SWAN编译以及测试部署手册

1. SWAN编译流程

1.1 编译环境准备

SWAN采用docker环境编译，请先在编译环境安装docker。

- docker: 27.1.1+

`docker`安装完成后，在编译环境登录私有镜像库`registry.amiasys.com`，账号是 `amia` ，密码是`2022@Amiasys`。

1.2 开始编译

1.2.1 编译SWAN服务

```
make build-first
```

> **注意**：如果编译报错提示`docker`没有执行权限`permission denied`，可执行下列步骤，**不可以直接`sudo make build-first`进行编译。**
>
> ```
> # 新增docker用户组
> sudo groupadd docker
>
> # 将当前用户添加到docker用户组
> sudo gpasswd -a $(whoami) docker
>
> # 执行命令修改生效或者登录登出
> newgrp docker
> ```
>
> 如果后续继续报错`ERROR: open /home/asn/.docker/buildx/current: permission denied`，执行`sudo chown -R $(whoami) ~/.docker`即可为当前操作添加权限。

1.2.2 编译SWAN-Manager镜像

1、修改项目`docker/`目录下`swan-manager.dockefile`，更新依赖的`registry.amiasys.com/asnc`版本。

2、修改项目Makefile文件第56行`SWAN_MANAGER_VERSION`变量，确定编译的SWAN版本。

3、在项目路径下执行

```
make swan-manager
```

> 如果报如下所示错误，可执行`sudo docker pull registry.amiasys.com/asnc:v25.0.22`先下载镜像。

```
rose@rose:~/a/0228/SWAN.25$ make swan-manager
[+] Building 4.7s (2/2) FINISHED                                                                                    docker:default
 => [internal] load build definition from swan-manager.dockerfile                                                           0.3s
 => => transferring dockerfile: 332B                                                                                        0.0s
 => ERROR [internal] load metadata for registry.amiasys.com/asnc:v25.0.22                                                   2.9s
------
 > [internal] load metadata for registry.amiasys.com/asnc:v25.0.22:
------
swan-manager.dockerfile:3
--------------------
   1 |     # Copyright 2025 Amiasys Corporation and/or its affiliates. All rights reserved.
   2 |
   3 | >>> FROM registry.amiasys.com/asnc:v25.0.22
   4 |     WORKDIR /asn
   5 |
--------------------
ERROR: failed to solve: registry.amiasys.com/asnc:v25.0.22: failed to resolve source metadata for registry.amiasys.com/asnc:v25.0.22
: unexpected status from HEAD request to https://registry.amiasys.com/v2/asnc/manifests/v25.0.22: 401 Unauthorized
make: *** [Makefile:58: swan-manager] Error 1
```

1.2.3 编译SWAN-Service node镜像

1、修改项目`docker/`目录下`swan-servicenode.dockerfile`，更新依赖的`registry.amiasys.com/asnsn`版本。

2、修改项目Makefile文件第56行`SWAN_MANAGER_VERSION`变量，确定编译的SWAN版本。（当前`Manager`与`Servicenode`版本号要求统一，编译`Manager`时修改过了，此处可以不用修改了。

3、在项目路径下执行

```
make swan-servicenode
```

> 如遇manager类似报错，可按相同方法解决。

1.3 更新安装资源

1、如果`SWAN Manager`的任何一个依赖`config`文件发生变化，或者部署的`swan-manager.yml`脚本变化，或者`Manager`命令行更新，请及时更新安装资源。更新方式如下：

- 登录`47.93.161.66`机器
- 进入到`/home/nginx/download/swan/manager/config`文件夹，找到对应的`config`文件进行更新
- 进入到`/home/nginx/download/swan/manager`文件夹，找到对应的`swan-manager.yml`进行更新
- 进入到`/home/nginx/download/swan/manager`文件夹，如果是替换命令行工具`mngcli`，未修改版本号，则找到对应版本目录，替换`mngcli`即可。如果是命令行工具版本升级，则创建对应版本文件夹（如`v25.0.0`），然后将对应`mngcli`工具放入文件夹下即可

2、如果`SWAN Server`的任何一个依赖`config`文件发生变化，或者部署的`swan-servicenode.yml`脚本变化，请及时更新安装资源。更新方式如下：

- 登录`47.93.161.66`机器
- 进入到`/home/nginx/download/swan/servicenode/config`文件夹，找到对应的`config`文件进行更新
- 进入到`/home/nginx/download/swan/servicenode`文件夹，找到对应的`swan-servicenode.yml`进行更新

2. SWAN Manager安装流程

SWAN Manager版本：

版本号	镜像ID	版本内容
v25.0.0	414ba11e5276	- 虚拟网络管理 - 虚拟网络组管理 - 节点管理 - 用户管理 - 用户组管理 - 日志审计功能 - 节点流量统计功能
v25.0.1	6730dc9a4fb9	- 优化服务 - 修复节点管理、网络组管理bug - 框架新增token校验功能：`asn.conf`配置文件新增`token_secret`字段
v25.0.2	cf7c4a258eb3	- config文件修改：移除DNS配置，通过gRPC接口实现；新增license服务访问地址配置 - 支持创建虚拟网络时配置DNS功能，DNS功能支持配置内网DNS服务器，用于解析内网域名 - 新增License服务交互功能。访问License服务获取许可证书并进行校验。 更新时请注意重新获取manager的配置文件`swan.conf`
v25.0.4	bcec2b028bb5	- 修复网页端授权问题
v25.0.5	8a8950d39c67	- 修复用户分页错误问题 - 统一influxdb数据库名（日志审计数据会重置）
v25.0.6	720d63b95b1a	- 修复用户超时退出后manager侧仍显示在线 - 修复节点不在线时没有基础数据问题 - 前端修复若干问题
v25.0.7	329c34acab19	- 总览页面饼图展示优化，跳转审计页面逻辑错误修复，网络表单联动等bug修复，字段调整等 - 客户端和WebManager区分不同的token有效期
v25.0.8	1d05ba550925	- 修复用户列表累计流量显示 - 网络引导流程中增加dns和domains配置 - 修复流量为负数的问题 - 修复servicenode状态问题 - 修复webmanager上用户流量数据计算不准确问题 - 前端修改若干禅道问题
v25.0.9	23277868d58b	- 修复用户登录设备解绑后导致没有计算该解绑设备流量的问题 - 修复webmanager上用户流量数据计算不准确问题 - manager重启后，network仍然展示存在可用节点(#1298) - client上线时，校验入参network和servicenode所属关系(#1239) - 引导页不可用加密算法隐藏，重要数据样式修改，节点基础数据样式调整

