



链滴

# 小米开源监控 Open-Falcon 3.0 部署

作者: [GeekBoyDqz](#)

原文链接: <https://ld246.com/article/1567855718747>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## 一、系统环境准备

目前我这里使用单机版进行部署，也可以前后端进行分离，前端，后端，数据库分别在不同的服务器进行部署搭建

### 1、更改主机名

```
[root@localhost ~]# hostnamectl set-hostname open-falcon
```

```
[root@localhost ~]# su
```

```
[root@open-falcon ~]#
```

### 2、配置阿里源

```
####备份系统BASE源
```

```
[root@open-falcon ~]# mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.backup
```

```
####阿里源####
```

```
[root@open-falcon ~]# wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
```

```
####epel源####
```

```
[root@open-falcon ~]# wget -O /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel.repo
```

el-7.repo

####更新缓存####

```
[root@open-falcon ~]# *yum clean all*
```

```
[root@open-falcon ~]# yum makecache
```

```
[root@open-falcon ~]# yum repolist
```

### 3、安装基础命令

```
[root@localhost ~]# yum -y install vim net-tools lrzsz ntp wget get curl rsync git
```

### 4、时间同步

```
[root@open-falcon ~]# systemctl start ntpd
```

```
[root@open-falcon ~]# systemctl enable ntpd
```

```
[root@open-falcon ~]# systemctl status ntpd
```

```
[root@open-falcon ~]# date #检查时间
```

### 5、关闭防火墙

```
[root@open-falcon ~]# systemctl stop firewalld
```

```
[root@open-falcon ~]# systemctl disable firewalld
```

```
[root@open-falcon ~]# setenforce 0
```

```
[root@open-falcon ~]# sed -ri 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
```

## 二、软件环境准备

### 1、部署go

**注意：**官方yum和阿里yum都没有go的安装包，只能通过fedora的epel仓库来安装

#安装

```
[root@open-falcon ~]# yum install -y epel-release
```

```
[root@open-falcon ~]# yum install golang -y
```

#检查

```
[root@open-falcon ~]# go version
```

```
go version go1.11.5 linux/amd64
```

### 2、部署redis

部署新版redis

```
[root@open-falcon ~]# yum install -y http://rpms.famillecollet.com/enterprise/remi-release-7.
pm
```

#安装

```
[root@open-falcon ~]# yum --enablerepo=remi install redis
```

#启动

```
[root@open-falcon ~]# systemctl start redis
```

#开机自启

```
[root@open-falcon ~]# systemctl enable redis
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/redis.service to /usr/lib/sy
temd/system/redis.service.
```

#检查状态

```
[root@open-falcon ~]# systemctl status redis
```

### 3、部署MySQL

yum源为5.5版本，我这里安装5.7版本

#### 3.1: 下载MySQL源

```
[root@open-falcon ~]# wget https://dev.mysql.com/get/mysql57-community-release-el7-11.
oarch.rpm
```

#### 3.2: 安装 yum Repository

```
[root@open-falcon ~]# rpm -ivh mysql57-community-release-el7-11.noarch.rpm
```

#### 3.3: 安装 MySQL 服务端

#查看mysql server

```
[root@open-falcon ~]# yum repolist enabled | grep "mysql.*-community.*"
```

```
[root@open-falcon ~]# yum repolist all | grep mysql
```

#安装

```
[root@open-falcon ~]# yum install mysql-community-server
```

#### 3.4: 启动 mysql 服务

```
[root@open-falcon ~]# systemctl start mysqld.service
```

```
[root@open-falcon ~]# systemctl enable mysqld.service
```

```
[root@open-falcon ~]# systemctl status mysqld.service
```

#### 3.5: 登陆数据库

MySQL5.7.6 之后会在启动 mysql 进程的时候生成一个用户密码，首次登陆需要这个密码才行。密码保存在 mysql 进程的日志里，即(/var/log/mysqld.log)

#查看登陆密码

```
[root@open-falcon ~]# cat /var/log/mysqld.log | grep 'password'
```

#登陆

```
[root@open-falcon ~]# mysql -uroot -p
```

## 5.6: 修改 root 密码

```
mysql> show databases;
ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.
```

以前的 password()函数将会被抛弃，官方建议使用下面的命令来修改密码

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';
```

同时，如果你设置的密码过于简单也会报错。

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '123';
```

```
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
```

