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

为了把所有组件的配置文件放在同一个文件夹中统一管理，容器中所有配置文件统一放在/open-falcon/configs目录中，服务器中所有配置文件统一放在/home/work/open-falcon/configs目录中，以xxx\_cfg.json命名。

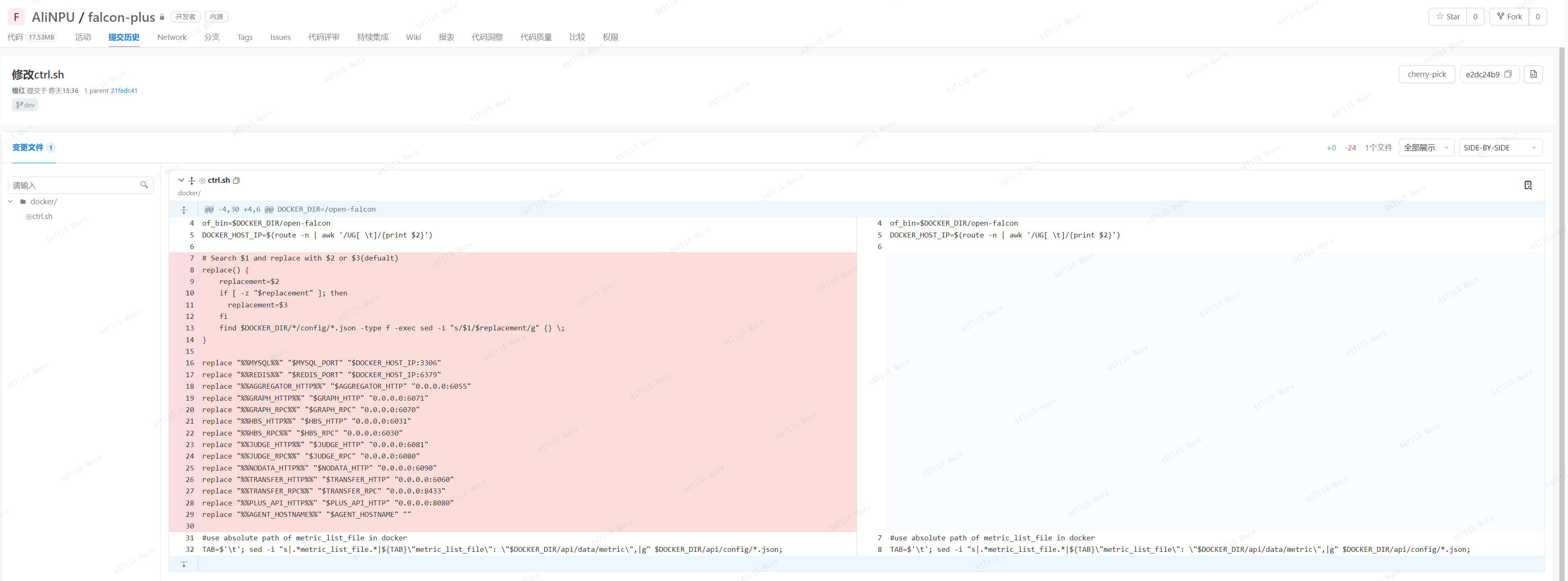
1. **修改系统工具Supervisor的配置文件**

在启动文件ctrl.sh中，使用的组件启动命令的配置文件是在各个组件的目录下config文件夹中的cfg.json，即/open-falcon/xxx/config/cfg.json。在对单独的组件进行启动时，还会直接显示log不会以后台的形式进行启动，因此将启动部分交于supervisor进行统一管理，需要对supervisor的配置文件进行修改。