版本号	镜像ID	版本内容
v25.0.10	ad80378fcae0	- 修改短期用户删除未删除用户设备的问题 - 修改问题#1315 #1316 - 节点展开行内容修改+样式调整，用户剩余流量计算逻辑修改+柱状图显示修改
v25.0.11	fb093023cff4	- 优化web端查看用户流量 - 初始化验证码错误提示修改 - 用户列表累计流量在展开流量控制中显示 - 用户流量控制柱状图不显示数字，鼠标移入才显示数值 - 修复日志审计记录错误 - 修复#1344
v25.0.12	efe81fd6d16d	- 修复删除用户时未删除用户个人的策略问题 - 修复当重启asnsn.service后客户端连接状态未重置问题 #1353 - token失效提示修改，节点展开全部折线图tooltip显示优化
v25.0.13	85397aeac5a8	- 优化web退出登录问题 - 用户设备列表增加在线状态 - 增加客户端心跳接口，记录客户端用户和设备是否在线 - 优化ldap获取用户列表逻辑 - 修复同一台设备用不同的激活码二次激活，激活码对应的manager上2个服务节点状态异常(#1387) - 用户设备列表显示在线标签 - 用户流量控制剩余流量优化 - 中英文提示对照检查补充 - 接口请求默认不缓存
v25.0.14	1246514823b6	- 后端增加心跳api；修复创建用户组问题# 1363；修改bps单位;修复多个激活码激活servicenode问题;修复mfa登录token失效问题;优化用户组排序 - web端增加心跳；中英文检查补充；日志总览饼图显示优化数据为0的情况；用户柱状图显示过长设备名称优化
v25.0.15	b9a2d5051d90	- 优化心跳机制，记录每台设备的在线状态 - 修复用户列表统计状态问题 #1425 - 优化日志审计默认OperationResult值为失败 - 优化mfa登录逻辑 - 修改已激活设备用错误激活码再次激活，设备状态异常，但资源还在刷新的问题(#1407) - 修复服务节点重启，客户端仍处于connect状态，无法下线的问题
v25.0.16	804c91935b86	- 节点bps单位去除/s并且优化y轴刻度值间隔 - 节点管理员越权解决
v25.0.17	89108296cec9	- 修复同一节点重复激活，连接显示错误(#1432) - 修复IP分配异常造成的网络异常(#1427) - 客户端轮询服务节点状态接口新增服务节点上线时间返回(#1355)

版本号	镜像ID	版本内容
v25.0.18	9d7dfe9fe8b6	- 优化设备激活逻辑处理，同一设备用不同激活码二次激活时，第一次激活的服务节点状态调整为异常 - 增加客户端异常日志打印 - 增加客户端服务断开的审计日志（#1433)
v25.0.19	426cf352650f	- 解决退出登陆后，页签存在残留显示问题
v25.0.20	91d988422727	- 修复不同client分配接入同一网络被分配到相同ip - 服务节点不可达之后，主动断开与所有客户端的连接
v25.0.21	a40bb4428f41	- 增加异常退出servicenode记录审计日志 - 解决登录后在浏览器输入login页面地址报错问题
v25.0.22	782e764a7819	- 优化部分日志记录语句 - 优化展示：用户展开行流量统计柱状图提示框展示单位 - 日志模块增加system字段并添加相应数据展示
v25.0.23	38797cb5ca5a	- 修复检查用户是否是普通用户
v25.0.24	78fd19229c17	- 日志总览事件用户源SYSTEM颜色更改 - 虚拟网络新增/编辑弹窗和初始化引导界面服务网段增加提示

2.1 准备工作

1、请在安装SWAN Manager的机器上提前准备好`docker`环境，并修改`/etc/docker/daemon.json`文件（如果不存在，请手动创建），增加以下内容：

```
{
  "insecure-registries": ["182.92.162.123:18080"]
}
```

2、`docker`安装完成后，在`Manager`服务器上登录2个私有镜像库。

（1）登录私有库`182.92.162.123:18080`，账号`admin`，密码`Harbor12345`。

（2）登录到`registry.amiasys.com`，账号是`amia`，密码是`2022@Amiasys`。

3、为了避免容器环境部署时，`docker`自动分配的子网与宿主机通信子网冲突，请提前创建一个容器网络`swan_network`。

```
docker network create --driver bridge --subnet <子网地址> --gateway <子网网关> swan_network

# 例如：docker network create --driver bridge --subnet 172.25.0.0/16 --gateway 172.25.0.1
swan_network
```

2.2 安装步骤

1、在服务器上新建安装目录

```
mkdir -p swan/manager
```

2、进入到安装目录，并拉取`manager`配置

```
cd swan/manager

# 命令拉取config文件夹
wget -r -np -nH --cut-dirs=2 -R index.html.tmp -R index.html http://pkg-
d.amianetworks.com.cn:19067/swan/manager/config/

# 命令行拉去swan-manager.yml文件
wget http://pkg-d.amianetworks.com.cn:19067/swan/manager/swan-manager.yml
```

3、请根据真实的安装部署环境定义网络拓扑文件`network1-topology.json`（测试环境可默认）。

4、修改`swan-manager.yml`，更新`sapphire-iam`和`swan-manager.yml`的镜像版本号。

```
# Copyright 2024 Amiasys Corporation and/or its affiliates. All rights reserved.

services:
  asn-mdb:
    container_name: asn-mdb
    image: 182.92.162.123:18080/swan/mongo:7.0
    restart: always   # auto restart the container if it fails
    ulimits:
      nofile: 100000
    environment:
      MONGO_INITDB_ROOT_USERNAME: amia  # db root username,
      MONGO_INITDB_ROOT_PASSWORD: 2022  # db root user password
    ports:
      - "27017:27017"  # port forwarding (localPort:containerPort)
    volumes:
      - mongodb_data:/data/db  # data volumes, (localDirectory:containerDirectory)
      - /etc/localtime:/etc/localtime
    command: --bind_ip_all --auth
    networks:
      - swan_network
  asn-idb:
    image: 182.92.162.123:18080/swan/influxdb:1.11.8
    container_name: asn-idb
    ports:
      - "8086:8086"
    volumes:
      - influxdb_data:/var/lib/influxdb
      - /etc/localtime:/etc/localtime
    environment:
      INFLUXDB_DB: asn
      INFLUXDB_ADMIN_USER: amia
```

```
      INFLUXDB_ADMIN_PASSWORD: 2022
      INFLUXDB_USER: amia
      INFLUXDB_USER_PASSWORD: 2022
    networks:
      - swan_network
  sapphire-iam:
    image: registry.amiasys.com/sapphire.iam:v25.0.11 # modify images version when it
updated
    container_name: sapphire-iam
    privileged: true
    restart: always
    ports:
      - "17930:17930"
      - "17931:17931"
    volumes:
      - ldap_slap:/etc/ldap/slapd.d/
      - ldap_data:/var/lib/ldap/
      - ./config/:/usr/local/sapphire/
      - /etc/localtime:/etc/localtime
    networks:
      - swan_network
  swan-manager:
    image: 182.92.162.123:18080/swan/swan-manager:v25.0.23  # modify images version when it
updated
    container_name: swan-manager
    restart: always   # auto restart the container if it fails
    network_mode: host
    depends_on:
      - "asn-mdb"
      - "asn-idb"
      - "sapphire-iam"
    volumes:
      - ./config/:/etc/asnc/config/ # config file
      - ./log/:/var/log/asnc/     # log files
      - ./cert/:/etc/asnc/cert/
      - /etc/localtime:/etc/localtime

volumes:
  mongodb_data:
    driver: local
  influxdb_data:
    driver: local
  ldap_slap:
    driver: local
  ldap_data:
    driver: local

networks:
  swan_network:
    external: true
```

