零信任安全管理平台

运维部署手册

江苏云涌电子科技股份有限公司

2022-08-31

目 录

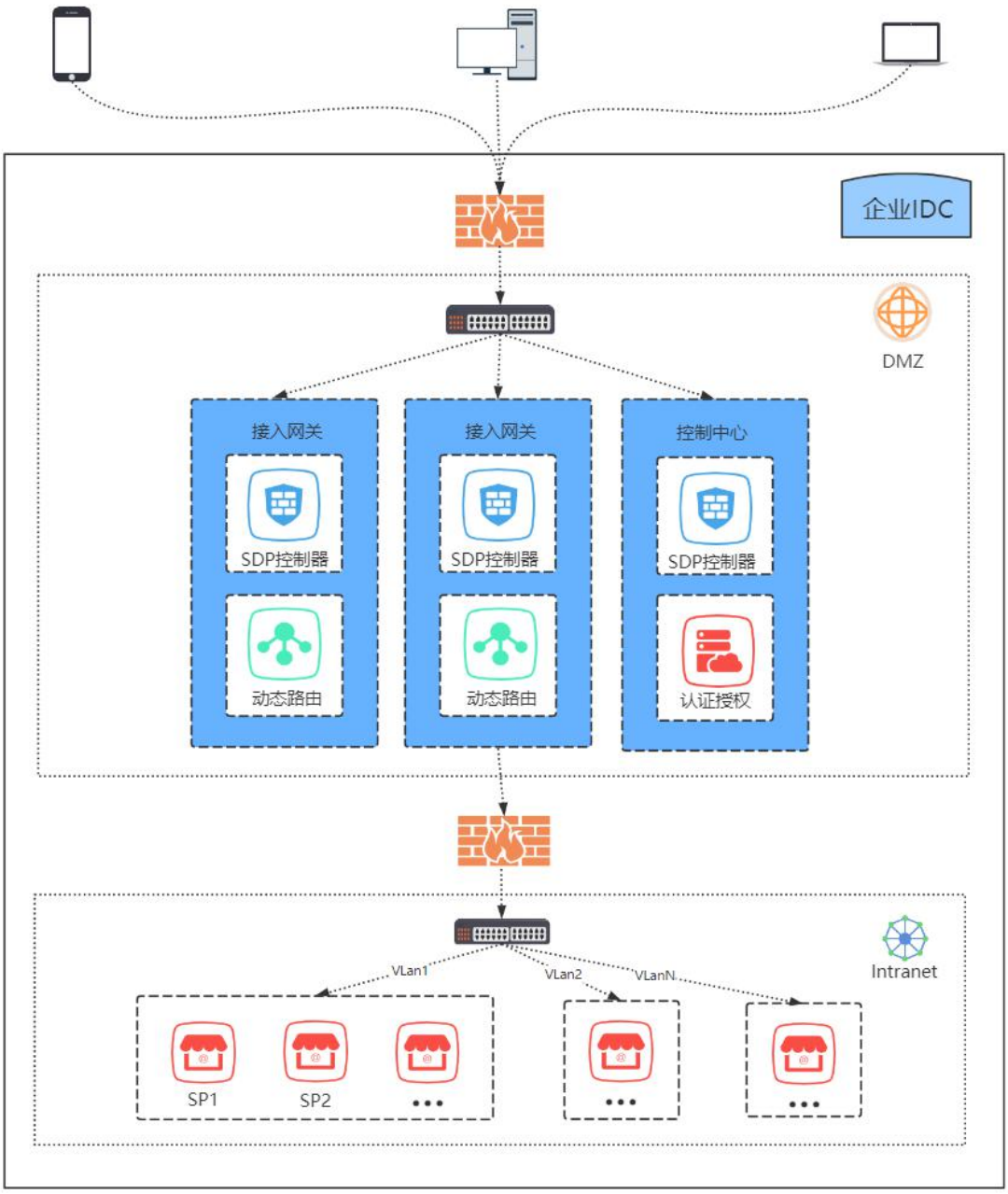
1 产品部署说明

本手册介绍了零信任安全管理平台产品架构、部署方式、运行环境、以及各个组件的部署方法，用于部署和运维指导，阅读对象为项目实施交付或运维人员，云涌零信任安全管理平台由安全大脑、接入网关、管理控制台、认证中心客户端等模块组合而成。

