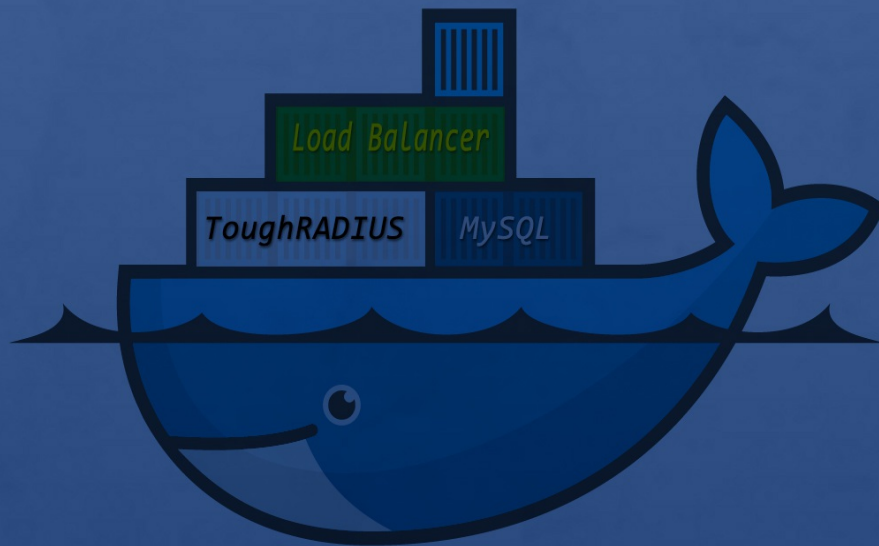


ToughRADIUS *Book*



docker

docs.toughradius.net

目錄

Introduction	1.1
常见问题	1.2
快速开始	1.3
数据库安装配置	1.4
Windows 安装运行	1.5

TOUGHRADIUS

ToughRADIUS是一个开源的Radius服务软件，支持标准RADIUS协议（RFC 2865, RFC 2866），提供完整的AAA实现。支持灵活的策略管理，支持各种主流接入设备并轻松扩展，具备丰富的计费策略支持。

至 6.x 版本开始，基于Java语言重新开发。仅提供纯粹的RADIUS 引擎功能，不再提供UI管理系统，但是更利于第三方扩展开发。

ToughRADIUS类似于freeRADIUS，但它使用起来更简单，更易于扩展开发。

常见问题

如何进行 **ToughRADIUS** 的扩展开发呢？

首先要看扩展开发的目的，如果是要实现对接已有的管理系统，可以在已有的系统中直接管理操作 **ToughRADIUS**的数据库，**ToughRADIUS**的数据库只有简单的数据结构，操作起来非常容易。

如果是要扩展**RADIUS**协议适配其他设备，可以根据相关文档修改**ToughRADIUS**的源码来实现。

关于拨号**691**，为什么明明密码正确，拨号认证还会提示**691** 密码错误呢？

691 是认证协议定义的一个通用错误吗，但并不表示一定是密码错误，他表示的是后端认证出了错误，导致认证失败，认证失败的原因可能是账号过期，密码错误，账号不存在，**MAC**或**VLAN**绑定错误，账号已经在线等等，可以通过系统日志去查询用户到底是什么原因导致的认证失败，在系统日志界面，输入用户名作为关键字即可查询。

快速开始

软件安装配置

系统环境依赖

操作系统

- Linux
- Windows
- MacOS

jdk 版本: 1.8+

数据库服务器(默认): MySQL/MariaDB

数据库配置

请参考 [数据库配置](#)

运行主程序

```
java -jar -Xms256m -Xmx1024G /opt/toughradius-latest.jar --spring.profiles.active=prod
```

注意jar文件 (toughradius-latest.jar) 的路径

Linux 系统服务配置

以 centos 7 为例

配置文件 (见 `scripts` 目录)

```
/etc/toughradius.env  
/usr/lib/systemd/system/toughradius.service
```