5、进入到`manager/config`文件夹，修改配置文件

- 修改`asn.conf`文件，具体修改内容如下：

```
# Copyright 2025 Amiasys Corporation and/or its affiliates. All rights reserved.

log:
  prefix: "asn"
  demo: true
  api_log:
    filename: "api.log"
    level: "info"
  runtime_log:
    filename: "runtime.log"
    level: "info"
  entity_log:
    filename: "entity.log"
    level: "info"
  perf_log:
    filename: "perf.log"
    level: "info"
db:
  mongodb:
    host: [docker0或本机地址]
    port: 27017
    database_name: asn
    username: amia
    password: 2022
  influxdbv1:
    host: [docker0或本机地址]
    port: 8086
    database_name: asn
    username: amia
    password: 2022
iam:
  provider: sapphire # local/sapphire
  host: localhost # ignore if using local
  port: 17930 # ignore if using local
  tls: false # ignore if using local
  ca_cert: /etc/asnc/cert/ca-cert # ignore if using local or not using TLS
  cert_pem: /etc/asnc/cert/cert-pem # ignore if using local or not using TLS
  key_pem: /etc/asnc/cert/key-pem # ignore if using local or not using TLS
grpc:
  port: 50051
restful:
  port: 58080
network:
  id: network1  # network name
  topo_file: /etc/asnc/config/network1-topology.json  # network topology description
file path
  token_secret: wjEEOnD1XOvtwG18yqt+JxEtb1JnO8raXOg+TUIbJzA=
servicenode:
  keepalive: 3  # service node heartbeat rate, n second / heartbeat

service:
```

```
swan:
  auto_start: false
  version:
    min: v25.0.0
    max: v25.0.100
  db:
    mongodb:
      host: [docker0或本机地址]
      port: 27017
      database_name: swan
      username: amia
      password: 2022
    influxdbv1:
      host: [docker0或本机地址]
      port: 8086
      database_name: swan
      username: amia
      password: 2022
```

5、启动`SWAN Manager`容器

```
cd swan/manager
docker compose -f swan-manager.yml up -d
```

6、查看SWAN Manager容器运行状态

```
docker ps -l
```

> 如果`STATUS`显示非`Up`状态，联系安装部署人员进行错误排查。

7、`SWAN Manager`进行更新时，先删除原来的`swan-manager`容器，然后修改容器镜像版本(**如果版本号未变，但镜像更新过，只需要`docker rm`删除本地原来的镜像即可**)，再重新启动容器

```
# 1.删除原来的swan-manager容器
docker compose -f swan-manager.yml down

# 2.修改swan-manager.yml文件，将swan-manager的镜像版本修改为需要安装的版本

# 3.重新安装镜像
docker compose -f swan-manager.yml up -d
```

8、卸载SWAN Manager

(1) 仅卸载容器，保留历史数据，删除容器即可

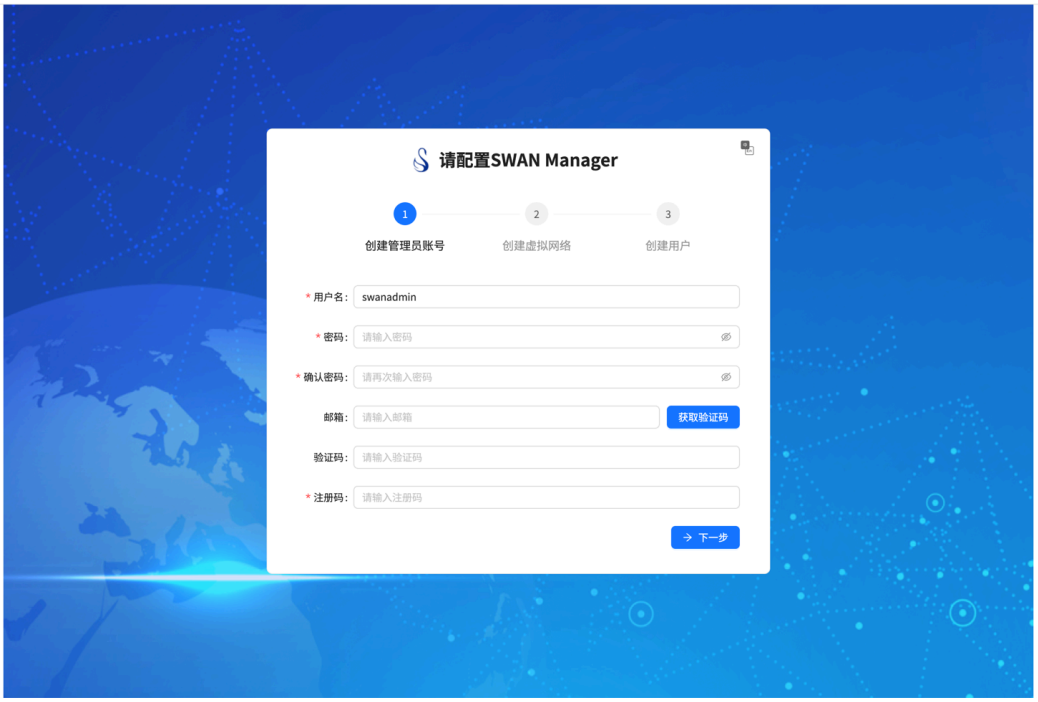
```
docker stop swan-manager
docker compose -f swan-manager.yml down
```

(2) 纯净卸载，不保留历史数据

```
docker stop swan-manager
docker compose -f swan-manager.yml down
# 删除数据卷
docker volume rm manager_influxdb_data manager_ldap_data manager_ldap_slap
# 删除安装目录和安装数据
sudo rm -rf swan/manager
```

2.3 SWAN Manager Web管理平台

浏览器访问 `http://<ip>:58080/service/swan`，按照引导程序进行配置，临时注册码：`4VMywc98SwUS`（一次性注册码，有效期1个月）



2.4 SWAN Manager命令行

2.4.1 命令行安装

```
wget http://pkg-d.amianetworks.com.cn:19067/swan/manager/v25.0.1/mngcli

chmod +x mngcli

sudo mv mngcli /usr/bin/
```

2.4.2 快速入门

```
# 创建Manager管理员账号
mngcli admin create -u <账号名> -p <密码> -a 0 -e <邮箱地址> -f true
# example: mngcli admin create -u swanadmin -p admin@1234 -a 0 -e xxx@qq.com -f true

# 登录Manager管理员账号
mngcli login -u <账号名> -p <密码>
# example: mngcli login -u swanadmin -p admin@1234
```

```
# 创建虚拟网络
mngcli network create -n <虚拟网络名称> -s <虚拟网络的代理网段>
# example: mngcli network create -n newnet -s 1.2.3.0/24

# 查看创建的虚拟网络
mngcli network list

# 创建虚拟网络组
mngcli network-group create -g <虚拟网络组名称> -n <加入网络组的虚拟网络ID>
# example: mngcli network-group create -g newnetgroup -n 181580

# 查看虚拟网络组
mngcli network-group list

# 创建SWAN节点
mngcli service-node create --name <节点名称> --address <节点的访问地址> --port-start <节点访问端口左边界> --port-end <节点访问端口右边界>  --joined-network <要加入的虚拟网络ID>
# example: mngcli service-node create --name servicenode --address 172.17.17.171 --port-start 21000 --port-end 22000  --joined-network 199709

# 查看创建的SWAN节点
mngcli service-node list

# 上线SWAN节点
mngcli service-node toggle -t on -c <节点激活码>
# example: mngcli service-node toggle -t on -c LXrmGB

# 下线SWAN节点
mngcli service-node toggle -t off -c <节点激活码>
# example: mngcli service-node toggle -t off -c LXrmGB

# 创建用户
mngcli user create -u <用户名> -p <用户密码>
# example: mngcli user create -u rose -p P@ssw0rd

# 查看用户
mngcli user list

# 创建用户组
mngcli user-group create -g <用户组名称> -u <加入用户组的用户名称> -n <加入用户组的网络组ID>
# example: mngcli user-group create -g roseuser -u rose -n 156581

# 查看用户组
mngcli user-group list
```