2 部署方式说明

2.1 典型部署模式

零信任安全管理平台支持云端部署、私有化部署及混合部署三种部署方式。支持高可用及负载均衡。以下是部署架构示意图：



零信任安全管理平台部署架构

云端部署

零信任安全大脑、管理控制台、网关均部署在云端。其中零信任连接器需要与业务服务部署在一个网络环境内。针对云端部署，系统支持多租户管理，租户间数据隔离。

云端部署可以将所有零信任核心组件上云，业务服务通过连接器反向连接到云端网关。业务服务侧网络环境只要保证可以访问公网，无需搭建专线及申请固

定公网 IP 等资源。

私有化部署

零信任安全大脑、管理控制台、接入网关均部署在企业 DMZ 区。其中接入网关需要与内网业务服务部署在一个网络环境内，同时客户端、安全大脑、零信任网关也需要保持网络连通性。

私有化部署方式适用于自建数据中心的中大型企业。产品支持多数据中心统一管理，每个数据中心均部署接入网关，安全大脑统一管理多接入点。

混合部署

安全大脑及管理控制台部署在云端，网关组件部署在企业内部。此方案适用于对核心业务系统防护安全性要求较高，同时又需要尽可能节省网络资源的客户场景。

2.2 推荐部署配置

标准部署（推荐用户数 100 以下）

产品模块名称	机器数	系统配置
控制中心服务器	1	CPU：4 核、内存：8G、硬盘：40G
接入网关服务器	1	CPU：2 核、内存：4G、硬盘：40G

* 数据库及中间件服务部署在控制中心服务器

* 以上配置不包括服务器双活或者数据备份服务器

高可用负载均衡部署

产品模块名称	机器数	系统配置
控制中心服务器	n	CPU：4 核、内存：8G、硬盘：40G
接入网关服务器	n	CPU：2 核、内存：4G、硬盘：40G
数据库及中间件服务器	n	CPU：4 核、内存：8G、硬盘：500G

* 其中 n 代表不同的服务器数目，n 从 2 开始。

注：以上推荐配置 CPU 是 X86_64 的架构。服务器 CPU 如果采用国产化 ARM、MIPS、飞腾等架构，需要根据实际情况做相应调整。网关服务器的配置跟实际接入的应用服务器相关，如果应用服务接入较多或者数据传输量较大的情况下，需要根据实际的应用场景进行添加。

3 运行环境要求

3.1 硬件环境

- 服务端：支持 x86 架构以及国产化 ARM、MIPS、飞腾等架构服务器。
- 客户端：支持全通用终端平台，包括 Windows，MacOS、Linux 等主流 PC 平台。

3.2 软件环境

- 服务端：
  - 操作系统：CentOS/RHEL/Ubuntu 等主流 Linux 操作系统，同时也支持统信、麒麟、鸿蒙等主流国产操作系统。

- 数据库：Mysql 8.0.27
- 中间件：Redis 6、EMQX 4.2
- JAVA：JDK 8

- 客户端：
  - CentOS/RHEL/Ubuntu 等主流 Linux 操作系统，同时也支持统信、麒麟、鸿蒙等国产化操作系统。

3.3 网络环境

公网 IP 及域名控制中心、接入网关/云联网关，RBI 网关、移动端代理各需要 1 个公网固定 IP，同时还需要 1 个泛域名，以及一套 CA 证书。

4. 部署控制中心服务

4.1 IP 配置

cd /etc/sysconfig/network-scripts/

编辑网卡文件

```
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="static"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
```

IPV6INIT="yes"

IPV6_AUTOCONF="yes"

IPV6_DEFROUTE="yes"

IPV6_FAILURE_FATAL="no"

IPV6_ADDR_GEN_MODE="stable-privacy"

