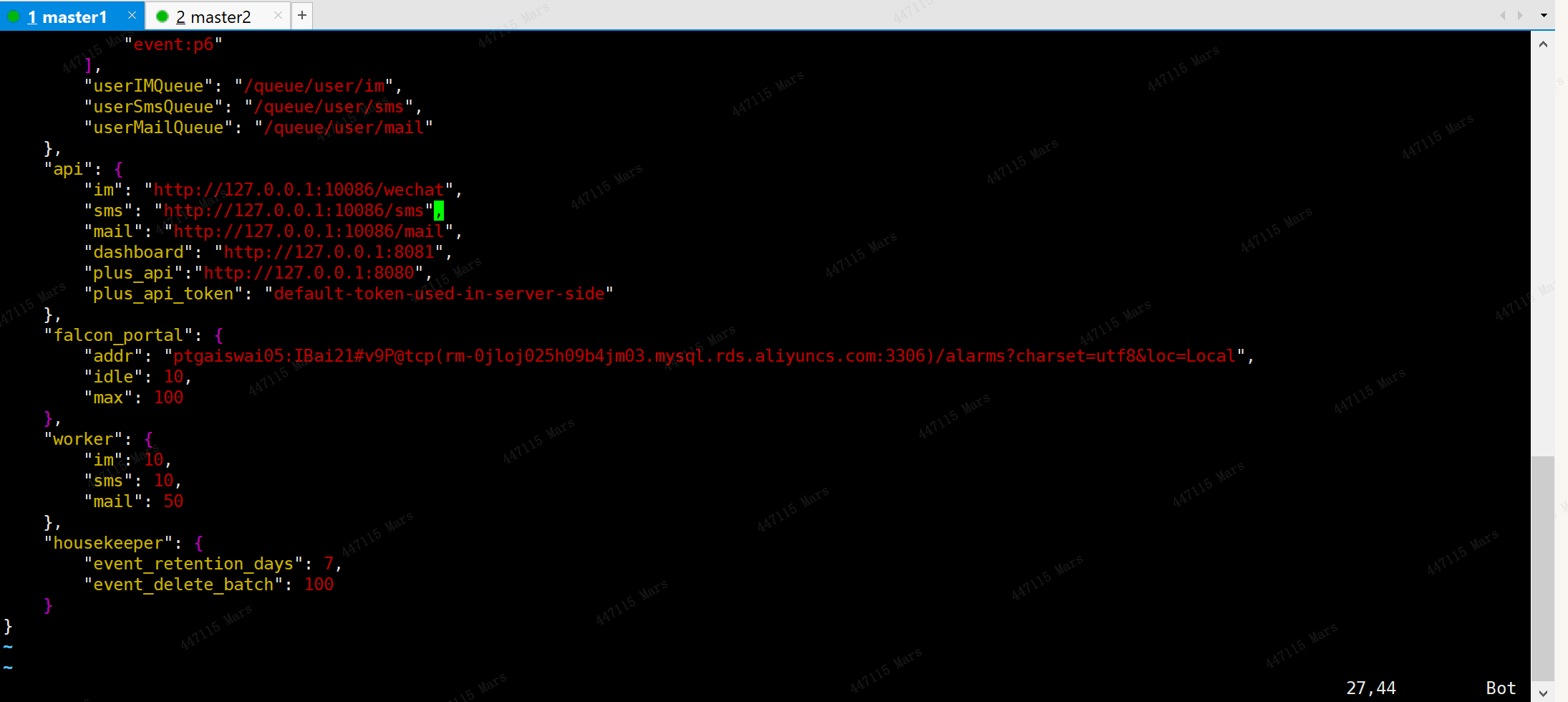


1. 修改alarm组件的配置文件中数据库部分vim /open-falcon/alarm/config/cfg.json

修改前：

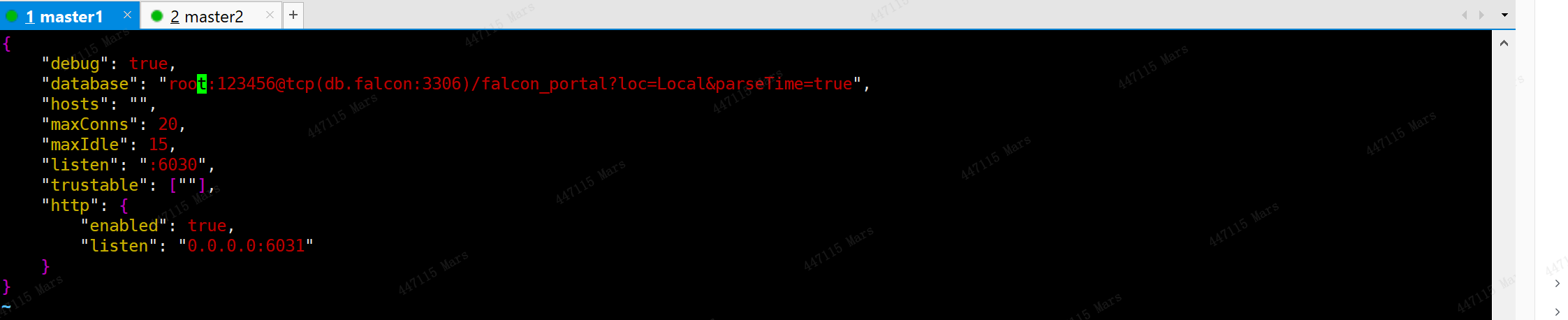


修改后：

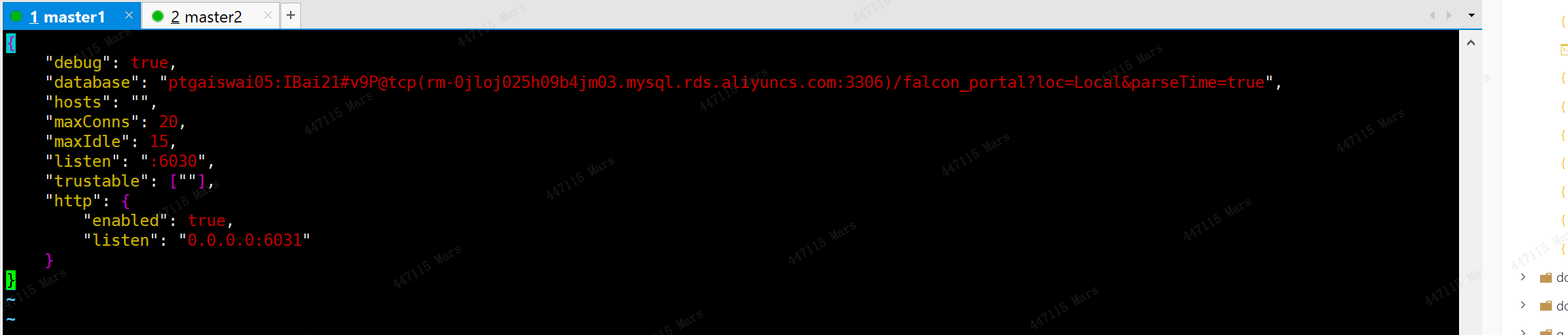


1. 修改hbs组件的配置文件中数据库部分vim /open-falcon/hbs/config/cfg.json

