**项目系统网站环境搭建**

Steven

Version 1.0

2019/7/2

Contents

[Install CentOS7 1](#_Toc3292991)

[开机进入CentOS自动启动网络连接 1](#_Toc3292992)

[Install PostgreSQL10 1](#_Toc3292993)

[Create Database, User and Password, and can connect it. 2](#_Toc3292994)

[manually Compile and Install Python3.64 3](#_Toc3292995)

[Install Python3 venv, Install Django, Create Django project and app, run server. 3](#_Toc3292996)

[在Python3 venv中安装uwsgi，开机自启动uwsgi ini 文件 5](#_Toc3292997)

[设定代理服务器Nginx 6](#_Toc3292998)

目前已经有一个硬盘空间有60G的KVM虚拟机CentOS7文件centos7.5\_project-clone.qcow2，网页搭建环境情况如下所示：

CentOS7 user: root, password: 1234qwer!@#$QWER

PostgreSQL user: postgres, password: 1234qwer!@#$QWER， port: 5432

# Install CentOS7

已经有安装好的虚拟机CentOS7

# 开机进入CentOS自动启动网络连接

虚拟机CentOS7已经设定好

1. cd /etc/sysconfig/network-scripts/
2. vim ifcfg-xxxxx
3. ONBOOT=yes
4. 重开机

# Install PostgreSQL10

虚拟机CentOS7已经安装好了

* Install the repository RPM:

yum install https://download.postgresql.org/pub/repos/yum/10/redhat/rhel-7-x86\_64/pgdg-centos10-10-2.noarch.rpm

* Install the client packages:

yum install postgresql10

* Optionally install the server packages:

yum install postgresql10-server

* Optionally initialize the database and enable automatic start:

/usr/pgsql-10/bin/postgresql-10-setup initdb

systemctl enable postgresql-10

systemctl start postgresql-10

You will need to become the operating system user under which PostgreSQL was installed (usually postgres) to create the first user account.

# Create Database, User and Password, and can connect it.

此步虚拟机CentOS7未实现。

* create postgreSQL user

-bash-4.2$ createuser -P -s -e tmcs

Enter password for new role: tmcs

Enter it again: tmcs

CREATE ROLE joe PASSWORD 'md5b5f5ba1a423792b526f799ae4eb3d59e'

SUPERUSER CREATEDB CREATEROLE INHERIT LOGIN;

如果要删除一个用户名，有两种方法：

1. 使用postgres用户登录后，使用dropuser命令。

2. 使用postgres用户登录后，执行psql命令，下SQL语句删除：drop user 用户名；。

查看数据库所有用户，使用postgres用户登录后，执行psql命令成功后，输入\du回车。

* create database

-bash-4.2$ createdb tmcs

如果要删除一个数据库，有两种方法：

1. 使用postgres用户登录后，使用dropdb命令。

2. 使用postgres用户登录后，执行psql命令，下SQL语句删除：drop database 数据库名；。

* 如果数据库所在的服务器是远程服务器，应该设定指定数据库输入密码。

[root@localhost ~]# cat /var/lib/pgsql/10/data/pg\_hba.conf

host all all 127.0.0.1/32 ident

-->change into

host tmcs tmcs 192.168.1.3/32 password

IP 192.168.1.3 is connection client.

* restart postgres server

[root@localhost /]# systemctl restart postgresql-10

# manually Compile and Install Python3.64

虚拟机CentOS7已经安装好了

已安装路径：/user/python3.64/

[root@localhost sites-enabled]# yum install epel-release

[root@localhost sites-enabled]# yum groupinstall "Development Tools"

如果出现以下错误，执行以下命令：

Maybe run yum groups mark install

No packages in any requested group availabe to install or update

[root@localhost sites-enabled]# yum groups mark install "Development Tools"

[root@localhost sites-enabled]# yum groups mark convert "Development Tools"

[root@localhost sites-enabled]# yum groupinstall "Development Tools"

