

安全 OTA 升级系统——部署文档

陈晨 519021910176

邹涛旭 519021910075

孟详贺 519021910190

王思皓 519021910160

伍杨子涵 519021910019

2022 年 6 月 17 日

目录

1 环境配置	2
2 云服务器部署	3
2.1 系统代码移植	3
2.2 系统环境配置	3
2.3 uwsgi 与 nginx 安装	3
2.4 uwsgi 启动文件编写	3
2.5 nginx 配置文件改写	4
2.6 系统测试	5
3 附录	5
3.1 配置完成的 conda 环境	5

1 环境配置

开发者注：本项目环境配置步骤甚为复杂，开发团队在摸索环境配置方法的过程中前后总共花费了不下600分钟，在进行环境配置的过程中请做好十足的心理准备。

本次项目配置采用 anaconda 进行环境管理，因此下面的配置过程均基于 conda 包管理，若不采用则无法保证最终环境配置成功。配置完成后的 conda 环境见附录3.1。

操作系统	Ubuntu 20.04/22.04
版本管理	Conda 4.13.0
编译器	Python 3.6.13

表 1: 项目环境

直接导入预先配置好的环境UP36tuf到 conda 中，随后单独配置 tuf 和 uptane 两个开发库。

```
$ cd Mobile-OTA/Flask/  
$ conda env create -f requirements/UP36tuf.yaml // 导入预先配置好的环境包  
$ conda activate UP36tuf // 激活环境  
$ pip install -r requirements/dev-requirement.txt // 单独安装开发模式的 tuf和uptane包
```

如果一切顺利，则环境配置完成，可运行程序；如果出现错误如图1，则需要考虑手动安装 tuf 库。步骤如下：

```
$ pip uninstall tuf // 先删除配置不成功的 tuf库  
// 正确的 tuf库版本及状态在 $ conda list中应该形如  
// tuf      0.10.0      dev_0      <develop>  
$ cd <AnyDir> // 打开任意目录  
$ git clone git@github.com:awwad/tuf.git // 推荐使用SSH方式，HTTP方式不稳定  
$ cd tuf  
$ pip install --editable . // 需以开发模式安装 tuf库  
$ cd Mobile-OTA/Flask/  
$ pip uninstall uptane  
$ pip install --editable . // 保险起见重新安装 uptane库，一般不需要重装
```

若此时\$ conda list命令显示的 tuf 和 uptane 库与附录3.1中一致，则配置成功。若配置依然存在问题，请在 GitHub 上联系我们。

```
(m1) root@iZuf6djj5Zr9103rhco7eiZ:/home/chenzhen/Mobile-OTA/python-tuf# pip
install --editable .
Looking in indexes: http://mirrors.cloud.aliyuncs.com/pypi/simple/
Obtaining file:///home/chenzhen/Mobile-OTA/python-tuf
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing wheel metadata ... done
Requirement already satisfied: requests>=2.19.1 in /root/miniconda3/envs/m1
/lib/python3.6/site-packages (from tuf==1.1.0) (2.27.1)
ERROR: Could not find a version that satisfies the requirement securesystem
slib>=0.22.0 (from tuf) (from versions: 0.10.0, 0.10.1, 0.10.2, 0.10.3, 0.1
0.4, 0.10.5, 0.10.6, 0.10.7, 0.10.8, 0.10.9, 0.10.10, 0.10.11, 0.11.0, 0.11
.1, 0.11.2, 0.11.3, 0.12.0, 0.12.1, 0.12.2, 0.13.0, 0.13.1, 0.14.0, 0.14.1,
0.14.2, 0.15.0, 0.16.0, 0.17.0, 0.18.0, 0.19.0, 0.20.0, 0.20.1, 0.21.0)
ERROR: No matching distribution found for securesystemslib>=0.22.0
```

图 1: tuf 库安装出错

2 云服务器部署

我们整个系统的前端开发使用 django 框架，在本地的调试成功后，计划在阿里云上使用 uwsgi+nginx 反向代理模式来对整个前端页面进行部署和调试，前端部署过程主要分为以下几个方面：

2.1 系统代码移植

由于项目整体托管到 github 上，直接采用 git clone 的方式从云端移植项目代码。

2.2 系统环境配置

从与代码一起移植到与服务器上的 requirement.txt 文件中获取项目运行所需的全部依赖库与依赖文件。

2.3 uwsgi 与 nginx 安装

使用如下命令：

```
$ apt install uwsgi
```

```
$ apt install nginx
```

进行 uwsgi 与 nginx 进行安装，安装后通过访问云服务器外网 ip 即可看到 nginx 的欢迎界面。

2.4 uwsgi 启动文件编写

在系统目录/Mobile-OTA/Django/djangoProject 下编辑 uwsgi 启动文件 uwsgi.ini，使整个项目依托于启动 swgi.py 而非 runserver 的方式启动。uwsgi.ini 文件格式如下：

```
[uwsgi]
```

```
# uwsgi 启动时所使用的地址与端口，端口可以使用其他端口
```

```

socket=127.0.0.1:8080
# 指向网站的项目根目录
chdir=/home/chenchen/Mobile-OTA/Django/
#python 启动程序文件
wsgi-file=/home/chenchen/Mobile-OTA/Django/djangoProject/wsgi.py
master=true
# 处理器数目
processes=4
# 线程数
threads=2
# 编程使用的 python 版本与 django 安装位置
plugins=python3
pythonpath=/root/miniconda3/envs/UP36tuf/lib/python3.6/site-packages