3. SWAN Server安装流程

3.1 主机部署（推荐）

版本	md5sum	ASN框架
v25.0.1	0efef7bbaa8d985aa4bd61f790d7aa3d	v25.0.21
v25.0.3	6cb3ae5d9082f3f9e533314cabebf22d	v25.0.21
v25.0.4	8a2609ec4d9459bf8e302c270ce50772	v25.0.21
v25.0.5	92941549227e5944acb3fdb6bfdf67bc	v25.0.22
v25.0.6	17f7445ab0bea430be47bd253a340d33	v25.0.22

3.2.1 安装步骤

1、创建安装目录

```
mkdir -p swan/servicenode
```

2、下载SWAN Servicenode deb安装包

```
cd swan/servicenode
wget http://pkg-d.amianetworks.com.cn:19067/swan/servicenode/asnsn_25.0.22_amd64.deb
```

3、执行安装命令

```
sudo dpkg -i asnsn_25.0.22_amd64.deb
```

4、完成后，下载配置文件

```
cd /etc/asnsn/config
# dpkg安装deb时，会在此目录下放置默认的asn.conf文件，建议删除后，重新下载asn.conf，否则wget的文件会议
asn.conf.1的形式放置在此目录下
sudo wget http://pkg-d.amianetworks.com.cn:19067/swan/servicenode/config/asn.conf
sudo wget http://pkg-d.amianetworks.com.cn:19067/swan/servicenode/config/swan.conf

cd /etc/asnsn/service
sudo wget http://pkg-d.amianetworks.com.cn:19067/swan/servicenode/v25.0.6/swan.so
```

5、修改配置文件

```
cd /etc/asnsn/config
sudo vim asn.conf

# asn.conf
controller:
  ip: localhost #[Manager IP地址]
```

```
  port: 50051
  retry_interval: 5
general:
  cli_port: 50052
  id:
  mode: cluster
  network_capacity: 1024
  network_path: network1.node2.switch1 # 节点在network1-topology.json中定义的位置
  node_name: switch1
  type: server
log:
  api_log:
    filename: api.log
    level: info
  entity_log:
    filename: entity.log
    level: info
  perf_log:
    filename: perf.log
    level: info
  prefix: asn
  runtime_log:
    filename: runtime.log
    level: info
netif:
  control: eno2      #[本机控制面网卡]
  data: eno2         #[本机数据面网卡]
  management: eno2   #[本机管理面网卡]
service:
  config_timeout: 20
tsdb:
  ip: localhost #[Manager IP地址]
  name: asn-dev
  password: 2022
  port: 8086
  type: influxdbv1
  username: amia


#----------------------------
# 注意：yaml文件中冒号后的值必须和冒号中间保留一个空格！
sudo vim swan.conf

# swan.conf
# Copyright 2025 Amiasys Corporation and/or its affiliates. All rights reserved.

# Activation code of a service node.
# Allocated when a service node is created by swan manager.
activationCode: #[由Manager创建的servicenode生成的激活码]

# The actual physical address of the service node is used to set up a data tunnel with other
service nodes.
privateAddress: #[用于和其他Servicenode建立隧道的IP]
```

5、重启`asnsn.service`

```
sudo systemctl start asnsn.service
```

6、查看启动状态

```
sudo systemctl status asnsn.service
```

7、查询日志

```
# 查询asnsn日志
tail -f /var/log/asnsn/asn_runtime.log

# 查询swan日志
tail -f /var/log/asnsn/swan_runtime.log
```

3.2 容器化部署（不推荐）

SWAN Server版本：

版本号	镜像ID	版本内容
v25.0.0	0cb55c1d2fde	- 数据隧道

3.2.1 准备工作

1、请在安装SWAN Server的机器上提前准备好`docker`环境，并修改`/etc/docker/daemon.json`文件（如果不存在，请手动创建），增加以下内容：

```
{
  "insecure-registries": ["182.92.162.123:18080"]
}
```

2、登录私有镜像库`182.92.162.123:18080`

```
docker login 182.92.162.123:18080
#账号
admin
#密码
Harbor12345
```

3.2.2 安装步骤

1、在服务器上新建安装目录

```
mkdir -p swan/servicenode
```

2、进入到安装目录，并拉取`manager`配置

```
cd swan/servicenode

# 命令拉取config文件夹
wget -r -np -nH --cut-dirs=2 -R index.html.tmp -R index.html http://pkg-
d.amianetworks.com.cn:19067/swan/servicenode/config/

# 命令行拉去swan-manager.yml文件
wget http://pkg-d.amianetworks.com.cn:19067/swan/servicenode/swan-servicenode.yml
```

3、修改`swan-servicenode.yml`，更新`swan-servicenode.yml`的镜像版本号。

```
# Copyright 2024 Amiasys Corporation and/or its affiliates. All rights reserved.

services:
  swan-servicenode:
    image: 182.92.162.123:18080/swan/swan-servicenode:v0.0.1 # 确认要安装的swan-servicenode镜像
版本
    container_name: swan-servicenode
    restart: always   # auto restart the container if it fails
    network_mode: host
    privileged: true
    volumes:
      - ./config/:/asn/config/ # config file
      - ./log/:/asn/log/ # log files
      - /etc/localtime:/etc/localtime
```

4、进入`servicenode/config`文件夹，修改`asn.conf`文件

```
# Copyright 2025 Amiasys Corporation and/or its affiliates. All rights reserved.

log:
  prefix: "asn"
  api_log:
    filename: "api.log"
    level: "info"
  runtime_log:
    filename: "runtime.log"
    level: "info"
  entity_log:
      filename: "entity.log"
      level: "info"
  perf_log:
      filename: "perf.log"
      level: "info"
general:
    mode: cluster
    id: ""
    network_path: network1.node2.switch1
    node_name: switch1
```

```
    type: server
    network_capacity: 1024
    cli_port: 50052
controller:
    ip: 172.17.0.1  # 请确认ASNC IP地址正确
    port: 50051
    retry_interval: 5
    token_secret: "wjEEOnD1XOvtwG18yqt+JxEtb1JnO8raXOg+TUIbJzA="
tsdb:
    type: influxdbv1
    name: asn-dev
    ip: 172.17.0.1  # 请确认时序数据库地址正确
    port: 8086
    username: amia
    password: 2022
service:
    config_timeout: 20
netif:
    data: eth0          # 请确认数据面网卡存在
    control: eth0       # 请确认控制面网卡存在
    management: eth0    # 请确认管理面网卡存在
```

5、进入`swan/servicenode`文件夹，修改`swan.conf`文件

```
# Copyright 2025 Amiasys Corporation and/or its affiliates. All rights reserved.

# Activation code of a service node.
# Allocated when a service node is created by swan manager.
activationCode:  # 请填写Manager创建的Servicenode对应生成的激活码

# The actual physical address of the service node is used to set up a data tunnel with other service nodes.
privateAddress:  # 请填写Servicenode用于与其他Servicenode建立数据隧道的物理IP，可以与访问地址相同
```

6、启动`SWAN Servicenode`容器

```
cd swan/servicenode
docker compose -f swan-servicenode.yml up -d
```