运行以下指令

```
systemctl enable toughradius  
systemctl start toughradius
```

数据库安装配置

数据库服务器(默认): MySQL/MariaDB

首先确保数据库服务器已经正常运行

创建数据库

```
create database toughradius DEFAULT CHARACTER SET utf8 COLLATE utf8mb4_unicode_ci;  
GRANT ALL ON toughradius.* TO raduser@'127.0.0.1' IDENTIFIED BY 'radpwd' WITH GRANT OPTION;  
FLUSH PRIVILEGES;
```

创建表

```
create table if not exists tr_bras  
(  
    id bigint auto_increment primary key,  
    identifier varchar(128) null,  
    name varchar(64) not null,  
    ipaddr varchar(32) null,  
    vendor_id varchar(32) not null,  
    secret varchar(64) not null,  
    coa_port int not null,  
    auth_limit int null,  
    acct_limit int null,  
    status enum('enabled', 'disabled') null,  
    remark varchar(512) null,  
    create_time datetime not null  
);  
  
create index ix_tr_bras_identifier on tr_bras (identifier);  
  
create index ix_tr_bras_ipaddr on tr_bras (ipaddr);  
  
create table if not exists tr_config  
(  
    id bigint auto_increment primary key,  
    type varchar(32) not null,  
    name varchar(128) not null,  
    value varchar(255) null,  
    remark varchar(255) null  
);  
  
create table if not exists tr_subscribe
```

```

(
    id bigint auto_increment primary key,
    node_id bigint default 0 not null,
    subscriber varchar(32) null,
    realname varchar(32) null,
    password varchar(128) not null,
    domain varchar(128) null,
    addr_pool varchar(128) null,
    policy varchar(512) null,
    is_online int null,
    active_num int null,
    bind_mac tinyint(1) null,
    bind_vlan tinyint(1) null,
    ip_addr varchar(32) null,
    mac_addr varchar(32) null,
    in_vlan int null,
    out_vlan int null,
    up_rate bigint null,
    down_rate bigint null,
    up_peak_rate bigint null,
    down_peak_rate bigint null,
    up_rate_code varchar(32) null,
    down_rate_code varchar(32) null,
    status enum('enabled', 'disabled') null,
    remark varchar(512) null,
    begin_time datetime not null,
    expire_time datetime not null,
    create_time datetime not null,
    update_time datetime null
);

create index ix_tr_subscribe_create_time
on tr_subscribe (create_time);

create index ix_tr_subscribe_expire_time
on tr_subscribe (expire_time);

create index ix_tr_subscribe_status
on tr_subscribe (status);

create index ix_tr_subscribe_subscriber
on tr_subscribe (subscriber);

create index ix_tr_subscribe_update_time
on tr_subscribe (update_time);

```

插入测试数据

```

INSERT INTO toughradius.tr_bras
(identifier, name, ipaddr, vendor_id, secret, coa_port, auth_limit, acct_limit, STATUS,
remark, create_time)
VALUES ('radius-tester', 'radius-tester', '127.0.0.1', '14988', 'secret', 3799, 1000, 1
000, NULL, '0', '2019-03-01 14:07:46');

```

```

INSERT INTO toughradius.tr_subscribe
(node_id, subscriber, realname, password, domain, addr_pool, policy, is_online, active
_num,
bind_mac, bind_vlan, ip_addr, mac_addr, in_vlan, out_vlan, up_rate, down_rate, up_peak_
rate, down_peak_rate, up_rate_code,
down_rate_code, status, remark, begin_time, expire_time, create_time, update_time)
VALUES (0, 'test01', '', '888888', null, null, null, null, 10, 0, 0, '', '', 0, 0, 10.
000, 10.000, 100.000, 100.000,
'10', '10', 'enabled', '', '2019-03-01 14:13:02', '2019-03-01 14:13:00', '2019-
03-01 14:12:59', '2019-03-01 14:12:56');

```


Windows 安装运行

java 运行环境

TOUGHRADIUS 依赖的 Java 版本最低为1.8，下载地址 [Java1.8](#)