```

2.5 nginx 配置文件改写

uwsgi 改写后在 uwsgi.ini 同目录下通过命令 `uwsgi -d -ini uwsgi.ini` 即可代替 `runserver` 后台启动网站，但是想要在外网访问地址则需要使用 `nginx` 监听 80 端口进行反向代理来实现动静分离。因此通过命令 `vim /etc/nginx/sites-available/default` 改写 `nginx` 默认设置来实现反向代理功能：

```

root /home/chenchen/Mobile-OTA/Django;
# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;
# 服务器名称，在此设置为公网地址      server_name 139.196.40.15;
location /static {# 静态资源目录
alias /home/chenchen/Mobile-OTA/Django/static;
}
location /media { # 媒体文件目录
alias /home/chenchen/Mobile-OTA/Django/media;
}
location / {# uwsgi 项目目录
# First attempt to serve request as file, then
# as directory, then fall back to displaying a 404.
include uwsgi_params;
uwsgi_pass 127.0.0.1 : 8080;
# 与 uwsgi.ini 中端口一致
uwsgi_param UWSGI_HTTP_ROOT /home/chenchen/Mobile - OTA/Django/;
}

```

2.6 系统测试

在 nginx 的 default 文件修改完毕之后,使用命令 `service nginx restart` 来重启 nginx 服务,在/Django/djangoProject 文件夹下,使用命令 `uwsgi -d -ini uwsgi.ini` 来后台启动网站服务。检测后与 runserver 相同且无异常,项目前端部署完成。

3 附录

3.1 配置完成的 conda 环境

```
$ conda list
```

# Name	Version	Build	Channel
_libgcc_mutex	0.1	main	
_openmp_mutex	4.5	1_gnu	
anyio	3.6.1	pypi_0	pypi
asgiref	3.4.1	pypi_0	pypi
asn1crypto	1.5.1	pypi_0	pypi
async-generator	1.10	pypi_0	pypi
ca-certificates	2022.3.29	h06a4308_1	
canonicaljson	1.1.4	pypi_0	pypi
certifi	2020.6.20	pyhd3eb1b0_3	
cffi	1.11.5	pypi_0	pypi
charset-normalizer	2.0.12	pypi_0	pypi
click	8.0.4	pypi_0	pypi
contextvars	2.4	pypi_0	pypi
cryptography	2.3	pypi_0	pypi
dataclasses	0.8	pypi_0	pypi
distlib	0.3.4	pypi_0	pypi
django	3.2.13	pypi_0	pypi
filelock	3.4.1	pypi_0	pypi
flask	2.0.3	pypi_0	pypi
flask-cors	3.0.10	pypi_0	pypi
frozendict	2.3.2	pypi_0	pypi
h11	0.12.0	pypi_0	pypi
httpcore	0.14.7	pypi_0	pypi
httpx	0.22.0	pypi_0	pypi
idna	3.3	pypi_0	pypi
immutables	0.18	pypi_0	pypi
importlib-metadata	4.8.3	pypi_0	pypi

importlib-resources	5.4.0	pypi_0	pypi
iso8601	1.0.2	pypi_0	pypi
itsdangerous	2.0.1	pypi_0	pypi
jinja2	3.0.3	pypi_0	pypi
ld_impl_linux-64	2.35.1	h7274673_9	
libffi	3.3	he6710b0_2	
libgcc-ng	9.3.0	h5101ec6_17	
libgomp	9.3.0	h5101ec6_17	
libstdcxx-ng	9.3.0	hd4cf53a_17	
markupsafe	2.0.1	pypi_0	pypi
ncurses	6.3	h7f8727e_2	
openssl	1.1.1n	h7f8727e_0	
packaging	21.3	pypi_0	pypi
pip	21.2.2	py36h06a4308_0	
platformdirs	2.4.0	pypi_0	pypi
pluggy	0.13.1	pypi_0	pypi
py	1.11.0	pypi_0	pypi
pyasn1	0.4.4	pypi_0	pypi
pycparser	2.21	pypi_0	pypi
pycrypto	2.6.1	pypi_0	pypi
pynacl	1.2.1	pypi_0	pypi
pyparsing	3.0.8	pypi_0	pypi
python	3.6.13	h12debd9_1	
pytz	2022.1	pypi_0	pypi
readline	8.1.2	h7f8727e_1	
requests	2.27.1	pypi_0	pypi
rfc3986	1.5.0	pypi_0	pypi
securesystemslib	0.21.0	pypi_0	pypi
setuptools	58.0.4	py36h06a4308_0	
simplejson	3.17.6	pypi_0	pypi
six	1.16.0	pypi_0	pypi
sniffio	1.2.0	pypi_0	pypi
sqlite	3.38.2	hc218d9a_0	
sqlparse	0.4.2	pypi_0	pypi
tk	8.6.11	h1ccaba5_0	
tox	3.1.2	pypi_0	pypi
tuf	0.10.0	dev_0	<develop>
typing-extensions	4.1.1	pypi_0	pypi

uptane	0.1.0	dev_0	<develop>
urllib3	1.26.9	pypi_0	pypi
virtualenv	20.14.1	pypi_0	pypi
werkzeug	2.0.3	pypi_0	pypi
wheel	0.37.1	pyhd3eb1b0_0	
xz	5.2.5	h7b6447c_0	
zipp	3.6.0	pypi_0	pypi
zlib	1.2.12	h7f8727e_2	