

MacroHub软件快速安装

准备工作

- 硬件要求

```
CPU:      >= 4 core
Memory: >= 16GiB
Storage:
  OS:      >= 50GiB
  /opt/wfc: >= 50GiB  # work and database storage
  /var/lib/docker: >= 50GiB
```

- 系统环境

```
Linux >= Ubuntu 22.04
JDK >= 1.8 (Recommended version 1.8) (用于Omada SDN Controller)
Omada SDN Controller > 5.15.6.7
docker >= 24.0.7, build 24.0.7-0ubuntu2~22.04.1
docker-compose >= v2.10.0
docker-buildx >= 0.14.1
```

- 安装docker, docker-compose和buildx插件

```
sudo apt-get update
sudo apt-get install -y docker.io

# 安装docker-compose-v2插件
sudo apt-get install -y docker-compose-v2
sudo ln -s /usr/libexec/docker/cli-plugins/docker-compose /usr/bin/docker-
compose

# 安装docker-buildx插件
sudo apt-get install -y docker-buildx
sudo ln -s /usr/libexec/docker/cli-plugins/docker-buildx /usr/bin/docker-
buildx
```

- 请确认docker, docker-compose和buildx插件的版本是否符合要求

```
docker -v
docker-compose -v
docker buildx version
```

获取安装包

- TAR安装包
- 文件名：wfc-1.0.7-20250221.tar.gz
- DEB安装包
- 文件名：wfc-1.0.7-20250221.deb

说明：

- 版本号：1.0.7
- 构建日期：20250221

根据情况获取最新版本的软件安装包

安装和配置软件

TAR包安装方式

- 创建安装目录
- 全新安装时需要此步骤，如果升级安装，则保留原有目录

```
sudo mkdir /opt/wfc
```

- 解压软件包至安装目录

```
sudo tar xvfz wfc-1.0.7-20250221.tar.gz -C /opt/wfc
```

DEB包安装方式

- 安装软件包

```
sudo dpkg -i wfc-1.0.7-20250221.deb
```

- 安装后目录树

```
/opt/wfc                                # work root directory
├── bin                                # shell script and binary file
├── docker                             # docker compose work directory
│   ├── env                           # default docker environment file, will be copy to
docker compose work directory
│   ├── java                           # java image directory
│   ├── mysql                           # mysql database container directory
│   └── conf
```

```

├── conf.d
├── data
├── db
├── logs
├── tmp
├── nacos                # nacos container directory
│   ├── conf
│   └── logs
├── nginx                # nginx container directory
│   ├── conf
│   ├── conf.d
│   ├── html
│   │   └── dist
│   │       ├── sys    # system portal
│   │       └── u      # user portal
│   └── logs
├── redis                # redis container directroy
│   ├── conf
│   └── data
├── wfc                  # wfc container root directory
│   ├── auth            # auth root directory
│   │   └── jar
│   ├── gateway         # gateway root directory
│   │   └── jar
│   ├── modules         # modules root directory
│   │   ├── file
│   │   │   └── jar
│   │   ├── gen
│   │   │   └── jar
│   │   ├── job
│   │   │   └── jar
│   │   ├── payment
│   │   │   └── jar
│   │   ├── system
│   │   │   └── jar
│   │   └── user
│   │       └── jar
│   ├── upload          # file upload directory
│   └── visual
│       └── monitor
│           └── jar
└── systemd              # service daemon file, will be copy to
    /etc/systemd/system
        └── system

```

- 配置软件环境

```

cd /opt/wfc/bin
sudo ./wfcsetup.sh env 192.168.13.128 # 参数2是宿主机IP地址