解决密码过短问题

```
mysql> set global validate_password_policy=0;
```

再次设置密码

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '123';
```

# 三、部署open-falcon

## 1、部署后端

### 1.1: 初始化数据库

```
[root@open-falcon ~]# cd /tmp/ && git clone https://github.com/open-falcon/falcon-plus.gi
```

```
[root@open-falcon ~]# cd /tmp/falcon-plus/scripts/mysql/db_schema/
```

```
[root@open-falcon ~]# mysql -h 127.0.0.1 -u root -p < 1_uic-db-schema.sql
```

```
[root@open-falcon ~]# mysql -h 127.0.0.1 -u root -p < 2_portal-db-schema.sql
```

```
[root@open-falcon ~]# mysql -h 127.0.0.1 -u root -p < 3_dashboard-db-schema.sql
```

```
[root@open-falcon ~]# mysql -h 127.0.0.1 -u root -p < 4_graph-db-schema.sql
```

```
[root@open-falcon ~]# mysql -h 127.0.0.1 -u root -p < 5_alarms-db-schema.sql
```

```
[root@open-falcon ~]# rm -rf /tmp/falcon-plus/
```

## 1.2: 创建工作目录

```
[root@open-falcon ~]# vim /etc/profile  
  
export FALCON_HOME=/usr/local/open-falcon  
  
[root@open-falcon ~]# source /etc/profile  
  
[root@open-falcon ~]# mkdir -p $FALCON_HOME
```

## 1.3: 程序包下载

| 将源码解压到该路径

```
[root@open-falcon ~]# wget https://github.com/open-falcon/falcon-plus/releases/download  
v0.3/open-falcon-v0.3.tar.gz  
  
[root@open-falcon ~]# tar -zxvf open-falcon-v0.3.tar.gz -C $FALCON_HOME  
  
[root@open-falcon ~]# cd $FALCON_HOME
```

## 1.4: 修改配置文件cfg.json

```
[root@open-falcon ~]# vim /usr/local/open-falcon/aggregator/config/cfg.json  
%s#root:#root:passwd#g
```



```
{  
  "debug": true,  
  "http": {  
    "enabled": true,  
    "listen": "0.0.0.0:6855"  
  },  
  "database": {  
    "addr": "root:dqz@tcp(127.0.0.1:3306)/falcon_portal?loc=Local&parseTime=true",  
    "idle": 10,  
    "ids": [1, -1],  
    "interval": 5  
  },  
  "api": {  
    "connect_timeout": 500,  
    "request_timeout": 2000,  
    "plus_api": "http://127.0.0.1:8080",  
    "plus_api_token": "default-token-used-in-server-side",  
    "push_api": "http://127.0.0.1:1988/v1/push"  
  }  
}
```

```
[root@open-falcon ~]# vim /usr/local/open-falcon/graph/config/cfg.json  
%s#root:#root:passwd#g
```

```
{
  "debug": false,
  "http": {
    "enabled": true,
    "listen": "0.0.0.0:6071"
  },
  "rpc": {
    "enabled": true,
    "listen": "0.0.0.0:6070"
  },
  "rrd": {
    "storage": "./data/6070"
  },
  "db": {
    "dsn": "root:@tcp(127.0.0.1:3306)/graph?loc=Local&parseTime=true",
    "maxIdle": 4
  },
  "callTimeout": 5000,
  "ioWorkerNum": 64,
  "migrate": {
    "enabled": false,
    "concurrent": 2,
    "password": "%s#root:#root:passwd#g"
  }
}
```

[root@open-falcon ~]# vim /usr/local/open-falcon/hbs/config/cfg.json

