

ToughRADIUS

Book

docs.toughradius.net

Table of Contents

1. [Introduction](#)
2. [ToughRADIUS文档](#)
 - i. [项目简介](#)
 - ii. [快速指南](#)

欢迎阅读ToughRADIUS用户文档。

及时向我们反馈问题是对我们最大的支持。

向我们发送邮件：support@toughstruct.com

加入我们的QQ交流群组：464025428, 247860313

访问我们的网站：<http://www.toughstruct.com>

最简单实用的 ToughRADIUS 说明书

ToughRADIUS是一个AAA应用软件。

ToughRADIUS支持标准RADIUS协议，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。

ToughRADIUS支持使用Oracle, MySQL, PostgreSQL, MSSQL等主流数据库存储用户数据，支持数据缓存，提供高性能的支持。

ToughRADIUS采用Docker容器模式安装部署，简单方便。

ToughRADIUS提供了RADIUS核心服务引擎与Web管理控制台,用户自助服务三个子系统，核心服务引擎提供高性能的认证计费服务，Web管理控制台提供了界面友好，功能完善的管理功能。用户自助服务系统提供了一个面向终端用户的网上服务渠道。

ToughRADIUS快速指南

安装部署

ToughRADIUS主要采用了Docker镜像部署的模式，ToughRADIUS的镜像基础是CentOS 7。

我们可以把Docker看作一个软件集装箱，半世纪之前，集装箱发挥了巨大的力量，改变了整个运输产业，也改变了人们的生活。而Docker就类似这样一个集装箱工具，只不过他封装的是软件。

还记得linux安装lamp的经历吗？现在可以对各种安装配置apache，php等繁琐的工作说再见了。

我们把ToughRADIUS相关的配置，运行依赖环境等全部打包在一个“Docker集装箱”里，我们只需要在我们的服务器上简单的安装一个支持运行“Docker集装箱”的环境，那么我们不用去折腾各种运行环境搭建就能简单的让ToughRADIUS跑起来。

通常我们把封装了软件应用的“Docker集装箱”叫做镜像，有点类似你可能了解的ISO文件。

Docker 安装

CentOS 6

```
$ sudo yum install http://mirrors.yun-idc.com/epel/6/i386/epel-release-6-8.noarch.rpm  
$ sudo yum install docker-io  
$ sudo service docker start
```

CentOS 7

```
$ sudo yum install docker  
$ sudo service docker start
```

Ubuntu

```
$ sudo apt-get update  
  
$ sudo apt-get install docker.io  
  
$ sudo ln -sf /usr/bin/docker.io /usr/local/bin/docker  
  
$ sudo sed -i '$acomplete -F _docker docker' /etc/bash_completion.d/docker.io
```

Windows

请下载Windows安装文件：<https://github.com/boot2docker/windows-installer/releases/download/v1.8.0/docker-install.exe>

注意：Window下的Docker服务是借助虚拟机来实现的，性能上远不如使用linux主机。

Docker 容器创建

确认Docker服务已经运行，通过以下指令创建ToughRADIUS的容器实例，当看到一串hash打印时，说明容器创建成功了。

```
$ docker run -d --name trserver --net host toughstruct/toughradius:latest
```

这样我们的服务就已经运行了。我们可以通过浏览器来访问我们的应用了。

营业管理：<http://ipaddr:1816> 管理权限 admin/root

自助服务：<http://ipaddr:1817>

系统管理：<http://ipaddr:1819> 管理权限 ctlman/ctlroot

容器的基本管理

启动容器

```
$ docker start trserver
```

停止容器

```
$ docker stop trserver
```

重启容器

```
$ docker restart trserver
```

查看容器日志，*Ctrl+c*退出

```
$ docker logs trserver
```