```

- 修改docker环境参数

根据实际情况修改如下环境参数配置，说明：** WFC_SERVER_IP的地址已经在上面配置过程被替换 ** TZ替换为宿主机系统时区

```
cd /opt/wfc/docker
cat .env
WFC_CONFIG_DATABASE=wfc_config_db
WFC_SYSTEM_DATABASE=wfc_system_db
WFC_USER_DATABASE=wfc_user_db
MYSQL_SERVICE_NAME=wfc-mysql
MYSQL_SERVICE_PORT=3306
MYSQL_ROOT_PASSWORD=123456
MYSQL_SERVICE_USER=root
MYSQL_SERVICE_PASSWORD=123456
REDIS_PORT=6379
NACOS_PROFILE_NAME=prod
NACOS_NAME_SPACE=wfc-prod
NACOS_SERVER_NAME=wfc-nacos
WFC_SERVER_PORT=80
NACOS_SERVER_PORT=8848
GATEWAY_SERVER_PORT=8080
AUTH_SERVER_PORT=8081
WFC_FILE_PORT=9201
WFC_JOB_PORT=9203
WFC_PAYMENT_PORT=9204
WFC_SYSTEM_PORT=9205
WFC_USER_PORT=9206
WFC_SERVER_IP=192.168.13.128
NACOS_SERVER_IP=${WFC_SERVER_IP}
GATEWAY_SERVER_IP=${WFC_SERVER_IP}
RESTART_OPTION=on-failure:5
TZ=Asia/Shanghai #docker容器的时区
```

如果这些容器环境参数没有更改，可以直接运行如下命令设置环境和创建容器：

```
cd /opt/wfc/bin
sudo ./wfcsetup.sh all 192.168.13.128 # 参数2是宿主机IP地址
```

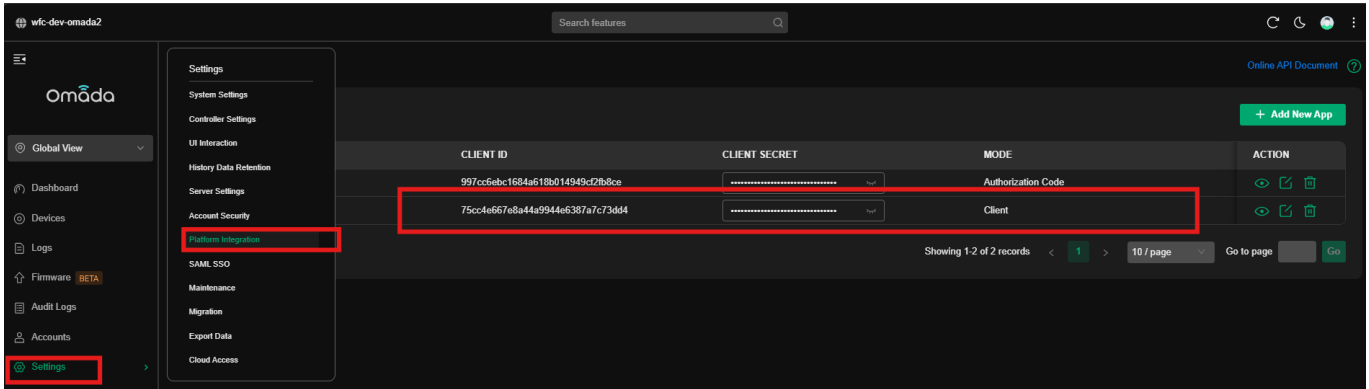
- 创建docker容器

如果已经运行'sudo ./wfcsetup.sh all 192.168.13.128'，如下命令可省略

```
cd /opt/wfc/bin
sudo ./wfcsetup.sh all
```

配置整合Omada平台的连接参数

- 从Omada服务中获取门户平台整合连接参数，如下图：



注意：连接的MODE采用Client， 具体的配置方式可参考Omada的相关文档

- 修改配置文件
 - 修改如下omada的配置项, 用于与Omada进行平台整合

```
cd /opt/wfc/docker/conf
cat application-common.yml

...

# Omada config
omada:
# web url or host exp: 'https://192.168.2.249:8043'
omada-url: 'https://aps1-omada-northbound.tplinkcloud.com'
omadac-id: 'c4decbf73f80094019d664506ec9b4d4'
client-id: '6110a77822234243a033e652d1e8a3ae'
client-secret: '651957da95f14f45b719876bb68bd6f5'

...
```

- 修改如下email的配置项, 用于发送邮箱验证码

```
cd /opt/wfc/docker/conf
cat application-common.yml

...

mail:
  enabled: true
  host: mail.agrandtech.com
  port: 25
  # Enable authentication
  auth: true
  # Sender, follows RFC-822 standard
  from: smtpext <smtpext@agrandtech.com>
  # User name
```