[root@localhost sites-enabled]# yum install python-devel libxml2-devel pcre openssl-devel zlib-devel bzip2-devel ncurses-devel zx-devel readline-devel tk-devel gdbm-devel db4-devel libcap-devel libffi-devel

a. yum install sqlite-devel

b. mkdir /usr/python3.64

c. install python 3.64

(./configure --prefix=/usr/python3.64 --enable-optimizations & make & make test & make install)

# Install Python3 venv, Install Django, Create Django project and app, run server.

此步虚拟机CentOS7未实现。

建议网站的项目在/home的用户目录下。

**1. 假设，用户名是tmcs，用户主目录/home/tmcs/。然后我们在主目录下创建一个项目目录名tmcs。**

$ mkdir tmcs

**2. 在项目目录下创建一个venv环境。**

$ cd tmcs

$ /usr/python3.64/bin/python3.6 -m venv venv

$ source venv/bin/activate

**3. 安装Django，psycopg2等需要的包。**

$ pip install django

$ pip install psycopg2

**4. 在项目目录下， 创建一个Django project aemslite**

$ django-admin startproject **aemslite**

**5 Let’s verify your Django project works.**

$ python manage.py runserver

**6 Creating the Polls app**

$ python manage.py startapp website

**7 configure settings.py file**

任何IP都可以访问服务器Websites。

ALLOWED\_HOSTS = ["\*"]

在开发阶段可以，开启DEBUG模式，可以在浏览器上看到Error信息；正式发布产品时，关闭DEBUG模式，为安全考虑。

DEBUG = False

**8 Configuring static files**

a. Make sure that django.contrib.staticfiles is included in your INSTALLED\_APPS in the settings.py file.

b. In your settings.py file, define STATIC\_URL, for example:

STATIC\_URL = '/static/'

c. In your templates, either hardcode the url like /static/my\_app/example.jpg or, preferably, use

the static template tag to build the URL for the given relative path by using the configured

STATICFILES\_STORAGE storage (this makes it much easier when you want to switch to a content delivery network (CDN) for serving static files).

{% load static %}

<img src="{% static "my\_app/example.jpg" %}" alt="My image"/>

d. Store your static files in a folder called static in your app. For example my\_app/static/my\_app/

example.jpg.

**9 Deployment**

django.contrib.staticfiles provides a convenience management command for gathering static files in a single directory so you can serve them easily.

a. Set the STATIC\_ROOT setting to the directory from which you’d like to serve these files, for example:

STATIC\_ROOT = os.path.join(BASE\_DIR,'static')

b. Run the collectstatic management command:

$ mkdir static

$ python manage.py collectstatic

This will copy all files from your static folders into the STATIC\_ROOT directory.

c. Use a web server of your choice to serve the files. Deploying static files covers some common deployment strategies for static files.

**Serving static files in production**

The basic outline of putting static files into production is simple: run the collectstatic command when static files change, then arrange for the collected static files directory (STATIC\_ROOT) to be moved to the static file server and served.

**10 连接数据库**

在setting.py中，设定

DATABASES = {

'default': {

#'ENGINE': 'django.db.backends.sqlite3',

#'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),

'ENGINE': 'django.db.backends.postgresql',

'NAME': 'tmcs',

'USER': 'tmcs',

'PASSWORD': 'tmcs',

'HOST': '127.0.0.1',

'PORT': '5432',}}

# 在Python3 venv中安装uwsgi，开机自启动uwsgi ini 文件

此步虚拟机CentOS7未实现。

建议网站的项目在/home的用户目录下。

**1. 假设，用户名是tmcs，用户主目录/home/tmcs/。然后我们在主目录下创建一个项目目录名tmcs。**

**2. 在项目目录有一个venv环境，且安装uwsgi包。**

**3. 在项目目录下，有一个tmcs\_uwsgi.ini文件。**

[tmcs@localhost tmcs]$ pwd

/home/tmcs/tmcs

[tmcs@localhost tmcs]$ cat tmcs\_uwsgi.pini

[uwsgi]

; project dir

chdir = /home/tmcs/tmcs/tmcs/

; sock file path

socket = 127.0.0.1:8080

wsgi-file = tmcs/wsgi.py

processes = 2

threads = 1

stats = 127.0.0.1:8181

uid = tmcs

gid = tmcs

; a master process will respawn your processes when they die.