NAME="enp2s0"

UUID="944788d6-78f3-4599-8b3d-91370c711b33"

DEVICE="enp2s0"

ONBOOT="yes"

IPADDR=192.168.x.x

NETMASK=255.255.255.0

GATEWAY=192.168.x.x

DNS1=8.8.8.8

注： IPADDR 就是静态 IP，NETMASK 是子网掩码，GATEWAY 就是网关或者路由地址 BOOTPROTO=static 设置为静态

4.2 控制中心 JDK 安装

下载 1.8-262 的 jdk 的 tar 文件， 加压 tar 文件到 /usr/local/jdk

配置环境变量

vi /etc/profile

export JAVA_HOME=/usr/local/jdk

export

CLASSPATH=.:${JAVA_HOME}/jre/lib/rt.jar:${JAVA_HOME}/lib/dt.jar:

${JAVA_HOME}/lib/tools.jar

export PATH=$PATH:${JAVA_HOME}/bin

source /etc/profile

4.3 安装 Mysql

通过官方的 rpm 包安装， 步骤如下：

检查目前系统的账号信息，如果没有 mysql 组和账号，添加组合账号

查看你用户和组的存在：

cat /etc/passwd|grep -v nologin|grep -v halt|grep -v shutdown|awk -F ":" '{print $1 "|" $3 "1" $4}' | more

如果不存在，创建 mysql 组合账号

创建 mysql 用户组，groupadd mysql

创建一个用户名为 mysql 的用户，并加入 mysql 用户组，useradd -g mysql mysql

修改 mysql 密码，passwd mysql

准备离线包 mysql-8.0.27-1.el7.x86_64.rpm-bundle.tar， 上传到 /usr/local/

mkdir /usr/local/mysql， 进入 /usr/local/mysql 目录，加压 tar 文件

tar -xvf mysql-8.0.27-1.el7.x86_64.rpm-bundle.tar

执行如下命令安装 mysql 包

rpm -ivh mysql-community-common-8.0.27-1.el7.x86_64.rpm

--nodeps --force

rpm -ivh mysql-community-libs-8.0.27-1.el7.x86_64.rpm --nodeps --force

rpm -ivh mysql-community-client-8.0.27-1.el7.x86_64.rpm --nodeps --force

rpm -ivh mysql-community-server-8.0.27-1.el7.x86_64.rpm --nodeps －force

初始化

mysqld --initialize

chown mysql:mysql /var/lib/mysql -R

systemctl start mysqld.service #启动 mysqld 服务

systemctl enable mysqld 将 MySQL 添加到开机启动

通过 cat /var/log/mysqld.log | grep password 命令查看数据库的密码

通过 mysql -uroot -p 敲回车键进入数据库登陆界面

输入刚刚查到的密码，进行数据库的登陆

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'root';

退出可以使用新的密码登录

mysql -uroot -proot

通过以下命令，进行远程访问的授权

create user 'zt'@'%' identified with mysql_native_password by 'zt';

grant all privileges on *.* to 'zt'@'%' with grant option;

flush privileges;

修改数据路径

编辑/etc/my.cnf，将如下内容放入文件中

[mysqld]

datadir=/data/mysql

socket=/data/mysql/mysql.sock

[client]

default-character-set=utf8

socket=/data/mysql/mysql.sock

[mysql]

