

License部署手册

部署前提(必要)

1. 部署License的机器系统时钟，必须配置好时区并同步当前时间
- 同步时区：执行 tzselect 命令，选择 4 Asia -> 11 China -> 1 Beijing Time -> 1 Yes

◦ 支持ntp的使用 timedatectl set-ntp true 同步

◦ 不支持ntp的使用 timedatectl set-time "YYYY-MM-DD hh:mm:ss" 手动设置时间
2. 对外开放端口：20088、16890 （用于给其他服务进行验证）

配置更新

暂无

1. License安装流程

► 点击查看License版本：

版本号	版本内容	其他依赖镜像
V25.7.100	- 普通用户不显示”创建人“搜索框 - 修复根据”证书状态“筛选总数不对的问题 sapphire.iam:25.6.2	mongo:8.0
V25.7.101	- 适配iam25.6.3 - 配置文件增加mtls-enabled sapphire.iam:25.6.3	mongo:8.0
v25.7.102	- 支持数据库配置为FileDB - 修复证书列表排序问题 - 修复不能刷新token的问题 sapphire.iam:25.6.4	mongo:8.0（配置为FileDB时不依赖该镜像）
v25.7.103	- 修复刷新token失败的问题 - 修复查看证书详情“应用限制”数不对的问题 sapphire.iam:25.6.4	mongo:8.0（配置为FileDB时不依赖该镜像）
v25.7.104	- 修复证书列表分页排序问题 - 修复激活时返回的“应用限制”数不对的问题 sapphire.iam:25.6.4	mongo:8.0（配置为FileDB时不依赖该镜像）
v25.7.105	- 优化激活证书过期时的返回	sapphire.iam:25.6.4 mongo:8.0（配置为FileDB时不依赖该镜像）
v25.7.106	- 适配iam25.6.5	sapphire.iam:25.6.5 mongo:8.0（配置为FileDB时不依赖该镜像）
v25.7.107	- 修复重启容器会重新进入引导页的问题	sapphire.iam:25.6.5 mongo:8.0（配置为FileDB时不依赖该镜像）
v25.8.100	- 适配iam25.8.0	sapphire.iam:25.8.0 mongo:8.0（配置为FileDB时不依赖该镜像）
v25.8.101	- 优化token过期错误码的处理	sapphire.iam:25.8.0 mongo:8.0（配置为FileDB时不依赖该镜像）

| **v25.8.102** | - 适配iam.modules25.8.5
| sapphire.iam:25.8.0
mongo:8.0 (配置为FileDB时不依赖该镜像) |
| **v25.8.103** | - 修复普通用户证书列表中显示了其他用户创建的证书的问题
- 适配iam25.8.0 | sapphire.iam:25.8.0
mongo:8.0 (配置为FileDB时不依赖该镜像) |
| **v25.8.104** | - 适配iam provisioning策略
| sapphire.iam:25.8.0
mongo:8.0 (配置为FileDB时不依赖该镜像) |
| **25.9.0** | - 适配 iam25.9.0版本
| sapphire.iam:25.9.0
mongo:8.0 (配置为FileDB时不依赖该镜像) |

1.1 准备工作

1、在 License 服务器上使用 `docker login` 命令登录下面的私有库 `registry.amianetworks.com.cn`，账号是 `amia`，密码是 `AmiaNetworks`

```
docker login registry.amianetworks.com.cn
```

1.2 安装步骤

1、下载 `license` 最新安装配置

```
# 下载license.tar.gz，解压到当前目录，并进入asn-license目录。  
wget https://downloads.amianetworks.com.cn/License/v25.9.0/license/license.tar.gz && tar -xvf license.tar.gz
```

2、asn-license目录结构说明

- `cert/`目录：包含必需的CA证书文件
- 与iam认证的证书，在开启iam `'mtls'` 认证时需要，向iam服务器管理员申请
 - `'client-website.pem'`：客户端证书
 - `'client-website.key'`：客户端私钥
 - `'server-ca.crt'`：服务器CA证书
- 邮箱账户的证书，在开启邮件服务 `'oauth2'` 时需要，向邮件服务管理员申请
 - `'mailer.amianetworks.com.cn.key'`：邮件服务私钥，向邮件服务管理员申请
 - `'mailer.amianetworks.com.cn.crt'`：邮件服务证书，向邮件服务管理员申请