修改前：

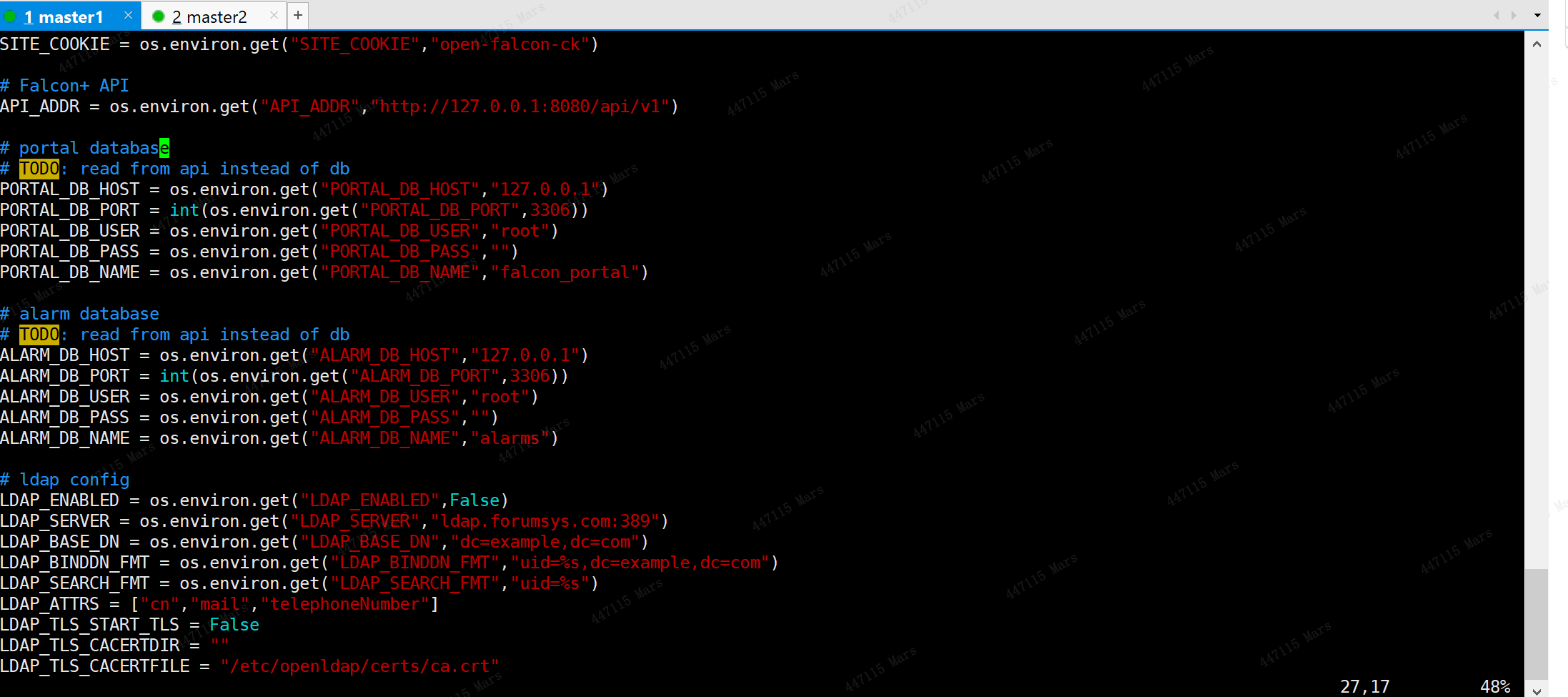


修改后：

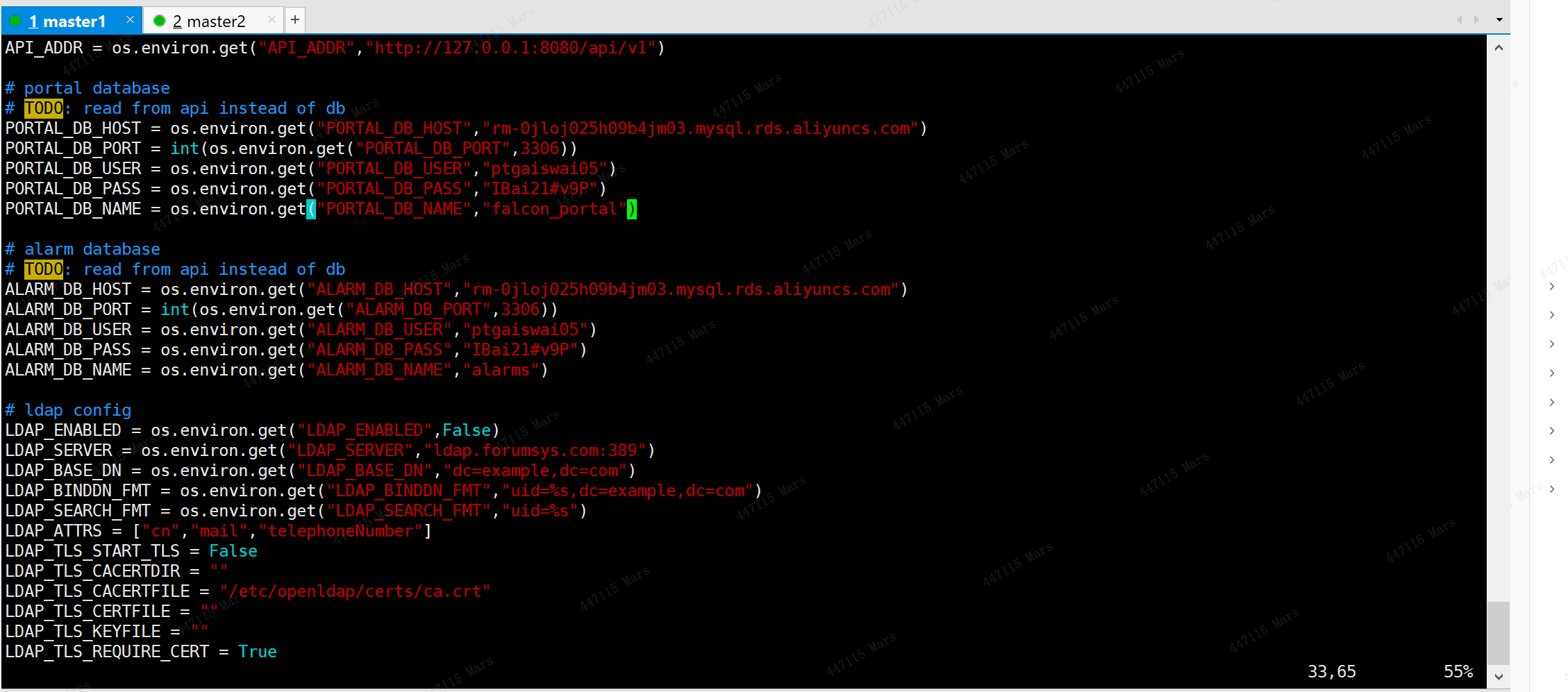


1. 修改dashboard配置文件 vim /open-falcon/dashboard/rrd/config.py

修改前：

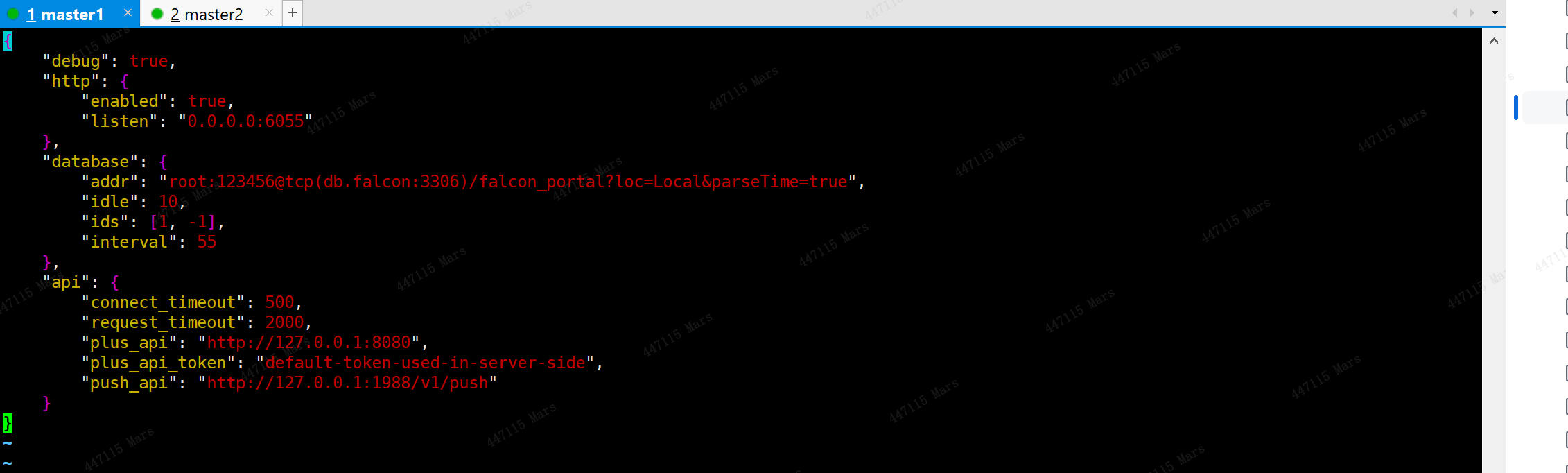


修改后：