default-character-set=utf8

socket=/data/mysql/mysql.sock

移动数据文件夹

mkdir /data

mv /var/lib/mysql /data

mkdir /data/mysql/log

chown -R mysql:mysql /data/mysql

重启服务 systemctl restart mysqld

4.4 安装 Redis

yum install epel-release

yum install redis

启动 redis　service redis start

停止 redis　service redis stop

查看 redis 运行状态 service redis status

设置开机启动 chkconfig redis on

进入 redis 服务 redis-cli

打开配置文件　vi /etc/redis.conf

#把这一行注释，监听所有 IP

#bind 127.0.0.1

#protected-mode yes

protected-mode yes

#requirpass，保护模式开启的时候要配置密码或者 bind ip

requirepass 123456

notify-keyspace-events Ex 开启 redis 过期配置

#修改本参数，指定数据目录

dir /data/redis/data

#修改本参数，指定日志目录

logfile /data/redis/redis_log.log

启动服务：systemctl start reids && systemctl enable redis

4.5 安装 emqx

安装所需要的依赖包

$ sudo yum install -y yum-utils device-mapper-persistent-data lvm2

使用以下命令设置稳定存储库，以 CentOS7 为例

sudo yum-config-manager --add-repo https://repos.emqx.io/emqx-ce/redhat/centos/7/emqx-ce.repo

安装最新版本的 EMQ X Broker

sudo yum install emqx

安装特定版本的 EMQ X Broker

查询可用版本

$ yum list emqx --showduplicates | sort -r

emqx.x86_64 4.0.0-1.el7

根据第二列中的版本字符串安装特定版本，例如 4.0.0

sudo yum install emqx-4.0.0

启动 EMQ X Broker

直接启动

$ emqx start

emqx 4.0.0 is started successfully!

$ emqx_ctl status

Node 'emqx@127.0.0.1' is started

emqx v4.0.0 is running

systemctl 启动

sudo systemctl start emqx

service 启动

sudo service emqx start

安装完 eqmx 需要做如下配置

在 emqx 服务器上

sudo vi /etc/emqx/emqx.conf

修改为 allow_anonymous = false

修改 acl_nomath = deny

sudo vi /etc/emqx/plugins/emqx_auth_redis.conf

修改 auth.redis.server = 127.0.0.1:6379

修改 auth.redis.password = 123456

重启 emqx sudo systemctl restart emqx

执行命令 emqx_ctl plugins load emqx_auth_redis

4.6 安装 nginx

sudo rpm -ivh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm

sudo yum install nginx

systemctl enable nginx

进入 nginx 的配置文件/etc/nginx/conf.d

vi default.conf

```
upstream mgt-service {
        server 127.0.0.1:8000;
```

```
    }

    server {
        listen 443 ssl;
        ssl_certificate /etc/ssl/server.pem;
        ssl_certificate_key /etc/ssl/server.key;
        access_log /var/log/nginx/host.access.log main;

        location / {
        root /usr/local/console;
            index index.html;
            add_header 'Access-Control-Allow-Methods' 'GET, POST,
OPTIONS';
            add_header 'Access-Control-Allow-Headers'
'DNT,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,C
ontent-Type,Range';
            add_header 'Access-Control-Expose-Headers'
'Content-Length,Content-Range';
            add_header 'Access-Control-Allow-Origin'
'https://m-dev.yytek.net';
        }
```

```
        location /mgt {
            if ($request_method = 'OPTIONS') {
                add_header 'Access-Control-Allow-Origin'
'https://m-dev.yytek.net' always;
                add_header 'Access-Control-Allow-Methods' 'GET,
POST, OPTIONS' always;
                add_header 'Access-Control-Allow-Headers' '-- main
    local function access ()    verify_uri()
    Authorization,DNT,User-Agent,X-Requested-With,If-Modified-Since,
Cache-Control,Content-Type,Range' always;

                add_header 'Access-Control-Max-Age' 1728000
always;
                add_header 'Content-Type' 'text/plain; charset=utf-8';
                add_header 'Content-Length' 0;
                return 204;
    }
            add_header 'Access-Control-Allow-Origin'
'https://m-dev.yytek.net' always;
            add_header 'Access-Control-Allow-Methods' 'GET, POST,
OPTIONS' always;
            add_header 'Access-Control-Allow-Headers'
```

```
tcode,Authorization,DNT,User-Agent,X-Requested-With,If-Modified-Sinc
e,Cache-Control,Content-Type,Range' always;
                add_header                'Access-Control-Expose-Headers'
'Content-Length,Content-Range' always;

                proxy_send_timeout 30m;
                proxy_pass http://mgt-service$request_uri;
                proxy_set_header   X-Forwarded-Port $server_port;
                proxy_set_header   X-Forwarded-Host $host;
                proxy_set_header   X-real-ip $remote_addr;
                proxy_set_header                          X-Forwarded-For
$proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
            }

        location /iam_watermark_info {
                proxy_send_timeout 30m;
                proxy_pass http://mgt-service$request_uri;
                proxy_set_header   X-Forwarded-Port $server_port;
                proxy_set_header   X-Forwarded-Host $host;
                proxy_set_header   X-real-ip $remote_addr;
                proxy_set_header                          X-Forwarded-For
```