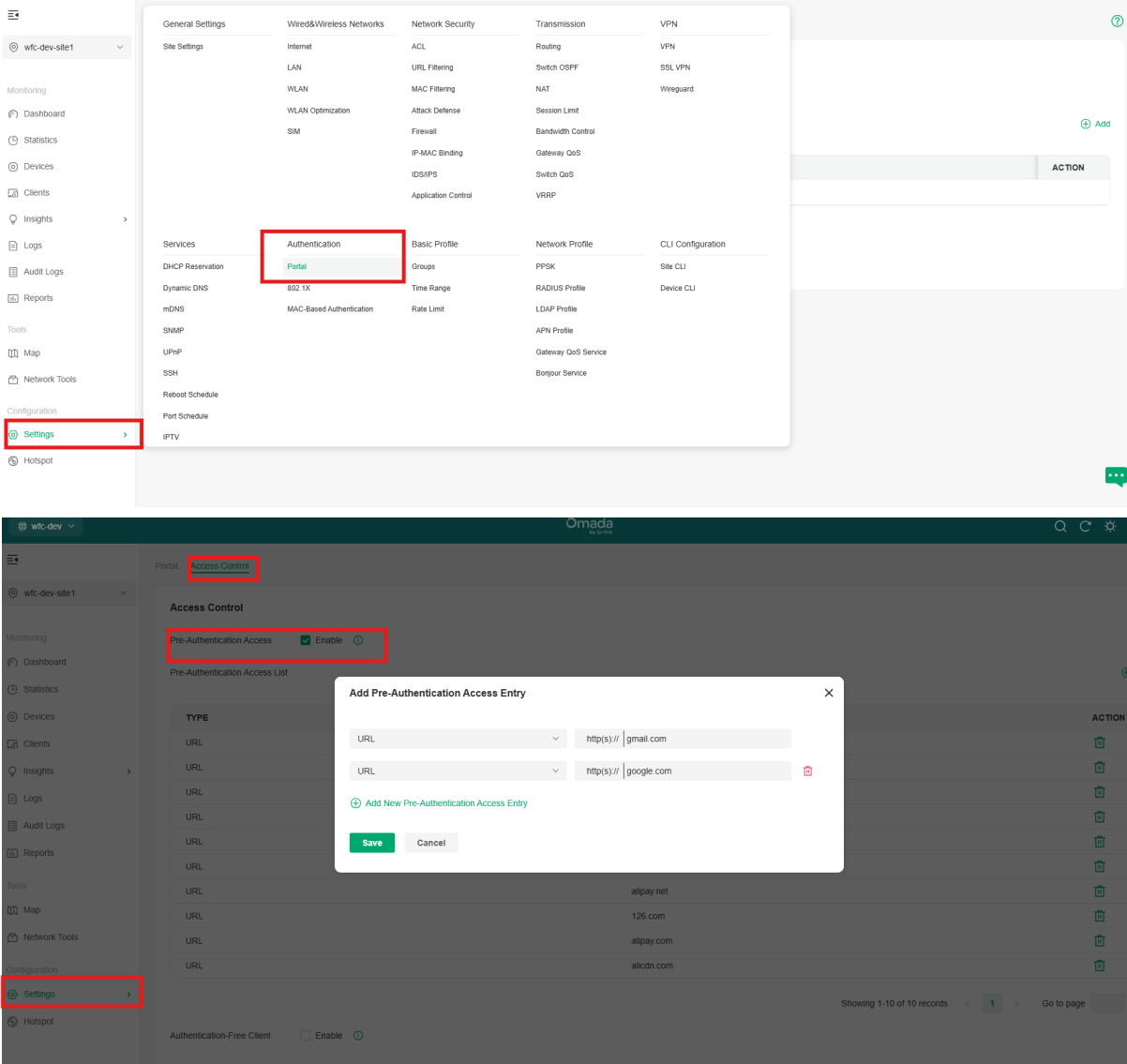
```
user: smtpext
# Password
pass: Smtpl23@agt
# Use TLS security connection
starttlsEnable: false
# Use SSL security connection
sslEnable: false
# Timeout value for sending mail, in milliseconds
timeout: 0
# Connection timeout value, in milliseconds
connectionTimeout: 0
```

...