3、进入`config` 目录下并修改`license.conf`文件

```
cd config && vim license.conf
```

```
#省略其他配置...  
  
# iam config  
iam:  
  provider: "sapphire"  
  host: "iam.amianetworks.com.cn" # 修改iam host  
  port: 17930 # 修改iam port  
  mtls:  
    enabled: false # 可选择修改：如果开启 "mtls"，需要向iam服务器管理员申请client-website.pem, client-website.key  
    client_pem: "/etc/website/cert/client-website.pem"  
    client_key: "/etc/website/cert/client-website.key"  
    server_ca: "/etc/website/cert/server-ca.crt"
```

```
# email config
email:
  tls: true
  type: "smtp" #"smtp" or "oauth2" # 可选择修改: 在开启邮件服务'oauth2'时需要, 向邮件服务管理员申请 mailer.amianetworks.
# smtp
host: "smtp.163.com" # 修改host
email: "amianetworks@163.com" # 修改email
username: "amianetworks@163.com" # 修改username
password: "" # 修改password
port: 465 # 修改port
# oauth2
client_email: "noreply@amianetworks.com.cn"
client_id: "91e66650-824f-4d0f-9cf9-6b64293219b5"
client_secret_key_path: "/etc/websited/cert/mailer.amianetworks.com.cn.key"
Client_secret crt_path: "/etc/websited/cert/mailer.amianetworks.com.cn.crt"
tenant_id: "cb1d6a3d-74ae-470b-ad99-77d32752edc6"
```

4、回到 `asn-license` 目录修改docker-compose配置文件

- 修改 `licensed.yml` ,更新 `asn-licensed` 为最新的镜像版本号



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

services:
  #省略其他配置...
  asn-licensed:
    image: registry.amianetworks.com.cn/asn-licensed:v25.9.0 # 修改license服务的版本号
    container_name: asn-licensed
    restart: always # auto restart the container if it fails
    ports:
      - "20088:20088"
      - "16890:16890"
    depends_on:
      - "asn-mdb"
      - "sapphire-iam"
    volumes:
      - ./cert:/etc/license/cert/
      - ./config:/etc/license/config/
      - ./log/license:/var/log/license/
      - /etc/localtime:/etc/localtime
    networks:
      - license_network
  #省略其他配置...
```

5、启动 License 容器

```
cd asn-license
docker compose -f licensed.yml up -d
```

6、查看License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、License 进行更新时, 先删除原来的 license 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license容器
docker compose -f licensed.yml down

# 2.修改licensed.yml文件，将asn-licensed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 licensed.yml up -d
```

8、卸载License

(1) 仅卸载容器，保留历史数据，删除容器即可

```
docker compose -f licensed.yml down
```

(2) 纯净卸载，不保留历史数据

```
docker compose -f licensed.yml down
# 删除数据卷
docker volume rm license_mongo_data
# 删除安装目录和安装数据
rm -rf asn-license
```

1.3 Web管理平台

浏览器访问 `http://<ip>:16890`，按照引导程序进行配置管理员后登录



2. 开发接入License

步骤2 仅需开发人员阅览，测试人员请直接 [跳到步骤3进行证书激活](#)

2.1 gRPC接入

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

syntax = "proto3";

option go_package = ".;/license";

package license;

service LicenseManager {
    // Activate or validate a license
    rpc ActivateOrValidateLicense (License.ActivateRequest) returns (License.ActivateResponse);
}

// License related messages
message License {
    // Fixed field definitions
    message Application {
        message SwanFields {
            int32 swan_max_service_nodes = 1; // the maximum number of service nodes
            int32 swan_max_clients = 2;      // the maximum number of clients
        }
    }
}
```

```

message ScarlettFields {
    int32 scar_max_monitoring_clients = 1; // the maximum number of monitoring clients
}

message ActivateRequest {
    string license_key = 1; // license key
    string device_id = 2; // device id
}

// activate/validate license response
message ActivateResponse {
    string license_key = 1; // license key
    string application = 2; // application name
    string license_type = 3; // license type
    string create_time = 4; // create time
    string expiration_time = 5; // expiration time
    string status = 6; // status: active, inactive, expired
    string device_id = 7; // device id

    // custom fields
    oneof custom_fields {
        Application.SwanFields swan = 10; // SWAN-specific fields
        Application.ScarletteFields scarlette = 11; // Scarlett-specific fields
    }
}

```

2.2 RESTful接入

发送请求

```

curl --location --request POST 'http://<ip>:16890/license/activate' \
--header 'Content-Type: application/json' \
--data-raw '{
    "licenseKey": "激活码",
    "deviceId": "设备唯一标识"
}'

```

响应返回证书信息如下：不同应用的激活返回的内容不同，目前是swan/scarlette：

```

{
    "errCode": 0,
    "errMsg": "",
    "resp": {
        "licenseKey": "38ddc9d1780e",
        "application": "swan",
        "licenseType": "trial",
        "createTime": "2025-06-27 14:02:53",
        "expirationTime": "2025-07-04 14:02:28",
        "deviceId": "E10ADC3949BA59ABBE56E057F20F883E",
        "status": "active",
        "swan": {
            "swanMaxServiceNodes": 100,
            "swanMaxClients": 100
        }
    }
}

```

3. License激活验证

3.1 swan证书激活

step1: 通过license服务的web管理平台来创建swan的证书获得“激活码”。



step2 : 然后在SWAN引导页的注册码输入框填入创建证书后获得的“激活码”来激活。

License界面变化如下：