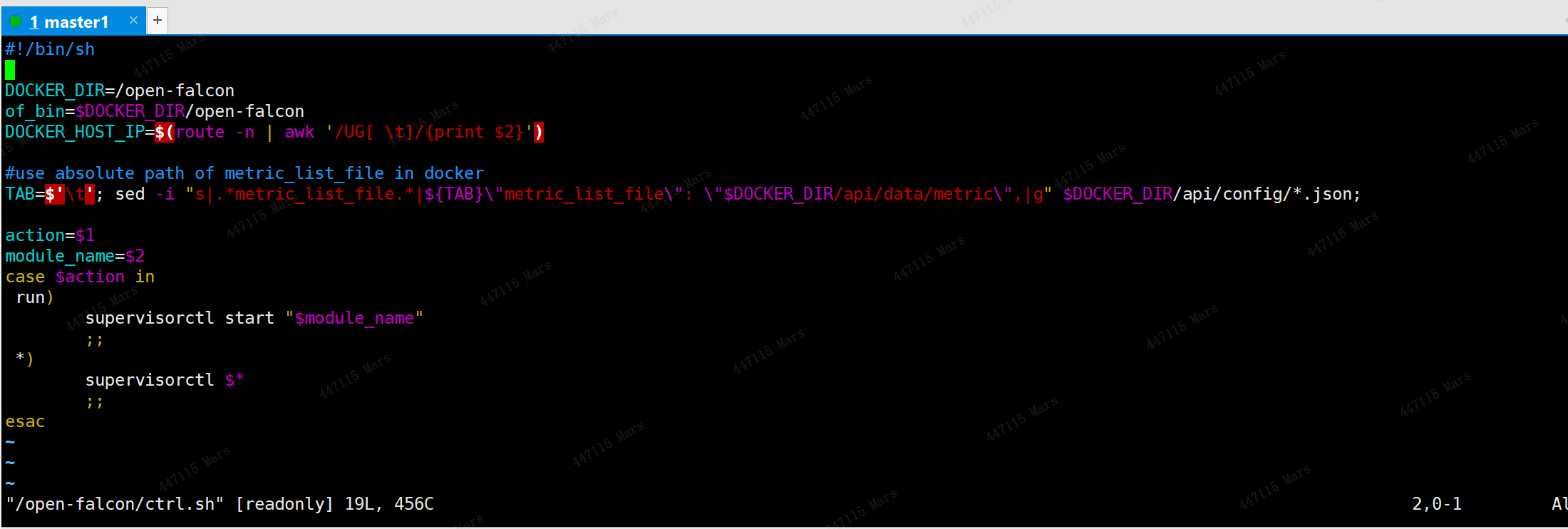
Supervisor的配置文件在容器中的路径为：/etc/supervisord.conf。将各个组件的启动配置文件设置为统一配置文件夹下的/open-falcon/configs/xxx\_cfg.json。已经对此修改，并且复制到服务器master1中，路径为/home/work/open-falcon/etc/supervisord.conf。在之后的启动命令中只需将其挂载到容器中即可。

1. **修改open-falcon的组件启动文件ctrl.sh**

在启动文件ctrl.sh中，利用docker构建容器后会对组件的配置文件进行遍历，修改其中的各个连接ip和端口的值，如果定义相应的环境变量则会修改为环境变量的值，否则设置成默认值0:0:0:0：port，为了挂载配置文件后不再对配置文件进行修改，删除这一部分。