6、查看`SWAN Servicenode`容器运行状态

```
docker ps -l
```

> 如果`STATUS`显示非`Up`状态，联系安装部署人员进行错误排查。

7、`SWAN Servicenode`进行更新时，先删除原来的`swan-servicenode`容器，然后修改容器镜像版本(**如果版本号未变，但镜像更新过，只需要`docker rm`删除本地原来的镜像即可**)，再重新启动容器

```
# 1.删除原来的swan-servicenode容器
docker stop swan-servicenode
docker compose -f swan-manager.yml down

# 2.修改swan-servicenode.yml文件，将swan-servicenode的镜像版本修改为需要安装的版本

# 3.重新安装镜像
docker compose -f swan-servicenode.yml up -d
```

8、卸载SWAN Servicenode

(1) 仅卸载容器，保留历史数据，删除容器即可

```
docker stop swan-servicenode
docker compose -f swan-servicenode.yml down
```

(2) 纯净卸载，不保留历史数据

```
docker stop swan-servicenode
docker compose -f swan-servicenode.yml down

# 删除安装目录和安装数据
sudo rm -rf swan/servicenode
```

4. SWAN Client客户端安装流程

Linux CLI版本

Linux CLI为引导式（仿App）运行，Client和CLI在同一个进程内运行，操作时自动循环并等待输入指令
注意：Linux CLI 和 Manager 不能安装在同一台机器上
注意：Linux CLI 和 ServiceNode 也不能安装在同一台机器上

测试版 version 2.21.10.10

1、下载SWAN Client命令行

```
wget http://pkg-d.amianetworks.com.cn:19067/swan/client/swanclient
```

2、运行时需要sudo权限

```
sudo ./swanclient
```

正式版

安装和使用方式和测试版相同

使用指南

1. 输入manager地址，默认端口为7926.
2. 输入用户名和密码以登录
3. 按 1 选择网络，然后按数字键选择服务节点
4. 按 3 来建立隧道连接
5. 按 5 以查看绑定设备

- 在任何时候，输入`exit`来退出SWAN Client

Linux App版本

测试版 V0.10.3

执行`wget http://pkg-d.amianetworks.com.cn:19067/swan/client/swan_linux.deb`，使用`dpkg -i swan_linux`即开始安装，会安装至`usr/bin`目录下。使用时，建议使用terminal打开，输入`swan`即可运行

正式版 V0.10.6

在浏览器中，打开`https://s-wan.amianetworks.com.cn/swan/client/`在
SWAN Client安装部署指南 目录下找到*Linux*的*版本列表* 复制最新版的安装脚本到terminal执行进行下载，完成后使用`sudo dpkg -i swan_linux`即开始安装，会安装至`usr/bin`目录下。使用时单击图标打开，输入密码即可运行。

Windows App版本

测试版 V0.10.3

在浏览器中，打开`http://pkg-d.amianetworks.com.cn:19067/swan/client/swan_windows.exe`即可开始下载，完成后双击打开即开始安装。使用时双击打开即可运行

正式版 V0.10.6

在浏览器中，打开`https://s-wan.amianetworks.com.cn/swan/client/`在
SWAN Client安装部署指南 目录下找到*Windows*的*版本列表* 选择最新版进行下载，完成后双击打开即开始安装。使用时双击打开即可运行。

MacOS App版本

测试版 V0.10.3

在App Store中搜索并下载`TestFlight`应用，打开后登录自己的苹果邮箱，然后把邮箱地址在Teams上发送给翟鸣冲来注册加入测试团队，之后在TestFlight应用中搜索`SwanX`下载安装即可