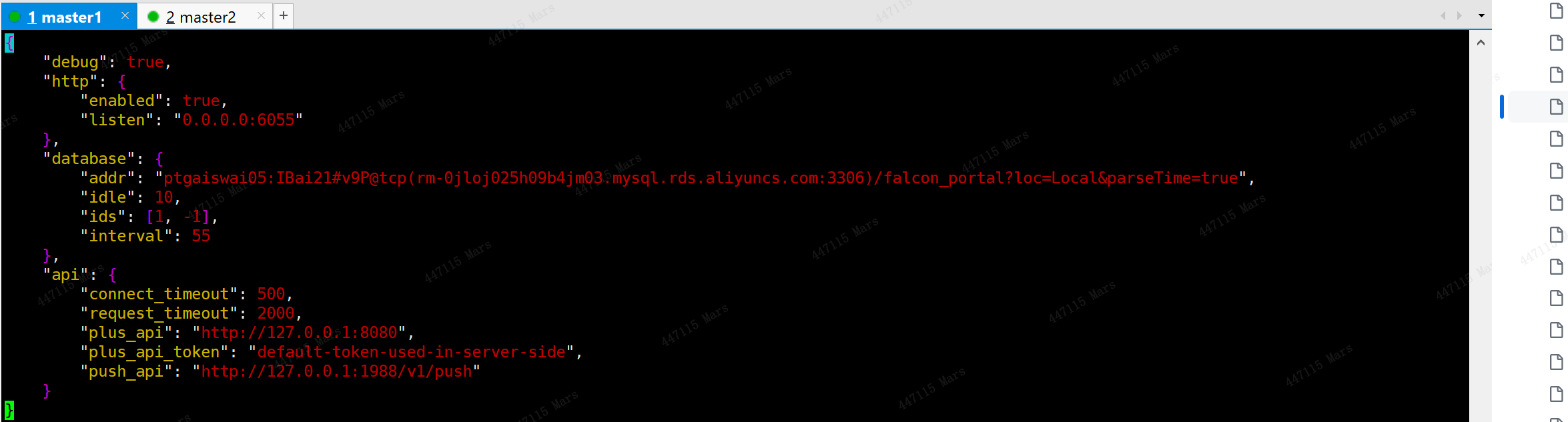


1. 修改aggregator组件的配置文件中数据库部分vim /open-falcon/aggregator/config/cfg.json

修改前：



修改后：



1. 最后修改容器的配置文件中的db部分为远程数据库。

### 支持远程使用密码的Redis：

在master2服务器上用docker方式搭建远程redis，并且设置密码123456，开放端口6379

**docker run -p 6379:6379 --name falcon-redis --restart=always -v /home/redis/data:/data -v /home/redis/conf/redis.conf:/etc/redis/redis.conf -d redis:4-alpine3.8 redis-server /etc/redis/redis.conf**

需要在服务器上新建redis的目录并将容器上的redis目录挂载到服务器上，并且在服务器中下载redis.conf文件修改其中内容加入密码、持久化等设置。

创建容器修改REDIS\_PORT部分，原文件中将容器与本地redis用—link方式连接并且设置了无密码的REDIS\_PORT。根据open-falcon项目中modules\alarm\g\redis.go的InitRedisConnPool函数，实际上是对模块的配置文件中的redis部分进行解析得到redis地址、端口和密码，因此只需要修改传入的REDIS\_PORT为符合解析的格式，就可以正确的连接远程redis。