$proxy_add_x_forwarded_for;

proxy_set_header X-Forwarded-Proto $scheme;

}

}

启动服务：systemctl start nginx && systemctl enable nginx

5.7 安装身份认证服务

新建/opt/deploy，拷贝安装包到此路径下，执行

tar -zxvf zt.tar.gz

ln -s /opt/deploy/zt/ /usr/local/zt

rm -rf zt.tar.gz

将服务器包加压到这

通过 systemctl start zt 启动服务

application.properties 需要在后面加上 proxy 的信息

server.port=8000

spring.flyway.enabled=true

spring.flyway.baseline-on-migrate=true

spring.flyway.baseline-version=1

spring.flyway.schemas=zero_trust

#spring.flyway.init-sqls=db/migration/init_structure.sql

```
spring.flyway.table=zt_mgt_schema_version
#spring.flyway.tablespace=
spring.flyway.validate-on-migrate=false
spring.flyway.clean-on-validation-error=false
spring.flyway.placeholder-replacement=false

#dds config
dds.general.defaultDataBase=yytek
dds.general.multiTenant=true
dds.general.defaultSchema=zero_trust
dds.general.headerSeparator=.
dds.general.tenantHeaderName=tcode
dds.general.filterUrls=/dds-sample/actuator/health,/error,/actuator,/
health
dds.general.filterUrlsSeparator=;
dds.general.activeDataBase=yytek
dds.general.enableGlobalInterceptor=false
dds.general.shardingDb=false

#mysql config
#dds.database.yytek.username=zerotrust
#dds.database.yytek.password=zerotrust1@3
```

```
#dds.database.yytek.urlParams=useSSL=false&useUnicode=true&c
haracterEncoding=utf-8&serverTimezone=CTT&allowMultiQueries=true
#dds.database.yytek.jdbcUrl=jdbc:mysql://localhost:3306
#dds.database.yytek.driverClassName=com.mysql.cj.jdbc.Driver

#dds.database.yytek.jdbcUrl=jdbc:p6spy:mysql://localhost:3306
#dds.database.yytek.driverClassName=com.p6spy.engine.spy.P6Sp
yDriver

# postgreSql\u914D\u7F6E
dds.database.yytek.username=postgres
dds.database.yytek.password=123456
dds.database.yytek.jdbcUrl=jdbc:postgresql://localhost:5432
dds.database.yytek.driverClassName=org.postgresql.Driver
dds.database.yytek.urlParams=currentSchema=zero_trust

spring.redis.host=localhost
spring.redis.port=6379
spring.redis.database=0
spring.redis.password=123456
#spring.redis.jedis.pool.max-wait=
#spring.redis.jedis.pool.max-active=
```