- 在omada放行邮箱相关域名

为了支持用户在首次登录AP时，无网络权限情况下，能够获取邮箱的验证码，可以在Omada上对邮箱相关的域名进行放行操作，具体操作如下：

进入Omada后，选择需要配的站点，选择Settings->Authentication->Portal->Access Control->Pre-Authentication Access->勾选Enable，然后将该邮箱相关的域名增加到Pre-Authentication Access List, 注意需将所有访问邮箱涉及的所有域名都进行放行配置，下列图所示（放行的域名不一定完整）：



○ 修改支付配置项

```
cd /opt/wfc/docker/wfc/modules/payment/conf/
cat application.yml

...

# payment configuration,
# alipay configuration
alipay:
  appId: 9021000122699258
  privateKey:
MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBAkkggSlAgEAAoIBAQC7YAcIXV5GJlTWB0oBcrVG1zKx
HunznSjYXwTS2DM3YgloxzKye4apfcmBDVnoL93m4QrJTtbiHYvu4yJjJfJsQEY+Wo44IczhDyp
u+Il6JpQ0WylTs94/+3aaQ1wThsDaGs8edaFGteuw+1pUVl4nxF00fS1VoIR30riqAGFAXrPhcSA
6Y9c8HK1htxCeIICnPzjpc4s8ZjrjS7FoyHP2ZAay5fx1VEtah30E4RvycFfV/i2+k3JGF0Seev
dCx6Ufu+9Y/RaG4Sh84QuYRQUhAHUswlFDHPrRAnxHzt0zDk0kg3BrCT7vMtBbSOP9BdIhpVTaZm
4diQdNbLSl4hAgMBAAECggEAKb1Xc7aQ1KGfTlmj4xSxawlvImOXjAwbWC+6fflq91BgdjXBh17b
/Y+mvpfBymY3UUIaTAPC/HXGgT2ZoGqImTKsMyGrArgM0qm1M//EfdtV3L96rqxirTqduoS1WL0d
aWuWXdUoRH8r9K3ZTmY67TLh7FpiKevq9mI7fqs8/qgq5QAN4UPfb3Cq9hDNTU4l6+Bywdg1Kkw
J2Jc98SlrIdVl+1eCMAguelR0aNHtobyPBZ/KfwGRV00Q3QqjLVgL3z1bzuy1YhaZqPgxxd46vgW
wkyMiiv7u20yCetq4LFd+tcNDz0Yqk6rK9/S0oxbN8NM5DLkpwFh/433BoPS3QKBgQD/dsxNr2PY
```

```

dPLp6p0vw7H2AFMH8CkwTlRg+2f/0fgQF14rJwZgEU6fL5A2JvQeRRC5mCff2m5tDqrS321JNyH/
JIKJ+4J8I34GEo/kSCHL3RmyvJGQOQ/EpEyxpP8BV3ob6U03LR58DOGLJr8eTsr3Y8o9cAMukNCN
xvsd2LvybwKBgQC7xKkx4KIPBny6sC0tCh0kgIrY3h0Ci+ZsHQVESXKzeku0NkH/qrAG+S9pb1XC
zXdM2HrzkMeuz6tqi+2TY8nORyY2DZHNW7FqzkAiZ/N02AgGI1ALq1ZfJZke2MxgtBnxSsQ03gfX