正式版 V0.10.6

在App Store中搜索`SwanX`下载安装即可。

5. SWAN Manager业务错误码

▶ 现已定义如下错误码 ⬇️

```
[
  {
    "err_code": 101001,
    "err_en": "internal server error occurred, please contact administrator",
    "err_zh": "发生内部错误，请联系管理员",
    "remark": ""
  },
  {
    "err_code": 101002,
    "err_en": "invalid request parameters detected",
    "err_zh": "检测到无效的请求参数",
    "remark": ""
  },
  {
    "err_code": 101003,
    "err_en": "authentication token has expired, login to renew",
    "err_zh": "身份验证令牌已过期，请登录以续订",
    "remark": "需要使用refreshToken进行刷新"
  },
  {
    "err_code": 101004,
    "err_en": "insufficient privileges for this operation",
    "err_zh": "操作权限不足",
    "remark": "可能是token无效，亦或者后端需要客户端强制重新登录"
  },
  {
    "err_code": 101005,
    "err_en": "unknown error occurred",
    "err_zh": "未知错误发生",
    "remark": "web端无需使用该code"
  },
  {
    "err_code": 201001,
    "err_en": "guide components installation required",
    "err_zh": "引导流程安装需要",
    "remark": ""
  },
  {
    "err_code": 201002,
    "err_en": "administrative guide verification needed",
    "err_zh": "管理员引导验证必需",
    "remark": ""
  },
  {
    "err_code": 201003,
    "err_en": "network guide implementation incomplete",
    "err_zh": "网络引导实施不完整",
    "remark": ""
  },
  {
    "err_code": 201004,
    "err_en": "client guide deployment validation failed",
    "err_zh": "客户端引导部署验证失败",
```

```
    "remark": ""
  },
  {
    "err_code": 202001,
    "err_en": "admin user login failed, username or password incorrect",
    "err_zh": "管理员用户登录失败，用户名或密码错误",
    "remark": ""
  },
  {
    "err_code": 202002,
    "err_en": "admin user logout failed",
    "err_zh": "管理员用户注销失败",
    "remark": ""
  },
  {
    "err_code": 202003,
    "err_en": "regular user creation failed: duplicate username",
    "err_zh": "普通用户创建失败：用户名重复",
    "remark": ""
  },
  {
    "err_code": 202004,
    "err_en": "regular user deletion failed: invalid username",
    "err_zh": "普通用户删除失败：用户名无效",
    "remark": ""
  },
  {
    "err_code": 202005,
    "err_en": "regular user batch creation failed: duplicate username(s)",
    "err_zh": "普通用户批量创建失败：用户名重复",
    "remark": ""
  },
  {
    "err_code": 202006,
    "err_en": "regular user batch creation failed: invalid username(s)",
    "err_zh": "普通用户批量创建失败：用户名无效",
    "remark": ""
  },
  {
    "err_code": 202007,
    "err_en": "regular user list failed",
    "err_zh": "普通用户列表获取失败",
    "remark": ""
  },
  {
    "err_code": 202008,
    "err_en": "regular user static information query failed",
    "err_zh": "普通用户静态信息查询失败",
    "remark": ""
  },
  {
    "err_code": 202009,
    "err_en": "failed to create traffic limit and access policy",
```

```
    "err_zh": "创建流量限制与访问策略失败",
    "remark": ""
  },
  {
    "err_code": 202010,
    "err_en": "failed to delete traffic limit and access policy",
    "err_zh": "删除流量限制与访问策略失败",
    "remark": ""
  },
  {
    "err_code": 202011,
    "err_en": "failed to query traffic limit and access policy",
    "err_zh": "查询流量限制与访问策略失败",
    "remark": ""
  },
  {
    "err_code": 202012,
    "err_en": "admin user creation failed: duplicate username",
    "err_zh": "管理员用户创建失败：用户名重复",
    "remark": ""
  },
  {
    "err_code": 202013,
    "err_en": "admin user deletion failed: invalid username",
    "err_zh": "管理员用户删除失败：用户名无效",
    "remark": ""
  },
  {
    "err_code": 202014,
    "err_en": "admin user list failed",
    "err_zh": "管理员用户列表获取失败",
    "remark": ""
  },
  {
    "err_code": 202015,
    "err_en": "admin user information query failed",
    "err_zh": "管理员用户信息查询失败",
    "remark": ""
  },
  {
    "err_code": 202016,
    "err_en": "admin user email set failed",
    "err_zh": "管理员邮箱设置失败",
    "remark": ""
  },
  {
    "err_code": 202017,
    "err_en": "admin user password retrieval failed",
    "err_zh": "管理员密码找回失败",
    "remark": ""
  },
  {
    "err_code": 202018,
```

```
    "err_en": "admin user password reset failed",
    "err_zh": "管理员密码重置失败",
    "remark": ""
  },
  {
    "err_code": 202019,
    "err_en": "number of users created exceeds allowed limit",
    "err_zh": "创建的用户数超出允许的限制",
    "remark": ""
  },
  {
    "err_code": 202020,
    "err_en": "number of admins created exceeds allowed limit",
    "err_zh": "创建的管理员数超出允许的限制",
    "remark": ""
  },
  {
    "err_code": 202021,
    "err_en": "number of users batch created exceeds allowed limit",
    "err_zh": "批量创建的用户数超出允许的限制",
    "remark": ""
  },
  {
    "err_code": 202022,
    "err_en": "you can set a maximum traffic limit of up to 1 year",
    "err_zh": "流量限制最多设置1年",
    "remark": ""
  },
  {
    "err_code": 202023,
    "err_en": "traffic limit format error",
    "err_zh": "流量限制格式错误",
    "remark": ""
  },
  {
    "err_code": 202024,
    "err_en": "admin with manager privilege must set security email",
    "err_zh": "管理员必须设置安全邮箱",
    "remark": ""
  },
  {
    "err_code": 203001,
    "err_en": "user group create failed",
    "err_zh": "用户组创建失败",
    "remark": ""
  },
  {
    "err_code": 203002,
    "err_en": "user group delete failed",
    "err_zh": "用户组删除失败",
    "remark": ""
  },
  {
```

```
    "err_code": 203003,
    "err_en": "user group update failed",
    "err_zh": "用户组更新失败",
    "remark": ""
  },
  {
    "err_code": 203004,
    "err_en": "user group list failed",
    "err_zh": "用户组列表获取失败",
    "remark": ""
  },
  {
    "err_code": 203005,
    "err_en": "number of user groups created exceeds allowed limit",
    "err_zh": "创建的用户组数量超过允许限制",
    "remark": ""
  },
  {
    "err_code": 203006,
    "err_en": "number of user group members exceeds allowed limit",
    "err_zh": "用户组成员数量超过允许限制",
    "remark": ""
  },
  {
    "err_code": 203007,
    "err_en": "the user group already exists",
    "err_zh": "用户组已存在",
    "remark": ""
  },
  {
    "err_code": 203008,
    "err_en": "the user group does not exist",
    "err_zh": "用户组不存在",
    "remark": ""
  },
  {
    "err_code": 203009,
    "err_en": "default user group cannot be deleted",
    "err_zh": "默认用户组不能删除",
    "remark": ""
  },
  {
    "err_code": 203010,
    "err_en": "admin user group is immutable",
    "err_zh": "管理员用户组不可修改",
    "remark": ""
  },
  {
    "err_code": 203011,
    "err_en": "user group name exceeds 16 characters",
    "err_zh": "用户组名称不能超过16个字符",
    "remark": ""
  },
```

```
  {
    "err_code": 203012,
    "err_en": "user group name cannot be empty",
    "err_zh": "用户组名称不能为空",
    "remark": ""
  },
  {
    "err_code": 203013,
    "err_en": "get user group members failed",
    "err_zh": "获取用户组成员失败",
    "remark": ""
  },
  {
    "err_code": 204001,
    "err_en": "virtual network creation failed",
    "err_zh": "虚拟网络创建失败",
    "remark": ""
  },
  {
    "err_code": 204002,
    "err_en": "virtual network deletion failed",
    "err_zh": "虚拟网络删除失败",
    "remark": ""
  },
  {
    "err_code": 204003,
    "err_en": "virtual network update failed",
    "err_zh": "虚拟网络更新失败",
    "remark": ""
  },
  {
    "err_code": 204004,
    "err_en": "virtual network list failed",
    "err_zh": "虚拟网络列表获取失败",
    "remark": ""
  },
  {
    "err_code": 204005,
    "err_en": "virtual network get encryption list failed",
    "err_zh": "虚拟网络算法列表获取失败",
    "remark": ""
  },
  {
    "err_code": 204006,
    "err_en": "number of networks created exceeds allowed limit",
    "err_zh": "虚拟网络数量超过允许限制",
    "remark": ""
  },
  {