mybatis-plus.global-config.db-config.logic-delete-field=deleted

mybatis-plus.global-config.db-config.logic-delete-value=1

mybatis-plus.global-config.db-config.logic-not-delete-value=0

aliyun.sms.ak=LTAI4GJ2rzydYVqubg9XcMTo

aliyun.sms.sk=ocPTa6jgvWYxoxZNzCRqznyBLPQVwm

aliyun.sms.signName=\u4E91\u6D8C\u96F6\u4FE1\u4EFB

aliyun.sms.tempPwdTemplate=SMS_204277571

aliyun.sms.oneTimePwdTemplate=SMS_206539350

one time password validity time interval

aliyun.sms.oneTimePwdValidity=300

mqtt.url=tcp://localhost:1883

mqtt.clientId=mgt

mqtt.username=mgt

mqtt.password=mgt

mqtt.topics=biz,gateway

send otp rate limit, 60 seconds default

api.rate.limit.otpDuration=60

proxy.url=https://m-qa.yytek.net

proxy.ak=LTAI4GJ2rzydYVqubg9XcMToy

proxy.sk=ocPTa6jgvWYxoxZNzCRqznyBLPQVwmy

proxy.salt=vKgADfZyY9uUBUk4J2seZg==

mail 配置

spring.mail.host=smtp.yytek.com

spring.mail.port=465

spring.mail.username=zerotrust@yytek.com

spring.mail.password=Yytek123qwe!@#

spring.mail.properties.mail.smtp.auth=true

spring.mail.properties.mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory

spring.mail.properties.mail.smtp.socketFactory.port=465

spring.mail.test-connection=true

根证书配置

ca.country=CN

ca.street=Haidian

ca.organization=Yytek

ca.organizationUnit=Security

ca.commandName=RootCA

ca.duration=20

启动服务：systemctl start zt && systemctl enable zt

4.8 安装前端页面

tar -zxvf console.tar.gz

ln -s /opt/deploy/dist/ /usr/local/console

rm -rf /root/deploy/console.tar.gz

4.9 安装安全访问控制服务

mkdir -p /root/deploy/sdp

tar -zxvf sdp.tgz -C /opt/deploy/sdp/

rm -rf sdp.tgz

ln -s /opt/deploy/sdp/ /usr/local/sdp

配置

redisAddr=127.0.0.1:6379

redisDB=0

redisAuth=123456

redisMaxidle=2

redisMaxActive=200

udpAddr=0.0.0.0:8111

httpAddr=:8000

httpsAddr=:443

maxHttpConn=5000

maxRoutine=1000

myiplist=127.0.0.1,192.168.1.110

openPort=22,8111

protectedPort=443

trustedIpList=192.168.0.62

trustedIpPort=8000,1883,6379,5432

iamAddr=http://127.0.0.1:8000

mqttEnabled=true

mqttBroker=tcp://127.0.0.1:1883

mqttUsername=mgt

mqttPassword=mgt

mqttClientId=sdp

sdpExpireTime=3600

ipExpireTime=36000

isController=true

isDebug=false

isKnockProxy=false

proxyIP=192.168.0.110 》配 ip 不配域名

syncIPinterval=60

performanceTest=false

启动服务：systemctl start sdp && systemctl enable sdp

5. 部署接入网关

5.1 IP 配置

cd /etc/sysconfig/network-scripts/

编辑网卡文件

TYPE="Ethernet"

PROXY_METHOD="none"

BROWSER_ONLY="no"

BOOTPROTO="static"

DEFROUTE="yes"

IPV4_FAILURE_FATAL="no"

IPV6INIT="yes"

IPV6_AUTOCONF="yes"

IPV6_DEFROUTE="yes"

IPV6_FAILURE_FATAL="no"

IPV6_ADDR_GEN_MODE="stable-privacy"

NAME="enp2s0"

UUID="944788d6-78f3-4599-8b3d-91370c711b33"

DEVICE="enp2s0"

ONBOOT="yes"