**docker run -itd --name falcon-plus \**

**-p 8433:8433 \**

**-p 8080:8080 \**

**-p 18433:18433 \**

**-p 6030:6030 \**

**-e MYSQL\_PORT=ptgaiswai05:IBai21#v9P@tcp\(rm-0jloj025h09b4jm03.mysql.rds.aliyuncs.com:3306\) \**

**-e REDIS\_PORT=123456@11.165.57.199:6379 \**

**-v /home/work/open-falcon/data:/open-falcon/data \**

**-v /home/work/open-falcon/logs:/open-falcon/logs \**

**-v /etc/localtime:/etc/localtime \**

**openfalcon/falcon-plus:v0.3**



由formatRedisAddr函数可以分析出将模块的配置文件中的Redis部分的Addr部分进行拆分，以@符合分割，所以将传入的REDIS\_PORT设置为：password@ip:port 的格式。

### 总体流程：

两台服务器分别命名为master1和master2，在master1服务器上用docker的方式搭建open-falcon的基础环境，作为server部分；在master2服务器上用docker的方式搭建agent组件环境，作为agent部分。

远程MySQL数据库：

地址和端口号：rm-0jloj025h09b4jm03.mysql.rds.aliyuncs.com:3306

账号：ptgaiswai05:IBai21#v9P

密码：IBai21#v9P

远程Redis：

地址和端口号： 11.165.57.199:6379

密码：123456

**一．安装open-falcon**

**docker run -itd --name falcon-plus \**

**-p 8433:8433 \**

**-p 8080:8080 \**

**-p 18433:18433 \**

**-p 6030:6030 \**

**-e MYSQL\_PORT=ptgaiswai05:IBai21#v9P@tcp\(rm-0jloj025h09b4jm03.mysql.rds.aliyuncs.com:3306\) \**

**-e REDIS\_PORT=123456@11.165.57.199:6379 \**

**-v /home/work/open-falcon/data:/open-falcon/data \**

**-v /home/work/open-falcon/logs:/open-falcon/logs \**

**-v /etc/localtime:/etc/localtime \**

**openfalcon/falcon-plus:v0.3**

注：1.-itd中-i指定以交互模式运行容器，并保持终端激活，-t指定为容器分配一个伪终端，-d指定以后台模式运行容器，并返回容器ID

2.-p开放四个端口用于hbs等组件的信息交互和dashboard的前端访问

3.-e配置Redis和MySQL的远程连接信息

4.-v将open-falcon的目录挂载到服务器本地，并且设置容器时间与服务器时间同步

5.指定使用最新版本v0.3