****

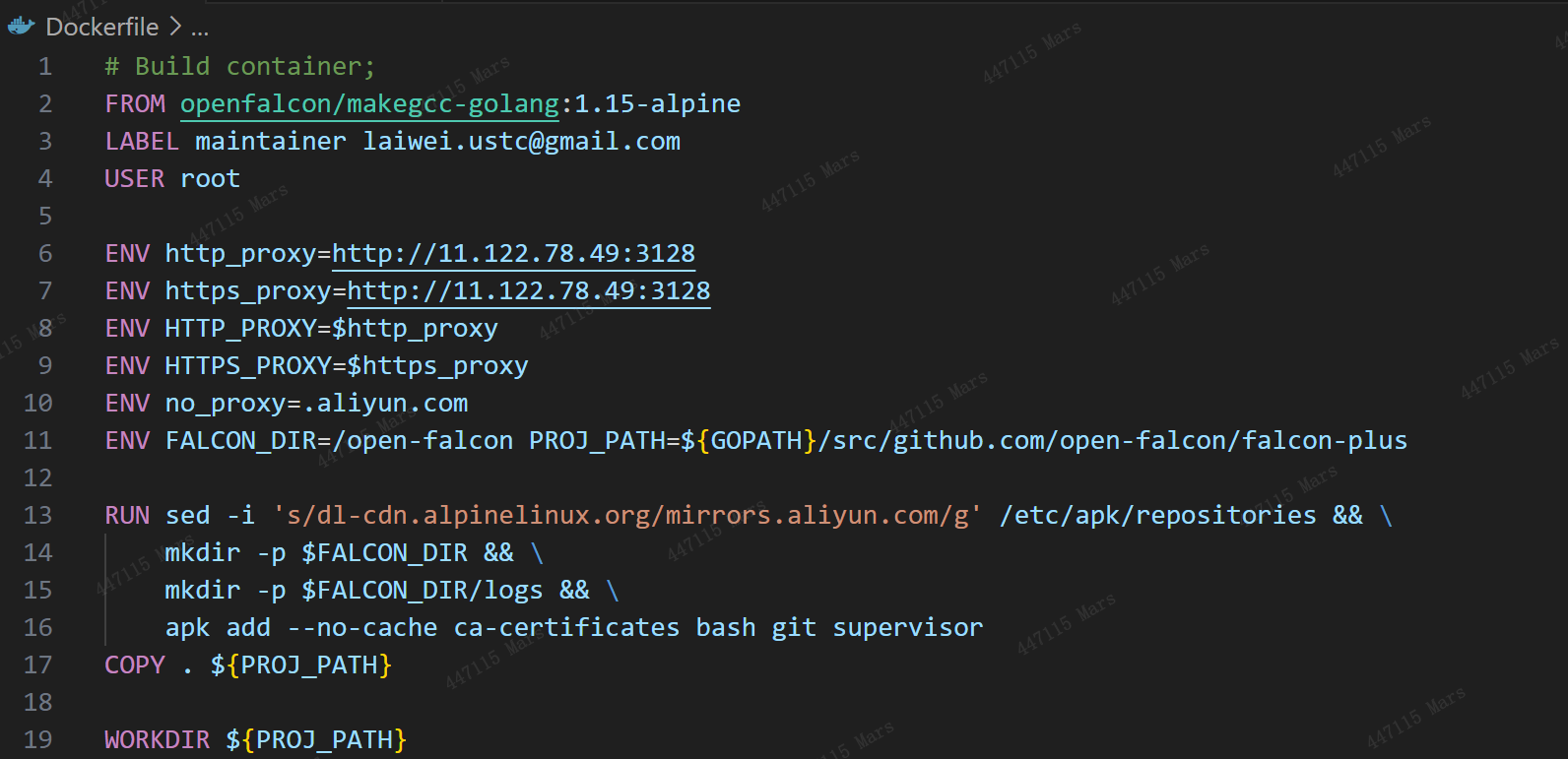
对启动命令进行修改，将启动命令改为由Supervisor进行处理。修改后文件内容如下：



在gitlab上进行修改后，将该文件存放在master1服务器上的/tmp/docker\_build\_image目录下。

1. **修改Dockerfile文件构建镜像**

在Dockerfile文件中加入代理解决连接超时的问题，并且加入阿里云镜像源加快镜像构建速度，修改后Dockerfile文件如下：



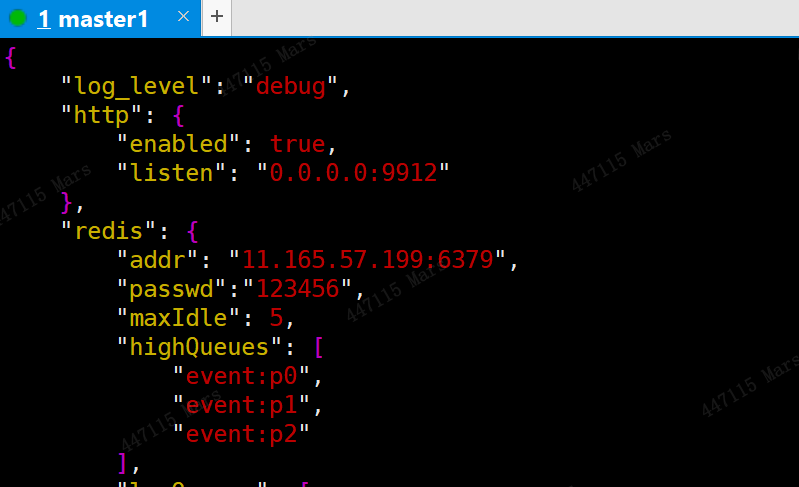
git clone -b dev [git@gitlab.alibaba-inc.com:chenhongquan.chq/falcon-plus.git](mailto:git@gitlab.alibaba-inc.com:chenhongquan.chq/falcon-plus.git)