master = true

enable-threads = true

; reload whenever this config file changes

; %p is the full path of the current config file

touch-reload = %p

pidfile = tmcs\_uwsgi.pid

daemonize = tmcs\_uwsgi.log

http-websockets = true

; websocket

async = 30

ugreen = ''

http-timeout = 300

**4. 在虚拟机CentOS7中路径/etc/systemd/system下有xxxx.uwsgi.service文件。此文件是开机自启动的模板。**

假设，创建这个网站的用户是tmcs，他在/home/tmcs/下创建了一个项目目录tmcs，且在该目录下有一个tmcs\_uwsgi.ini文件，那么你只要把所有xxxx字符串的地方（包括文件名xxxx.uwsgi.service上的）改成tmcs就好。

[root@localhost system]# pwd

/etc/systemd/system

[root@localhost system]# cat xxxx.uwsgi.service

#xxxx is your user name, you replace it with your user name.

[Unit]

Description=uWSGI instance to serve xxxx

After=syslog.target

[Service]

ExecStart=/bin/bash -c 'cd /home/xxxx/xxxx; source venv/bin/activate; uwsgi --ini xxxx\_uwsgi.ini'

User=xxxx

Group=xxxx

RemainAfterExit=yes

[Install]

WantedBy=multi-user.target

**5. 命令启动**

# systemctl enable xxxx.uwsgi.service

# systemctl start xxxx.uwsgi.service

在Python3 venv中安装uwsgi，开机自启动Django的wsgi.py所在的uwsgi ini 文件。Django, uwsgi, pyscopg2 and so on.

# 设定代理服务器Nginx

此步虚拟机CentOS7未实现。

**SeLinux关闭**

为了避免Nginx被阻止导致无妨正常使用，关闭SELinux。

(venv) [root@localhost example]# vim /etc/selinux/config

"/etc/selinux/config" 15L, 565C# This file controls the state of SELinux on the system.

# SELINUX= can take one of these three values:

# enforcing - SELinux security policy is enforced.

# permissive - SELinux prints warnings instead of enforcing.

# disabled - No SELinux policy is loaded.

#SELINUX=enforcing

SELINUX=disabled

# SELINUXTYPE= can take one of three two values:

# targeted - Targeted processes are protected,

# minimum - Modification of targeted policy. Only selected processes are prootected.

# mls - Multi Level Security protection.

SELINUXTYPE=targeted

(venv) [root@localhost example]# shutdown -r now

(venv) [root@localhost example]# sestatus

SELinux status: disabled

**防火墙配置**

sudo firewall-cmd --permanent --zone=public --add-service=http

sudo firewall-cmd --permanent --zone=public --add-service=https

sudo firewall-cmd --reload

Firewall启动和关闭

systemctl start firewalld

systemctl stop firewalld

**1. CentOS7安装Nginx三部曲**

Nginx就不多介绍了，下面是三部曲具体的操作。

1.安装EPTL源

sudo yum install epel-release

2.安装NGINX

sudo yum install nginx

3.配置NGINX

系统自动启动：

sudo systemctl enable nginx.service

立刻启动服务

sudo systemctl start nginx.service

**2. 配置服务器：**

在/etc/nginx/nginx.conf中:

a. 修改user nginx为当前系统用户(解决**Permission denied问题**)，如：user root

b. 添加一个服务器;

# aemslite web

server {

listen 80 default\_server;

listen [::]:80 default\_server;

server\_name \_;

root /usr/share/nginx/html;

# Load configuration files for the default server block.

include /etc/nginx/default.d/\*.conf;

client\_max\_body\_size 100m;

location / {

include uwsgi\_params;

uwsgi\_pass 127.0.0.1:8080;

}

location /static {

#match with STATIC\_ROOT = os.path.join(BASE\_DIR,'static') in Django

alias /home/tmcs/tmcs/tmcs/static;}

error\_page 404 /404.html;

location = /40x.html {}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {}}

c. 重启nginx

# systemctl restart nginx

**3. Test:**

浏览器-> http://服务器IP

🡪 Centos7.0 的个人总结以及实现步骤的方法：({这里是建立在已经安装了centos7.0 postgres python3的前提下面})