IPADDR=192.168.x.x

NETMASK=255.255.255.0

GATEWAY=192.168.x.x

DNS1=8.8.8.8

注： IPADDR 就是静态 IP，NETMASK 是子网掩码，GATEWAY 就是网关或者路由地址 BOOTPROTO=static 设置为静态

5.2 安装控制服务

mkdir -p /root/deploy/sdp

tar -zxvf sdp.tgz -C /opt/deploy/sdp/

rm -rf sdp.tgz

ln -s /opt/deploy/sdp/ /usr/local/sdp

配置

redisAddr=127.0.0.1:6379

redisDB=0

redisAuth=123456

redisMaxidle=2

redisMaxActive=200

udpAddr=0.0.0.0:8111

httpAddr=:8000

httpsAddr=:443

maxHttpConn=5000

maxRoutine=1000

myiplist=127.0.0.1,192.168.1.110

openPort=22,8111

protectedPort=443

trustedIpList=192.168.0.62

trustedIpPort=8000,1883,6379,5432

iamAddr=http://127.0.0.1:8000

mqttEnabled=true

mqttBroker=tcp://127.0.0.1:1883

mqttUsername=mgt

mqttPassword=mgt

mqttClientId=sdp

sdpExpireTime=3600

ipExpireTime=36000

isController=false

isDebug=false

isKnockProxy=false

proxyIP=192.168.0.110 》配 ip 不配域名

syncIPinterval=60

performanceTest=false

启动服务：systemctl start sdp && systemctl enable sdp

5.3 安装安全路由服务

mkdir /opt/deploy/traefik

cd /opt/build, tar -zxvf traefik.tar.gz -C /opt/deploy/traefik

设置配置文件

```
entryPoints:
  web:
    address: ":80"
  websecure:
    address: ":443"
providers:
  http:
    endpoint:
      # 从 iam 服务器动态获取路由配置，每秒轮训一次
      - "http://192.168.0.110:8000/mgt/traefik/dynamic_config"
    pollInterval: "1s"
log:
  level: INFO
  filePath: "/usr/local/traefik/traefik.log"
api:
  dashboard: true
```

```
mqtt:
  broker: tcp://192.168.0.110:1883
  username: mgt
  password: mgt
  clientId: traefik
```

从 deploy 安装程序（./install.sh）

在/usr/local/traefik 下面新建 cert 文件夹，将 url 的证书拷贝进入，将名字修改成 traefik.pem traefik.key

路由水印需要部署 nginx，在 default.conf 中添加如下信息：当前静态页面放置在 traefik 的部署文件夹中

```
server {
    listen       81;
    #charset koi8-r;
    access_log  /var/log/nginx/host.access.log  main;
    location / {
        root   /usr/local/traefik/deploy/nginx;
        index  index.html index.htm;
    }
}
```

启动服务：systemctl start traefik && systemctl enable traefik

5.4 安装接入网关服务

装包拷贝到/opt/deploy/zt-gateway/

在/opt/deploy/zt-gateway/config 目录下修改 gateway.yml 文件

修改配置文件

```
# Tunnel
tunnel:
  protocol: relay
  transport: tls
  host: 0.0.0.0
  port: 12345
  sso: true
  inter: true
  cert:
    certFile: cert/cert.pem
    keyFile: cert/private/key.pem


# Identity and Access Management (IAM) Center
iam:
  address: http://192.168.0.110
  port: 8000
```

```
    # The Standard for IoT Messaging
    mqtt:
      broker: mqtts://yytek.yytek.net:28083
      tlsInsecure: false
      clientId: gateway
      username: mgt
     # username: tenant_yytek
     #                                                    password:
aw66edssjm2vcc1syifpf3o7dg22v8oktcy3lx3f7avvwngu6sb7u3c4nbp5
p51y
      password: mgt
      keepAlive: 60
      pingTimeout: 1
      tls: false
      tenantId: yytek
    sdp:
      certPath: /opt/deploy/sdp/config/gm-udp.cer
      keyPath: /opt/deploy/sdp/config/gm-udp.key

    gmtls:
      load: true
      rootCert: /opt/deploy/zt-gateway/cert/CA.pem
```

```
    signCert: /opt/deploy/zt-gateway/cert/SS.cert.pem
    signKey: /opt/deploy/zt-gateway/cert/SS.key.pem
    encryptCert: /opt/deploy/zt-gateway/cert/SE.cert.pem
    encryptKey: /opt/deploy/zt-gateway/cert/SE.key.pem
log:
    fileName: "logs/zt-gateway.log"
    maxSize: 100
    maxBackups: 5
    maxAge: 30
    level: INFO
    console: true
httpServer:
    httpAddr: :8000
```

启动方式， 执行命令 /opt/deploy/zt-gateway/zt-gateway

启动服务：systemctl start zt-gateway && systemctl enable zt-gateway