**二．安装open-falcon的dashboard部分**

**docker run -itd --name falcon-dashboard \**

**-p 8081:8081 \**

**--link=falcon-plus:api.falcon \**

**-e API\_ADDR=http://api.falcon:8080/api/v1 \**

**-e PORTAL\_DB\_HOST=rm-0jloj025h09b4jm03.mysql.rds.aliyuncs.com \**

**-e PORTAL\_DB\_PORT=3306 \**

**-e PORTAL\_DB\_USER=ptgaiswai05 \**

**-e PORTAL\_DB\_PASS=IBai21#v9P \**

**-e PORTAL\_DB\_NAME=falcon\_portal \**

**-e ALARM\_DB\_HOST=rm-0jloj025h09b4jm03.mysql.rds.aliyuncs.com \**

**-e ALARM\_DB\_PORT=3306 \**

**-e ALARM\_DB\_USER=ptgaiswai05 \**

**-e ALARM\_DB\_PASS=IBai21#v9P \**

**-e ALARM\_DB\_NAME=alarms \**

**-v /etc/localtime:/etc/localtime \**

**-w /open-falcon/dashboard openfalcon/falcon-dashboard:v0.2.1 \**

**'./control startfg'**

注：1.-itd中-i指定以交互模式运行容器，并保持终端激活，-t指定为容器分配一个伪终端，-d指定以后台模式运行容器，并返回容器ID

2.-p，dashboard通过8081端口访问

3.-e配置dashboard所需数据库MySQL的远程连接信息

4.—link将与falcon-plus的api部分进行连接

5.需要设置容器时间与服务器时间同步，并且指定版本v0.2.1

三．**安装agent**

**sudo docker run -d --restart always --name falcon-agent \**

**-e NUX\_ROOTFS=/rootfs \**

**-v /:/rootfs:ro \**

**-v /etc/localtime:/etc/localtime \**

**--network=host \**

**openfalcon/falcon-plus:v0.3 \**

**./agent/bin/falcon-agent -c /open-falcon/agent/config/cfg.json**

注：1. -d指定以后台模式运行容器，并返回容器ID

2. --restart always 设置容器开机启动

3. --network 将容器连接到host网络

4. -v /:/rootfs:ro \ 把系统的/目录的东西，拷贝到容器/rootfs中，然后给这个目录赋予ro权限

5.需要设置容器时间与服务器时间同步，并且指定版本v0.3

**四．修改agent配置文件**

1.agent部分搭建好后遇到ss工具无法正常使用的情况，需要配置agent环境，参考：

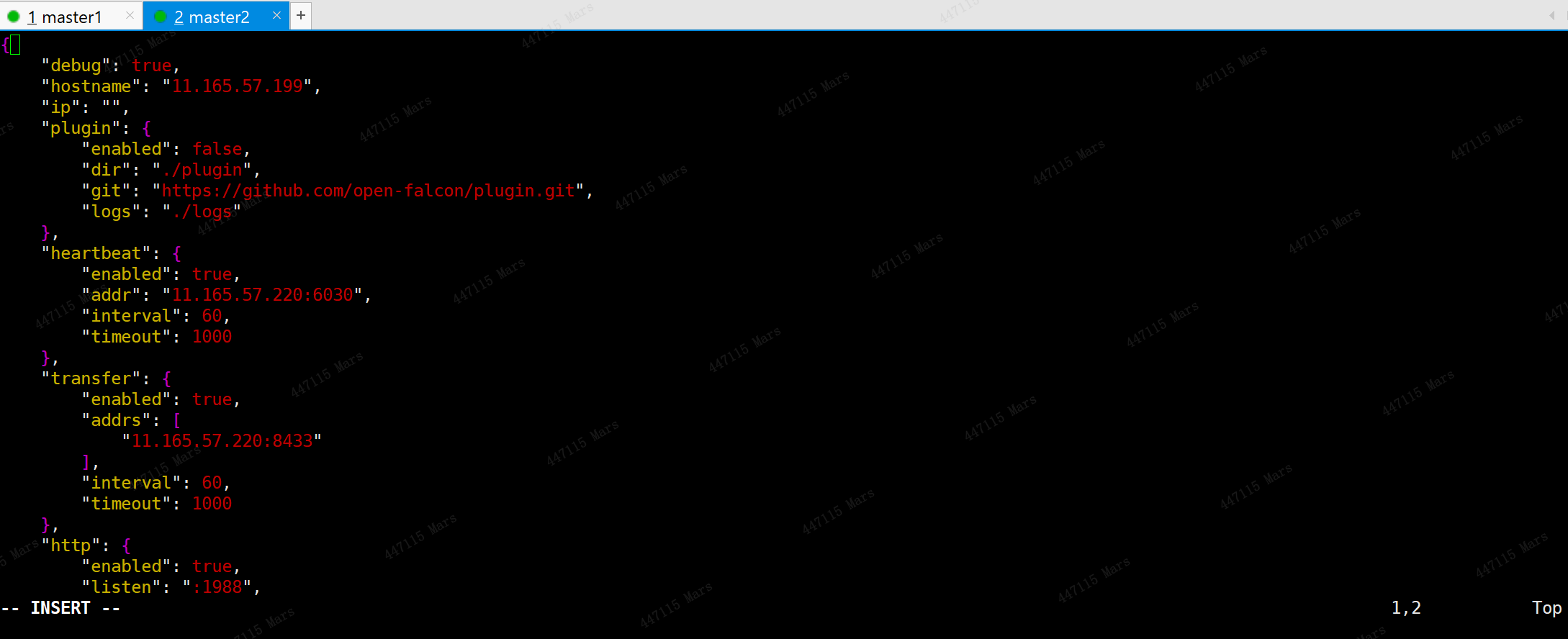
<https://aliyuque.antfin.com/alinpu_engineering/og0bia/edhumgxu0sshiv7q>