1,先在home目录上面去建立一个放置项目文件的文件夹，（例如项目名字叫aemslite） 🡪mkdir aemslite 并进去(cd aemslite)

2,创建虚拟环境command:/usr/python3.64/bin/python3.6 –m venv {{虚拟环境的名字venv}}🡪source venv/bin/activate

3,接下来安装环境需要的包,方法①pip install Django

②🡪迁移项目的是,在之前的环境项目环境下面下 pip freeze > requirements.txt; --然后拷贝这份文件到你建立的环境下面

🡪pip install –r requirements.txt

4,开始建立一个项目app，cd /home/aemslite; source venv/bin/activate ,Django-admin startproject AEMSLite

5,使用你最熟悉的远程download项目文件下来, 然后再配置相应的设置,这里是以我做过的项目为例子

🡪(1)DEBUG = *True 改成 False*

🡪(2)ALLOWED\_HOSTS = [] 里面添加 “\*”

🡪(3) INSTALLED\_APPS=[]里面添加你创建的app目录{例如 app.login,}

🡪(4) 下面红色背景的是需要更改的。

TEMPLATES = [  
 {  
 'BACKEND': 'django.template.backends.django.DjangoTemplates',  
 'DIRS': [os.path.join(BASE\_DIR,'templates')],  
 'APP\_DIRS': *True*,  
 'OPTIONS': {  
 'context\_processors': [  
 'django.template.context\_processors.debug',  
 'django.template.context\_processors.request',  
 'django.contrib.auth.context\_processors.auth',  
 'django.contrib.messages.context\_processors.messages',  
 ],  
 'builtins':["django.templatetags.static"],  
 },  
 },  
]

🡪数据库的连接部分

DATABASES = {  
 'default': {  
 # 'ENGINE': 'django.db.backends.sqlite3',  
 # 'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),  
 'ENGINE': 'django.db.backends.postgresql',  
 'NAME': 'aemslite',  
 'USER': 'postgres',  
 'PASSWORD': '1234qwer!@#$QWER',  
 'HOST': '127.0.0.1',  
 'PORT': '5432',  
 }  
}

🡪 语言，时区设置

LANGUAGE\_CODE = 'en-us'  
TIME\_ZONE = 'Asia/Shanghai'  
USE\_I18N = *True*USE\_L10N = *True*USE\_TZ = *False*

🡪静态文件的路径设置

STATIC\_URL = '/static/'  
STATIC\_ROOT = os.path.join(BASE\_DIR,'static')  
TEMPLATE\_DIRS = (os.path.join(BASE\_DIR,'templates'),)

🡪收集静态文件的python manage.py collectstatic（可以不做）

🡪测试django环境是否可以用python manage.py runserver

6, 在Python3 venv中安装uwsgi，开机自启动uwsgi ini 文件

🡪 (1)pip install uwsgi(在虚拟环境的项目文件下)**建立一个aemslite\_uwsgi.ini文件。**

[uwsgi]

; project dir

chdir = /home/AEMSLite/ AEMSLite / #写项目的uwsgi.py文件的路径

; sock file path

socket = 127.0.0.1:8080 #8080需要跟nginx的端口一致

wsgi-file = AEMSLite/wsgi.py

processes = 2

threads = 1

stats = 127.0.0.1:8181

uid = root

gid = root

; a master process will respawn your processes when they die.

master = true

enable-threads = true

; reload whenever this config file changes

; %p is the full path of the current config file

touch-reload = %p

pidfile = aemslite\_uwsgi.pid

daemonize = aemslite\_uwsgi.log

http-websockets = true

; websocket

async = 30

ugreen = ''

http-timeout = 300

以下为命令启动uwsgi

uwsgi --http :9000 --chdir /opt/mysite/ --wsgi-file mysite/wsgi.py --master --processes 4 --threads 2 --stats 127.0.0.1:9090🡪可以不做

**(2)在虚拟机CentOS7中路径/etc/systemd/system下有xxxx.uwsgi.service文件。此文件是开机自启动的模板。**

假设，创建这个网站的用户是tmcs，他在/home/tmcs/下创建了一个项目目录tmcs，且在该目录下有一个tmcs\_uwsgi.ini文件，那么你只要把所有xxxx字符串的地方（包括文件名xxxx.uwsgi.service上的）改成tmcs就好。

[root@localhost system]# pwd

/etc/systemd/system

[root@localhost system]# cat xxxx.uwsgi.service

#xxxx is your user name, you replace it with your user name.