    "err_code": 204007,
    "err_en": "the network already exists",
    "err_zh": "虚拟网络已存在",
    "remark": ""
```

```
  },
  {
    "err_code": 204008,
    "err_en": "virtual network name exceeds 16 characters",
    "err_zh": "虚拟网络名称不能超过16个字符",
    "remark": ""
  },
  {
    "err_code": 204009,
    "err_en": "Virtual network name cannot be empty",
    "err_zh": "虚拟网络名称不能为空",
    "remark": ""
  },
  {
    "err_code": 205001,
    "err_en": "virtual network group creation failed",
    "err_zh": "虚拟网络组创建失败",
    "remark": ""
  },
  {
    "err_code": 205002,
    "err_en": "virtual network group deletion failed",
    "err_zh": "虚拟网络组删除失败",
    "remark": ""
  },
  {
    "err_code": 205003,
    "err_en": "virtual network group update failed",
    "err_zh": "虚拟网络组更新失败",
    "remark": ""
  },
  {
    "err_code": 205004,
    "err_en": "virtual network group list failed",
    "err_zh": "虚拟网络组列表获取失败",
    "remark": ""
  },
  {
    "err_code": 205005,
    "err_en": "number of network groups created exceeds allowed limit",
    "err_zh": "虚拟网络组数量超过允许限制",
    "remark": ""
  },
  {
    "err_code": 205006,
    "err_en": "the virtual network group name already exists",
    "err_zh": "虚拟网络组名称已存在",
    "remark": ""
  },
  {
    "err_code": 205007,
    "err_en": "the virtual network group does not exist",
    "err_zh": "虚拟网络组不存在",
```

```
    "remark": ""
  },
  {
    "err_code": 205008,
    "err_en": "bind user group failed: invalid state",
    "err_zh": "绑定用户组失败: 无效状态",
    "remark": ""
  },
  {
    "err_code": 205009,
    "err_en": "virtual network group name exceeds 16 characters",
    "err_zh": "虚拟网络组名称不能超过16个字符",
    "remark": ""
  },
  {
    "err_code": 205010,
    "err_en": "virtual network group name cannot be empty",
    "err_zh": "虚拟网络组名称不能为空",
    "remark": ""
  },
  {
    "err_code": 206001,
    "err_en": "service node creation failed",
    "err_zh": "服务节点创建失败",
    "remark": ""
  },
  {
    "err_code": 206002,
    "err_en": "service node deletion failed",
    "err_zh": "服务节点删除失败",
    "remark": ""
  },
  {
    "err_code": 206003,
    "err_en": "service node update failed",
    "err_zh": "服务节点更新失败",
    "remark": ""
  },
  {
    "err_code": 206004,
    "err_en": "service node list failed",
    "err_zh": "服务节点列表获取失败",
    "remark": ""
  },
  {
    "err_code": 206005,
    "err_en": "service node information query failed",
    "err_zh": "服务节点信息查询失败",
    "remark": ""
  },
  {
    "err_code": 206006,
    "err_en": "service node bring online failed",
```

```
    "err_zh": "服务节点上线失败",
    "remark": ""
  },
  {
    "err_code": 206007,
    "err_en": "service node set offline failed",
    "err_zh": "服务节点下线失败",
    "remark": ""
  },
  {
    "err_code": 206008,
    "err_en": "service node get errors failed",
    "err_zh": "获取服务节点错误失败",
    "remark": ""
  },
  {
    "err_code": 206009,
    "err_en": "service node clear errors failed",
    "err_zh": "清除服务节点错误失败",
    "remark": ""
  },
  {
    "err_code": 206010,
    "err_en": "service node get version info failed",
    "err_zh": "获取服务节点版本信息失败",
    "remark": ""
  },
  {
    "err_code": 206011,
    "err_en": "service node id cannot be empty",
    "err_zh": "service node id不能为空",
    "remark": ""
  },
  {
    "err_code": 206012,
    "err_en": "service node name cannot be empty",
    "err_zh": "service node name不能为空",
    "remark": ""
  },
  {
    "err_code": 207001,
    "err_en": "log audit template name length exceeds 16 characters",
    "err_zh": "日志审计模板名称长度超过限制，请缩短名称，建议在16个字符以内",
    "remark": ""
  },
  {
    "err_code": 207002,
    "err_en": "log audit template creation failed: duplicate template name",
    "err_zh": "日志审计模板创建失败，模板名重复",
    "remark": ""
  },
  {
    "err_code": 207003,
```

```
    "err_en": "log audit template deletion failed: invalid template name",
    "err_zh": "日志审计模板删除失败，无效的模板名",
    "remark": ""
  },
  {
    "err_code": 207004,
    "err_en": "log audit export failed",
    "err_zh": "日志审计导出失败",
    "remark": ""
  },
  {
    "err_code": 207005,
    "err_en": "log audit get event distribution failed",
    "err_zh": "获取日志审计事件分布失败",
    "remark": ""
  },
  {
    "err_code": 207006,
    "err_en": "log audit get event num failed",
    "err_zh": "获取日志审计事件数量失败",
    "remark": ""
  },
  {
    "err_code": 207007,
    "err_en": "log audit get event trend failed",
    "err_zh": "获取日志审计事件趋势失败",
    "remark": ""
  },
  {
    "err_code": 207008,
    "err_en": "log audit get logs failed",
    "err_zh": "获取日志审计日志失败",
    "remark": ""
  },
  {
    "err_code": 207010,
    "err_en": "log audit get report time trend failed",
    "err_zh": "获取日志审计报告时间趋势失败",
    "remark": ""
  },
  {
    "err_code": 207011,
    "err_en": "log audit get risk level distribution failed",
    "err_zh": "获取日志审计风险级别分布失败",
    "remark": ""
  },
  {
    "err_code": 207012,
    "err_en": "log audit get template list failed",
    "err_zh": "获取日志审计模板失败",
    "remark": ""
  },
  {
```

```
    "err_code": 207013,
    "err_en": "log audit get top user failed",
    "err_zh": "获取日志审计顶级用户失败",
    "remark": ""
  },
  {
    "err_code": 207014,
    "err_en": "log audit get user distribution failed",
    "err_zh": "获取日志审计用户分布失败",
    "remark": ""
  },
  {
    "err_code": 207015,
    "err_en": "log audit template update failed",
    "err_zh": "日志审计模板更新失败",
    "remark": ""
  },
  {
    "err_code": 207016,
    "err_en": "log audit rule creation failed",
    "err_zh": "日志审计规则创建失败",
    "remark": ""
  },
  {
    "err_code": 207017,
    "err_en": "log audit rule does not exist",
    "err_zh": "日志审计规则不存在",
    "remark": ""
  },
  {
    "err_code": 207018,
    "err_en": "user type cannot be empty",
    "err_zh": "user type不能为空",
    "remark": ""
  },
  {
    "err_code": 207019,
    "err_en": "event type cannot be empty",
    "err_zh": "event type不能为空",
    "remark": ""
  },
  {
    "err_code": 207020,
    "err_en": "risk level cannot be empty",
    "err_zh": "risk level不能为空",
    "remark": ""
  },
  {
    "err_code": 207021,
    "err_en": "user type cannot contain empty value",
    "err_zh": "user type不能包含空值",
    "remark": ""
  },
```

```
{
  "err_code": 207022,
  "err_en": "event type cannot contain empty value",
  "err_zh": "event type不能包含空值",
  "remark": ""
},
{
  "err_code": 207023,
  "err_en": "risk level cannot contain empty value",
  "err_zh": "risk level不能包含空值",
  "remark": ""
},
{
  "err_code": 208001,
  "err_en": "license status query failed",
  "err_zh": "license状态查询失败",
  "remark": ""
},
{
  "err_code": 208002,
  "err_en": "license activation failed",
  "err_zh": "license激活失败",
  "remark": ""
},
{
  "err_code": 209001,
  "err_en": "get version failed",
  "err_zh": "获取版本失败",
  "remark": ""
},
{
  "err_code": 301001,
  "err_en": "user login failed",
  "err_zh": "用户登录失败",
  "remark": ""
},
{
  "err_code": 301002,
  "err_en": "user logout failed",
  "err_zh": "用户注销失败",
  "remark": ""
},
{
  "err_code": 301003,
  "err_en": "user group information query failed",
  "err_zh": "用户组信息查询失败",
  "remark": ""
},
{
  "err_code": 301004,
  "err_en": "user access policy query failed",
  "err_zh": "用户访问策略查询失败",
  "remark": ""
```

```
  },
  {
    "err_code": 301005,
    "err_en": "user online check failed",
    "err_zh": "用户在线检查失败",
    "remark": ""
  },
  {
    "err_code": 302001,
    "err_en": "network status query failed",