P9opu/FsLH9a+s6edfHPvKD4kHx1LBfAbwKBgQDYvr6QmuKn4HHJLZGoau0uvw+ziHwp1AubTTLr
xgYmA02Qx0eNcQG2RfChQW0V2xtZnFMYhodyWCNvTsm2e0SLSTrIP3ByStq95DnZganLAGivsXX/
W6qZ0J0poCTryob79enSHKHktjksmjdfS6zBQRVYEvgtgJDcC03Rz21N8WQKBgQCCoJE4otb7bv7s
/ccEBe0eaNKvhue5xUwbYKEeXZ6PYAA0sh5+GCf34flq90qJiF17u4r10Y4pfk35hL1V9XEasYLS
zePmIuktrak0hpTNABus8MgRJjSjYsvAmwCsMf9uTx5qzSBVThWfPSH6qIY9LAj8Un9UjxzJb/Mq
i19GvWKBgQC0/KB+3robTqy9K0V4iEKJ4kmnXDRu7ecpMyc8k5HzpolIHcXLZHkKBK9pMCXBQ9Kwy
6BjRGNNuZOahS+tvnMcjHpwdhu90r7D51GXW/+PeFtphYTscFhHv1yuWbbojzhKf69pZnsVmyl1I
BqRgtbks2WNHxWo/peCg47TcLaQNMA==
        publicKey:
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAhBZoE6p1TKpd/WsAJRTA2D6yUTzqLmZ7
3scxbPxxKwPIg2ys2Ap21u/lVmqBW09Wf2Z7wUwr/6BP1PQqVpg0GAqU3yjjZ0LGfBitzMBMGr1
OLqJQ3PKgnUNmnOdLR4F1HCAjKtuyxjHkxBTHg/L4punk4lkFJQD8YvzFrYalS6WYw5j1/oq080N
bcjPirPCRLsU7MRraQuUqSTdPwm2iHkroKwCEbGv+RtNJCf0kKNQVtRFFgPGMHU4nQ620gdfzsGJ
sNFKVJnD5SjeDwniBAbaGaVF4rEfKJPe6+XrcMg3h+9Fz+A2IU1KCCoth2neEZEaU1t0KBGxXIuo
4efi/QIDAQAB
        appCertPath: \u5E94\u7528\u516C\u94A5\u8BC1\u4E66
        aliPayCertPath: \u652F\u4ED8\u5B9D\u516C\u94A5\u8BC1\u4E66
        aliPayRootCertPath: \u652F\u4ED8\u5B9D\u6839\u8BC1\u4E66
        serverUrl: https://openapi-sandbox.dl.alipaydev.com/gateway.do
        domain: http://192.168.2.249
        testDomain: http://129.204.171.210:8085

# wxpay configuration
wxpay:
    appId:
    appSecret:
    mchId:
    partnerKey:
    certPath:
    domain: http://192.168.2.249/u

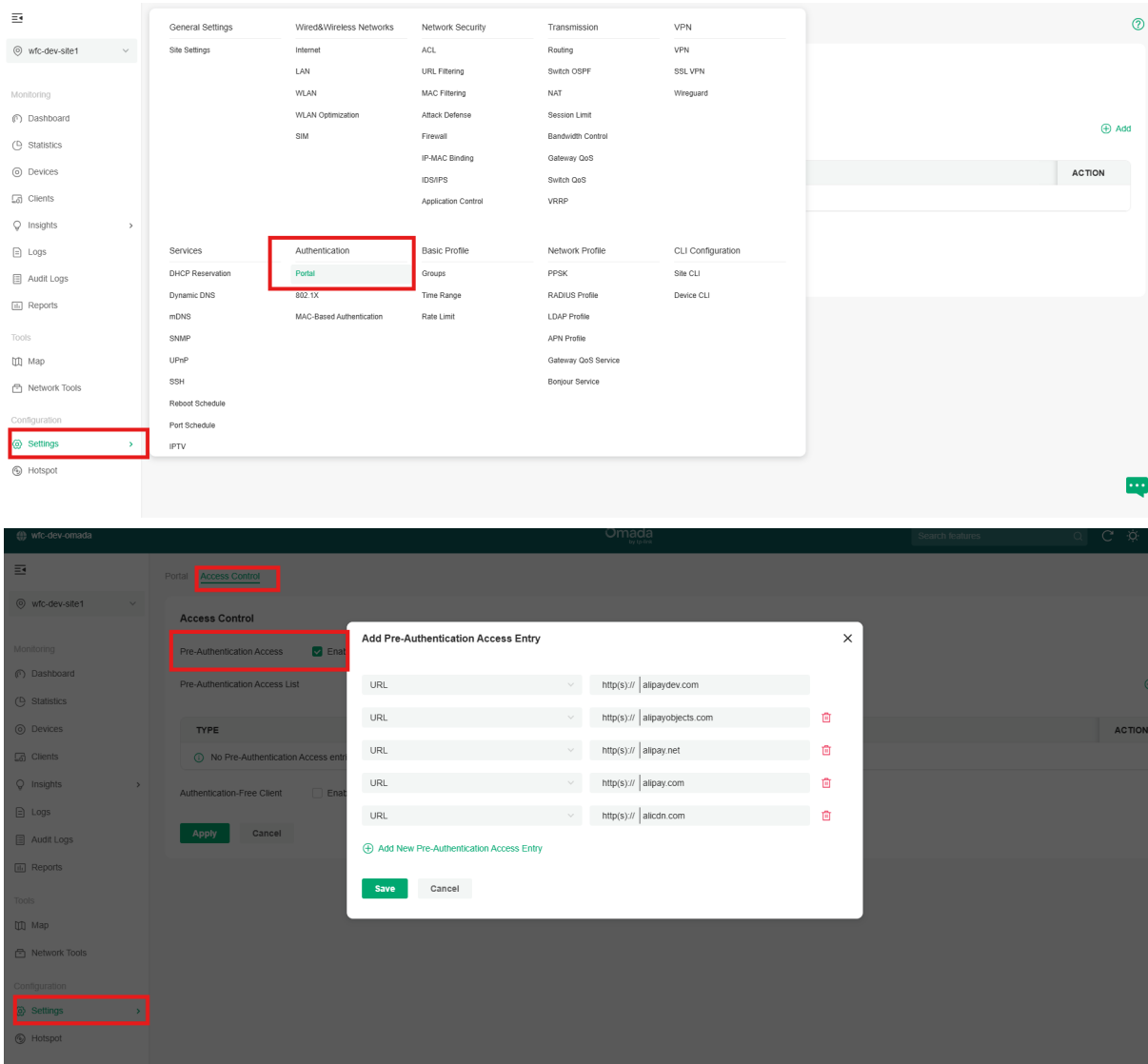
...