```
{
  "debug": true,
  "database": "root:@tcp(127.0.0.1:3306)/falcon_portal?loc=Local&parseTime=true",
  "hosts": "",
  "maxConns": 20,
  "maxIdle": 15,
  "listen": ":6030",
  "trustable": [""],
  "http": {
    "enabled": true,
    "listen": "0.0.0.0:6031"
  }
}
```

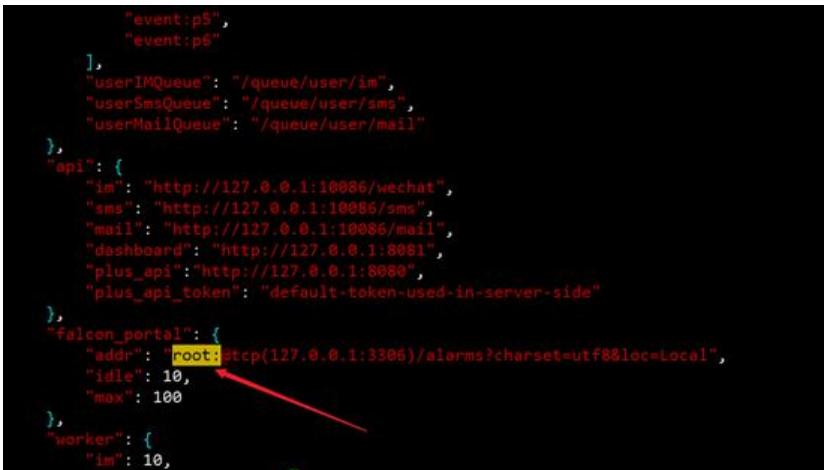
[root@open-falcon ~]# vim /usr/local/open-falcon/nodata/config/cfg.json

```
{
  "debug": true,
  "http": {
    "enabled": true,
    "listen": "0.0.0.0:6090"
  },
  "plus_api": {
    "connectTimeout": 500,
    "requestTimeout": 2000,
    "addr": "http://127.0.0.1:8080",
    "token": "default-token-used-in-server-side"
  },
  "config": {
    "enabled": true,
    "dsn": "root:@tcp(127.0.0.1:3306)/falcon_portal?loc=Local&parseTime=true&wait_timeout=604800",
    "maxIdle": 4
  },
  "collector": {
    "enabled": true,
    "batch": 200,
    "concurrent": 10
  }
}
```

[root@open-falcon ~]# vim /usr/local/open-falcon/api/config/cfg.json

```
{
  "log_level": "debug",
  "db": {
    "falcon_portal": "root:@tcp(127.0.0.1:3306)/falcon_portal?charset=utf8&parseTime=True&loc=Local",
    "graph": "root:@tcp(127.0.0.1:3306)/graph?charset=utf8&parseTime=True&loc=Local",
    "uic": "root:@tcp(127.0.0.1:3306)/uic?charset=utf8&parseTime=True&loc=Local",
    "dashboard": "root:@tcp(127.0.0.1:3306)/dashboard?charset=utf8&parseTime=True&loc=Local",
    "alarms": "root:@tcp(127.0.0.1:3306)/alarms?charset=utf8&parseTime=True&loc=Local",
    "db_bug": true
  },
  "graphs": {
    "cluster": {
      "graph-00": "127.0.0.1:6070"
    },
    "max_conns": 100,
    "max_idle": 100,
    "conn_timeout": 1000,
  }
}
```

```
[root@open-falcon ~]# vim /usr/local/open-falcon/alarm/config/cfg.json
```



```
    "event": "ps",
    "event": "ps"
  ],
  "userIMQueue": "/queue/user/im",
  "userSmsQueue": "/queue/user/sms",
  "userMailQueue": "/queue/user/mail"
},
"api": {
  "im": "http://127.0.0.1:10086/wechat",
  "sms": "http://127.0.0.1:10086/sms",
  "mail": "http://127.0.0.1:10086/mail",
  "dashboard": "http://127.0.0.1:8081",
  "plus_api": "http://127.0.0.1:8080",
  "plus_api_token": "default-token-used-in-server-side"
},
"falcon_portal": {
  "addr": "root:@tcp(127.0.0.1:3306)/alarms?charset=utf8&loc=Local",
  "idle": 10,
  "max": 100
},
"worker": {
  "im": 10,
```

## 1.5: 启动后端

```
[root@open-falcon ~]# cd $FALCON_HOME
[root@open-falcon open-falcon]# ./open-falcon start
```

#检查

```
[root@open-falcon open-falcon]# ./open-falcon check
```

## 2、部署前端

### 1.2: 前端代码下载

```
[root@open-falcon ~]# cd $FALCON_HOME
[root@open-falcon open-falcon]# git clone https://github.com/open-falcon/dashboard.git
```

### 1.3: 安装依赖包

```
[root@open-falcon open-falcon]# yum install -y python-virtualenv
[root@open-falcon open-falcon]# yum install -y python-devel
[root@open-falcon open-falcon]# yum install -y openldap-devel
[root@open-falcon open-falcon]# yum install -y mysql-devel
[root@open-falcon open-falcon]# yum groupinstall "Development tools" -y
```

### 1.6: 查看需要安装的模块