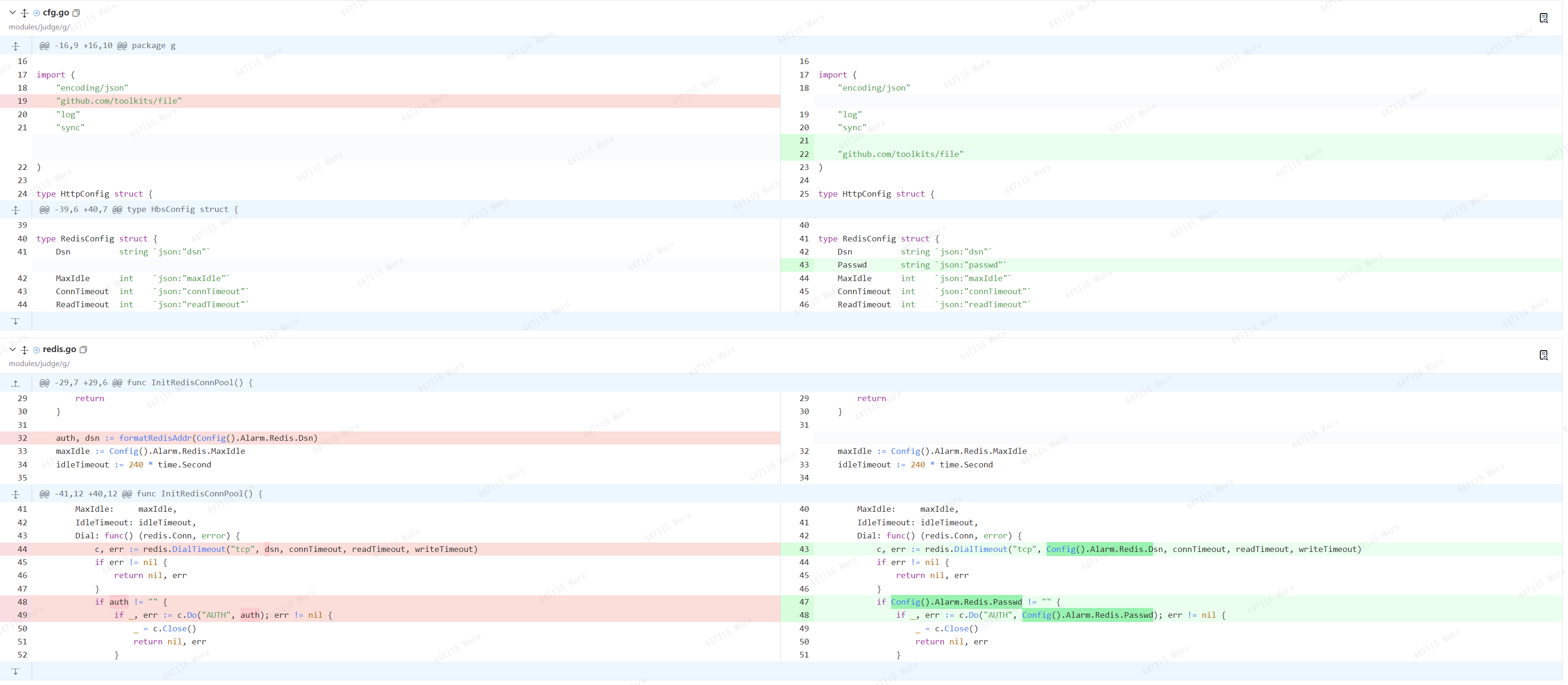
docker build -t falcon-plus-test:v0.2 .