[Unit]

Description=uWSGI instance to serve xxxx

After=syslog.target

[Service]

ExecStart=/bin/bash -c 'cd /home/xxxx/xxxx; source venv/bin/activate; uwsgi --ini xxxx\_uwsgi.ini'

User=xxxx

Group=xxxx

RemainAfterExit=yes

[Install]

WantedBy=multi-user.target

**命令启动**

# systemctl enable xxxx.uwsgi.service

# systemctl start xxxx.uwsgi.service

7,防火墙的设置{主要是开端口和数据库连接的端口 90和5432 端口的}

firewall-cmd --zone=public --add-port=90/tcp --permanent

firewall-cmd --zone=public --add-port=5432/tcp --permanent

启动设置

systemctl start firewalld.service

systemctl enable firewalld.service

查看状态

firewall-cmd --state或者systemctl status firewalld.service

启动设置

systemctl start firewalld.service

systemctl enable firewalld.service --开机时启动firewall

停止设置

stop firewalld.service --停止firewall

systemctl disable firewalld.service --禁止firewall开机启动

查看已启动的服务列表

systemctl list-unit-files|grep enabled

开启端口：

firewall-cmd --zone=public --add-port=80/tcp --permanent

命令含义：

–zone #作用域

–add-port=80/tcp #添加端口，格式为：端口/通讯协议

–permanent #永久生效，没有此参数重启后失效

查看已经开放的端口

firewall-cmd --list-port

修改规则

yum -y install iptables-services

vi /etc/sysconfig/iptables

增加规则

-A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT

8,nginx文件的设置和配置

Nginx就不多介绍了，下面是三部曲具体的操作。

1.安装EPTL源

sudo yum install epel-release

2.安装NGINX

sudo yum install nginx

3.配置NGINX

系统自动启动：

sudo systemctl enable nginx.service

立刻启动服务

sudo systemctl start nginx.service

**2. 配置服务器：**

在/etc/nginx/nginx.conf中:

a. 修改user nginx为当前系统用户(解决**Permission denied问题**)，如：user root

b. 添加一个服务器;

# aemslite web

server {

listen 90 default\_server; #改成开放的端口

listen [::]:90 default\_server; #改成开放的端口

server\_name \_; # 这里需要注意一下，监听为空就行了，不需要添加值

root /usr/share/nginx/html;

# Load configuration files for the default server block.

include /etc/nginx/default.d/\*.conf;

client\_max\_body\_size 100m;

location / {

include uwsgi\_params;

uwsgi\_pass 127.0.0.1:8080; #8080需要跟uwsgi的端口一致

}

location /static {

#match with STATIC\_ROOT = os.path.join(BASE\_DIR,'static') in Django

alias /home/tmcs/tmcs/tmcs/static;} #静态文件的具体路径

error\_page 404 /404.html;

location = /40x.html {}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {}}

c. 重启nginx

# systemctl restart nginx

D, 重新再启动uwsgi

# systemctl enable aemslite.uwsgi.service

# systemctl start aemslite.uwsgi.service

# systemctl status aemslite.uwsgi.service{查看uwsgi运行状态}

ps -ef|grep uwsgi 检查uwsgi的端口和IP

systemctl status nginx.service 检查nginx运行状态

systemctl restart nginx.service 重新运行nginx

\*\*\*\*\*\*\*\*如果发现错误的话\*\*\*\*\*\*\*\*\*\*\*

🡪 杀死所有进程再重新启动uwsgi.Service

**3. Test:**

浏览器-> http://服务器IP