```
[root@open-falcon dashboard]# cat pip_requirements.txt
Flask==1.0
Flask-Babel==0.9
Jinja2>=2.10.1
Werkzeug==0.15.3
gunicorn==19.9.0
python-dateutil==2.2
requests==2.3.0
mysql-python
python-ldap
```



注意：3.0版本修改Werkzeug==0.15.4 才可以启动

```
[root@open-falcon dashboard]# vim pip_requirements.txt
Werkzeug==0.15.4
```

## 1.7: 安装模块

```
[root@open-falcon open-falcon]# cd dashboard/
[root@open-falcon dashboard]# virtualenv ./env
[root@open-falcon dashboard]# ./env/bin/pip install -r pip_requirements.txt
```

## 1.8: 修改配置

配置说明:

dashboard的配置文件的为: 'rrd/config.py', 根据实际情况修改:

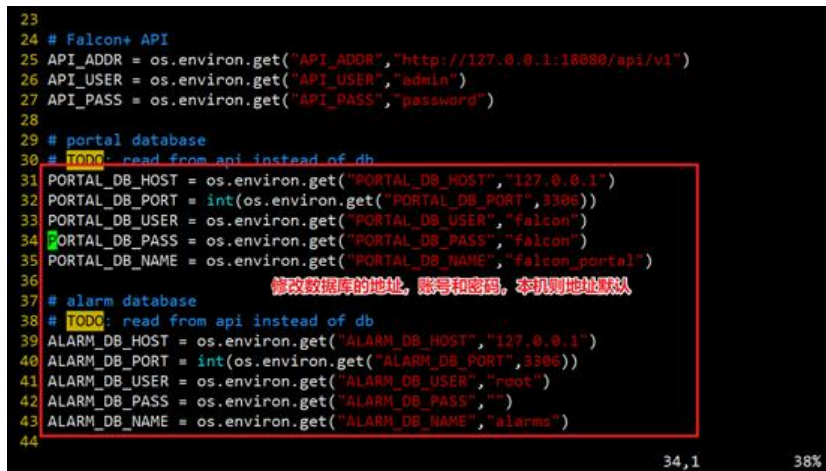
API\_ADDR 表示后端api组件的地址

```
API_ADDR = "http://127.0.0.1/api/v1"
```

根据实际情况, 修改PORTAL\_DB\_, 默认用户名为root, 默认密码为""

根据实际情况, 修改ALARM\_DB\_, 默认用户名为root, 默认密码为""

```
[root@open-falcon dashboard]# cp rrd/config.py{,_bak}
[root@open-falcon dashboard]# vim rrd/config.py
```



```
23
24 # Falcon+ API
25 API_ADDR = os.environ.get("API_ADDR", "http://127.0.0.1:18080/api/v1")
26 API_USER = os.environ.get("API_USER", "admin")
27 API_PASS = os.environ.get("API_PASS", "password")
28
29 # portal database
30 # TODO: read from api instead of db
31 PORTAL_DB_HOST = os.environ.get("PORTAL_DB_HOST", "127.0.0.1")
32 PORTAL_DB_PORT = int(os.environ.get("PORTAL_DB_PORT", 3306))
33 PORTAL_DB_USER = os.environ.get("PORTAL_DB_USER", "falcon")
34 PORTAL_DB_PASS = os.environ.get("PORTAL_DB_PASS", "falcon")
35 PORTAL_DB_NAME = os.environ.get("PORTAL_DB_NAME", "falcon_portal")
36
37 # alarm database
38 # TODO: read from api instead of db
39 ALARM_DB_HOST = os.environ.get("ALARM_DB_HOST", "127.0.0.1")
40 ALARM_DB_PORT = int(os.environ.get("ALARM_DB_PORT", 3306))
41 ALARM_DB_USER = os.environ.get("ALARM_DB_USER", "root")
42 ALARM_DB_PASS = os.environ.get("ALARM_DB_PASS", "")
43 ALARM_DB_NAME = os.environ.get("ALARM_DB_NAME", "alarms")
44
```

## 1.9: 启动服务

#生产环境启动

```
[root@open-falcon dashboard]# bash control start
```

#开发模式启动

```
[root@open-falcon dashboard]# ./env/bin/python wsgi.py
```

## 四、访问网站

默认端口8081

http://ip:8081



- 注意：dashbord默认没有创建任何账号包括管理账号，需要你通过页面进行注册账号