```

- 在omada放行支付相关域名

为了支持用户在首次登录AP时，无网络权限情况下，能够购买套餐或者余额充值正常进行支付，需要在Omada上对支付相关的域名进行放行操作，具体操作如下：

进入Omada后，选择需要配的站点，选择Settings->Authentication->Portal->Access Control->Pre-Authentication Access->勾选Enable，然后将该支付方式相关的域名增加到Pre-Authentication Access List, 注意需将所有支付涉及的所有域名都进行放行配置，下列图所示（放行的域名不一定完整）：



○ 邮件内容自定义

客户平台邮件模板位于目录/opt/wfc/docker/conf/templates/system/ 用户平台邮件模板位于目录/opt/wfc/docker/conf/templates/user/ 验证码模板：mail.html 流量提醒模板：trafficReminder.html 余额提醒模板：balanceReminder.html 发票模板: invoice.html 修改邮件模板即可，修改后直接生效，模板采用Thymeleaf引擎，支持html语法，可自定义文本内容和样式。除了占位符变量如等，其余内容均可自定义修改

○ 修改套餐流量不足或余额不足配置项

```
cd /opt/wfc/docker/wfc/modules/user/conf/
cat application.yml

...

# 提醒配置
reminder:
  # 是否开启流量提醒
  trafficEnable: true
  # 流量提醒阈值，单位：百分比，如剩余小于10%会触发邮件提醒
  trafficThreshold: 10
  # 是否开启余额提醒
```

```
balanceEnable: true
# 余额提醒阈值，单位：跟随系统，如余额小于5元会触发邮件提醒
balanceThreshold: 5
# 流量提醒邮件标题
trafficTitle: 'Traffic Reminder'
# 余额提醒邮件标题
balanceTitle: 'Balance Reminder'

...
```

- 统一修改系统名

修改环境变量文件

```
cd /opt/wfc/docker
cat .env

...
SYSTEM_TITLE=MacroHub
...
```

修改后需要重新执行并重启

```
cd /opt/wfc/bin
sudo ./wfcsetup.sh all 192.168.13.128 # 参数2是宿主机IP地址
```

或者直接修改国际化文件无需重启

```
cd /opt/wfc/docker/nginx/html/dist/u/langs
cat zh-cn.js
cat en-us.js

...

  system: {
    title: 'MacroHub'
  }

...
```

- 启动/查看/停止/版本软件

```
cd /opt/wfc/bin
sudo ./wfccontrol.sh start/restart/status/stop/version
```

通过浏览器访问客户平台和用户平台

- 客户平台门户地址：

```
http://192.168.13.128/sys
```

- 用户平台门户地址：

```
http://192.168.13.128/u
```