2.需要修改配置文件使得指向server端：

vim /open-falcon/agent/config/cfg.json

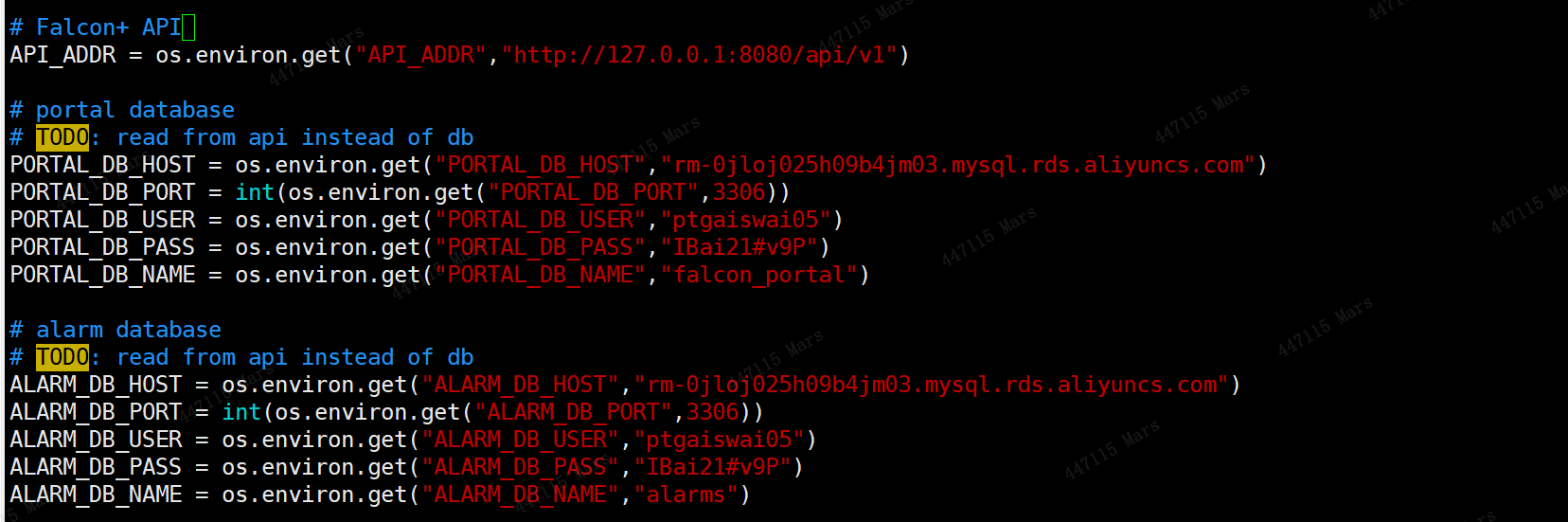
修改其中的hostname：唯一辨别agent的属性

修改heartbeat和transfer中的addr部分，使得ip指向服务器ip，修改后如下：



**五．修改dashboard配置文件**

vim /open-falcon/dashboard/rrd/config.py



API\_ADDR 表示后端api组件的地址，一般不修改指向本地

修改PORTAL\_DB和ALARM\_DB的地址，指向远程数据库

**六．启动open-falcon并检查配置**

启动open-falcon的所有组件：

docker exec falcon-plus sh ctrl.sh start \

graph hbs judge transfer nodata aggregator agent gateway api alarm

启动/停止/重启某个单独的组件：

docker exec falcon-plus sh ctrl.sh start/stop/restart xxx

检查open-falcon的组件是否成功启动：

docker exec falcon-plus ./open-falcon check

启动所有组件后，可以通过docker exec -it falcon-plus bash进入容器中，vim /open-falcon/xxx/config/cfg.json查看各个组件的配置信息，容器中是Alpine系统，首次进入需要安装vim，可以通过sed -i 's/dl-cdn.alpinelinux.org/mirrors.aliyun.com/g' /etc/apk/repositories 进行换源加快下载速度，apk add vim进行下载。

至此所有部分完毕。