    "err_zh": "网络状态查询失败",
    "remark": ""
  },
  {
    "err_code": 302002,
    "err_en": "network configuration failed",
    "err_zh": "网络服务列表查询失败",
    "remark": ""
  },
  {
    "err_code": 302003,
    "err_en": "network name query failed",
    "err_zh": "网络名称查询失败",
    "remark": ""
  },
  {
    "err_code": 302004,
    "err_en": "service node status query failed",
    "err_zh": "服务节点状态查询失败",
    "remark": ""
  },
  {
    "err_code": 302005,
    "err_en": "service node name query failed",
    "err_zh": "服务节点名称查询失败",
    "remark": ""
  },
  {
    "err_code": 303001,
    "err_en": "client node registration failed",
    "err_zh": "客户端节点注册失败",
    "remark": ""
  },
  {
    "err_code": 303002,
    "err_en": "client node get online config failed",
    "err_zh": "客户端节点获取在线配置失败",
    "remark": ""
  },
  {
    "err_code": 303003,
    "err_en": "client node bring online failed",
    "err_zh": "客户端节点上线失败",
```

```
    "remark": ""
  },
  {
    "err_code": 303004,
    "err_en": "client node set offline failed",
    "err_zh": "客户端节点下线失败",
    "remark": ""
  },
  {
    "err_code": 303005,
    "err_en": "client node get latest version failed",
    "err_zh": "客户端节点获取最新版本失败",
    "remark": ""
  },
  {
    "err_code": 303006,
    "err_en": "client node is already online",
    "err_zh": "客户端节点已经在线",
    "remark": ""
  },
  {
    "err_code": 303007,
    "err_en": "client node is already offline",
    "err_zh": "客户端节点已经下线",
    "remark": ""
  },
  {
    "err_code": 303008,
    "err_en": "target server not in network",
    "err_zh": "目标节点不在网络中",
    "remark": ""
  },
  {
    "err_code": 401001,
    "err_en": "token refresh failed",
    "err_zh": "令牌刷新失败",
    "remark": ""
  },
  {
    "err_code": 401002,
    "err_en": "password modification failed",
    "err_zh": "密码修改失败",
    "remark": ""
  },
  {
    "err_code": 401003,
    "err_en": "Password strength too low",
    "err_zh": "密码验证失败，密码不符合要求",
    "remark": ""
  },
  {
    "err_code": 401004,
    "err_en": "device list failed",
```

```
    "err_zh": "设备列表获取失败",
    "remark": ""
  },
  {
    "err_code": 401005,
    "err_en": "device unbind failed",
    "err_zh": "设备解绑失败",
    "remark": ""
  },
  {
    "err_code": 401006,
    "err_en": "device name update failed",
    "err_zh": "设备名称更新失败",
    "remark": ""
  },
  {
    "err_code": 401007,
    "err_en": "user bound too many device, please check",
    "err_zh": "设备绑定数量超出限制，请检查",
    "remark": ""
  },
  {
    "err_code": 401008,
    "err_en": "username is empty",
    "err_zh": "用户名为空",
    "remark": ""
  },
  {
    "err_code": 401009,
    "err_en": "access control error: user has no group",
    "err_zh": "访问控制错误：用户没有组",
    "remark": ""
  },
  {
    "err_code": 401010,
    "err_en": "unrecognized account types",
    "err_zh": "无法识别的帐户类型",
    "remark": ""
  },
  {
    "err_code": 401011,
    "err_en": "only admin user operation",
    "err_zh": "只有管理员用户操作",
    "remark": ""
  },
  {
    "err_code": 401012,
    "err_en": "invalid user",
    "err_zh": "无效的用户",
    "remark": ""
  },
  {
    "err_code": 401013,
```

```
    "err_en": "the user does not exist",
    "err_zh": "用户不存在",
    "remark": ""
  },
  {
    "err_code": 401014,
    "err_en": "username or password is incorrect",
    "err_zh": "用户名或密码不正确",
    "remark": ""
  },
  {
    "err_code": 402001,
    "err_en": "mfa get qr code failed",
    "err_zh": "MFA获取二维码失败",
    "remark": ""
  },
  {
    "err_code": 402002,
    "err_en": "mfa bind failed",
    "err_zh": "MFA绑定失败",
    "remark": ""
  },
  {
    "err_code": 402003,
    "err_en": "mfa unbind failed",
    "err_zh": "MFA解绑失败",
    "remark": ""
  },
  {
    "err_code": 402004,
    "err_en": "mfa verify failed",
    "err_zh": "MFA验证失败",
    "remark": ""
  },
  {
    "err_code": 402005,
    "err_en": "mfa login failed",
    "err_zh": "MFA登录失败",
    "remark": ""
  },
  {
    "err_code": 402006,
    "err_en": "mfa code incorrect",
    "err_zh": "MFA验证码不正确",
    "remark": ""
  },
  {
    "err_code": 402007,
    "err_en": "mfa login code incorrect",
    "err_zh": "MFA登录验证码不正确",
    "remark": ""
  },
  {
```

```
    "err_code": 403001,
    "err_en": "email send code failed",
    "err_zh": "邮件发送验证码失败",
    "remark": ""
  },
  {
    "err_code": 403002,
    "err_en": "security email verification failed",
    "err_zh": "安全电子邮件验证失败",
    "remark": ""
  },
  {
    "err_code": 403003,
    "err_en": "requests rate exceeds once per minute",
    "err_zh": "发邮件请求过于频繁，每分钟只能请求一次",
    "remark": ""
  },
  {
    "err_code": 403004,
    "err_en": "this email is bind by other user",
    "err_zh": "该邮箱已被其他用户绑定",
    "remark": ""
  },
  {
    "err_code": 403005,
    "err_en": "Send email verification code timeout",
    "err_zh": "邮件发送验证码超时",
    "remark": ""
  },
  {
    "err_code": 404001,
    "err_en": "please configure the Manager management address first",
    "err_zh": "请先配置Manager管理地址",
    "remark": ""
  },
  {
    "err_code": 405001,
    "err_en": "the search page exceeds the total page, please check the search conditions",
    "err_zh": "搜索页数超过总页数，请检查搜索条件",
    "remark": ""
  },
  {
    "err_code": 405002,
    "err_en": "the search page cannot be less than 1",
    "err_zh": "搜索页数不能小于1",
    "remark": ""
  },
  {
    "err_code": 409001,
    "err_en": "the server node config update failed: server port not sufficient",
    "err_zh": "服务器节点配置更新失败：服务器端口不足",
    "remark": ""
  },
```

```
  {
    "err_code": 409002,
    "err_en": "the server node config update failed: sync network port change to manager
failed",
    "err_zh": "服务器节点配置更新失败：同步网络端口更改到Manager失败",
    "remark": ""
  },
  {
    "err_code": 409003,
    "err_en": "the server node config update failed: port range invalid",
    "err_zh": "服务器节点配置更新失败：端口范围无效",
    "remark": ""
  },
  {
    "err_code": 409004,
    "err_en": "the server node config update failed: server address invalid",
    "err_zh": "服务器节点配置更新失败：服务器地址无效",
    "remark": ""
  },
  {
    "err_code": 409005,
    "err_en": "the server node config update failed: server network format error, please
check, eg: 192.168.0.0/24;172.171.1.1/32",
    "err_zh": "服务器节点配置更新失败：服务器网络格式错误，请检查，例如：
192.168.0.0/24;172.171.1.1/32",
    "remark": ""
  },
  {
    "err_code": 409006,
    "err_en": "the server node start failed, please check",
    "err_zh": "服务器节点启动失败，请检查",
    "remark": ""
  },
  {
    "err_code": 409007,
    "err_en": "the server node stop failed, please check",
    "err_zh": "服务器节点停止失败，请检查",
    "remark": ""
  },
  {
    "err_code": 409008,
    "err_en": "the server node update network failed",
    "err_zh": "服务器节点更新网络失败",
    "remark": ""
  },
  {
    "err_code": 409009,
    "err_en": "the server node distribute addr for networks failed",
    "err_zh": "服务器节点为网络分配地址失败",
    "remark": ""
  },
  {
    "err_code": 409010,
```

```
    "err_en": "the server node disconnect to manager, please check network status",
    "err_zh": "服务器节点与Manager断开连接，请检查网络状态",
    "remark": ""
  },
  {
    "err_code": 409011,
    "err_en": "the device can not be recognized, please check if the device machine code has
changed",
    "err_zh": "设备无法识别，请检查设备机器码是否已更改",
    "remark": ""
  },
  {
    "err_code": 409012,
    "err_en": "the server node's software version is too low to be accessed",
    "err_zh": "服务器节点的软件版本过低，无法访问",
    "remark": ""
  }
]
```