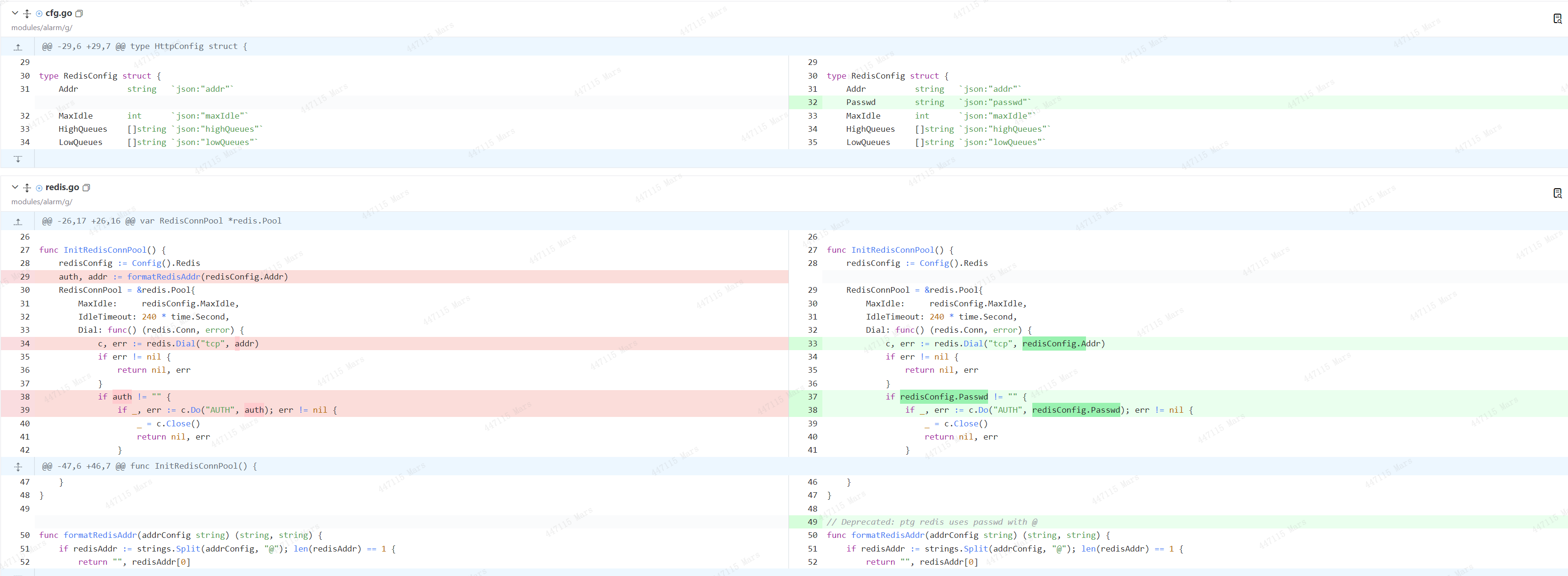
目前服务器master上构建镜像为falcon-plus-test:v0.2。

1. **修改源代码中的Redis验证逻辑和对应的config配置文件中Redis属性**

在源代码中分别修改judge和alarm组件中对Redis的读取和验证部分的代码，将Redis的原本配置方式：password@host:port修改为单独存放密码的格式，并且修改读取配置文件部分。





1. **Server启动命令**

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

**-p 8433:8433 \**

**-p 8080:8080 \**

**-p 18433:18433 \**

**-p 6030:6030 \**

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

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

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

**-v /home/work/open-falcon/etc/supervisord.conf:/etc/supervisord.conf \**

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

**falcon-plus-test:v0.2**

1. **Dashboard配置文件挂载和启动命令**

将修改好的Dashboard配置文件/open-falcon/dashboard/rrd/config.py拷贝到master1服务器上的/home/work/open-falcon/dashboard/rrd目录中，启动时进行挂载

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

**-p 8081:8081 \**

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

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

**-v /home/work/open-falcon/dashboard/rrd/config.py:/open-falcon/dashboard/rrd/config.py \**

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

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

**'./control startfg'**

1. **其他**

1.新构建open-falcon容器后需要加入阿里镜像源来提高容器内的下载工具速度：

sed -i 's/dl-cdn.alpinelinux.org/mirrors.aliyun.com/g' /etc/apk/repositories

2. 容器中是Alpine系统，容器内需要下载vim等工具使用apk add xxx/vim进行下载

3. 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

4.启动/停止/重启某个单独的组件：

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

5.检查open-falcon的组件是否成功启动：

docker exec falcon-plus ./open-falcon check