获得 TOUGHRADIUS 的编译文件 toughradius-latest.jar

- 可以自行通过Maven工具或Java Ide工具进行编译
- 从官方 QQ 群（247860313）获取最新编译好的jar文件

准备 MySQL 数据库环境

请参考 [数据库配置](#)

自定义配置文件

在运行目录下新建配置文件 `application.properties`, 根据实际情况自行修改配置，特别是数据库部分

```
server.port = 1816
server.security.require-ssl=true
server.ssl.key-store-type=PKCS12
server.ssl.key-store=classpath:toughradius.p12
# The password used to generate the certificate
server.ssl.key-store-password=toughstruct
# The alias mapped to the certificate
server.ssl.key-alias=toughradius

server.http2.enabled=true
server.address=0.0.0.0
server.connection-timeout=600
server.server-header=toughradius/1.0
server.tomcat.accesslog.enabled=true
server.tomcat.accesslog.directory=/var/log/toughradius
spring.application.name=toughradius

mybatis.config-locations=classpath:mybatis-config.xml
mybatis.mapper-locations=classpath:mapper/*.xml

# logging
logging.config=classpath:logback-dev.xml
logging.level.root=INFO
logging.level.org.springframework.web=INFO
```

```

# http
spring.http.encoding.force=true
spring.http.encoding.charset=UTF-8
spring.http.encoding.enabled=true
server.tomcat.uri-encoding=UTF-8
spring.output.ansi.enabled=detect
spring.jackson.time-zone=GMT+8
spring.jackson.date-format=yyyy-MM-dd HH:mm:ss

# component
spring.cache.type=ehcache
spring.cache.ehcache.config=classpath:/ehcache.xml

#radius config
org.toughradius.running=true
org.toughradius.authport=${RADIUSD_AUTH_PORT:1812}
org.toughradius.acctport=${RADIUSD_ACCT_PORT:1813}
org.toughradius.rejectdelayEnabled=${RADIUSD_REJECT_DELAY_ENABLED:1}
org.toughradius.rejectdelayTimes=${RADIUSD_REJECT_DELAY_TIMES:10}
org.toughradius.rejectdelay=${RADIUSD_REJECT_DELAY:3}
org.toughradius.trace=${RADIUSD_DEBUG:1}
org.toughradius.authPool=${RADIUSD_AUTH_POOL:32}
org.toughradius.acctPool=${RADIUSD_ACCT_POOL:32}
org.toughradius.interimUpdate=120
org.toughradius.maxSessionTimeout=86400
org.toughradius.statDir=${RADIUSD_STAT_DIR:/var/toughradius/data/stat}
org.toughradius.ticketDir=${RADIUSD_TICKET_DIR:/var/toughradius/data/ticket}
org.toughradius.statfile=${RADIUSD_STAT_FILE:/var/toughradius/radiusd_stat.json}
org.toughradius.isBillInput=true
org.toughradius.isBillBackFlow=true
org.toughradius.allowNegative=${RADIUSD_ALLOW_NEGATIVE:false}

application.apikey = toughradius
application.apisecret = toughradius
application.version = v6.0.1

logging.config=classpath:logback-prod.xml

# database config
spring.datasource.url=${RADIUSD_RPCD_JDBC_URL:jdbc:mysql://127.0.0.1:3306/toughradius?serverTimezone=Asia/Shanghai&useUnicode=true&characterEncoding=utf-8&allowMultiQueries=true}
spring.datasource.username=${RADIUSD_RPCD_DBUSER:raduser}
spring.datasource.password=${RADIUSD_RPCD_DBPWD:radpwd}
spring.datasource.max-active=${RADIUSD_RPCD_DBPOOL:120}
spring.datasource.driver-class-name=com.mysql.jdbc.Driver

```

运行

在运行目录下新建 **startup.bat** 文件，内容为

```
java -jar -Xms256M -Xmx1024M toughradius-latest.jar
```

双击运行 **startup.bat** 即可

附注

如果熟悉 **Springboot** 程序部署的可以忽略以上说明