□Engineering Directive(ED) ■Engineering Specification (ES) □Manufacturing System (MS)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Subject: Hugin Installation Guide  Document classification: Confidential | | | Doc. No.: | | | Rev.:V001 | |
| Effective Date: 2019/04/09 | | | Revision Status | |
| Description:  Hugin Installation Guide | | | | | | Page | Rev. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Reason for Change: | | | | | | Reviewed by/ |  |
| ----------------- | ----------------- |
| Copy to(All contents): | | | | | | | | ----------------- | ----------------- |
|  | | | | | | | |
| Approved by/ | Reviewed by/ |  | |  |  | Prepared by/  Sam Kwok | |
| ----------------- | ----------------- | | ----------------- | ----------------- |
| ----------------- | ----------------- | | ----------------- | ---------------- |

Notice: This cover sheet forms a part of the following directive (specification) and is not to be discarded unless superseded by a revised issue

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Comment** |
| V001 | 2019/04/09 | Sam Kwok | Initial Version |
| V002 | 2019/07/19 | Sam Kwok | 重新规划章节和内容清晰细化 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目录**

[1 预设环境 4](#_Toc14440053)

[1.1 硬件信息 4](#_Toc14440054)

[1.2 安装CentOS 4](#_Toc14440055)

[1.3 开机自动网络连接 4](#_Toc14440056)

[1.4 手动编译安装Python 3.64 5](#_Toc14440057)

[1.4.1 安装epel release 5](#_Toc14440058)

[1.4.2 安装Development Tools 5](#_Toc14440059)

[1.4.3 编译Python3.64依赖库 5](#_Toc14440060)

[1.4.4 编译安装Python 3.64 6](#_Toc14440061)

[1.5 安装PostgreSQL 6](#_Toc14440062)

[1.5.1 安装PostgreSQL仓库 6](#_Toc14440063)

[1.5.2 安装PostgreSQL客户端程序包 6](#_Toc14440064)

[1.5.3 安装PostgreSQL服务器端程序包 7](#_Toc14440065)

[1.5.4 初始化数据库和开机自启动 7](#_Toc14440066)

[1.5.5 设置PostgreSQL超级用户密码 7](#_Toc14440067)

[1.5.6 连接服务器数据库配置 8](#_Toc14440068)

[1.6 安装代理服务器Nginx 9](#_Toc14440069)

[1.6.1 安装EPTL源 9](#_Toc14440070)

[1.6.2 安装NGINX 9](#_Toc14440071)

[1.6.3 自动启动NGINX 9](#_Toc14440072)

[1.7 配置防火墙 10](#_Toc14440073)

[1.8 关闭SELinux 10](#_Toc14440074)

[2 搭建系统 11](#_Toc14440075)

[2.1 创建操作系统用户 11](#_Toc14440076)

[2.2 创建Python3 venv 11](#_Toc14440077)

[2.2.1 在用户主目录创建项目目录 11](#_Toc14440078)

[2.2.2 创建Python3 venv 11](#_Toc14440079)

[2.3 布置发布的Web项目 11](#_Toc14440080)

[2.3.1 复制发布的Web项目 11](#_Toc14440081)

[2.3.2 安装Web项目依赖包 12](#_Toc14440082)

[2.3.3 创建数据库 12](#_Toc14440083)

[2.3.3.1 查看Web项目系统的数据库连接配置 12](#_Toc14440084)

[2.3.3.2 创建数据库 13](#_Toc14440085)

[2.3.4 数据表迁移 13](#_Toc14440086)

[2.3.5 新建Web项目系统admin账号 13](#_Toc14440087)

[2.3.6 设定Nginx服务器 14](#_Toc14440088)

[2.3.6.1 使用root用户权限 14](#_Toc14440089)

[2.3.6.2 添加服务器 14](#_Toc14440090)

[2.3.7 设定uwsgi服务器 15](#_Toc14440091)

[2.3.7.1 新建uwsgi ini文件 15](#_Toc14440092)

[2.3.7.2 新增uwsgi服务 16](#_Toc14440093)

## 预设环境

目前已经有一个硬盘空间有60G的KVM虚拟机CentOS7文件。

操作系统CentOS7超级用户信息：

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

### 硬件信息

具体需求搭不同配置。

### 安装CentOS

有无桌面没关系。可以安装系统服务器数据中心系统，或者最小配置系统的服务器。

CentOS的版本是7.5。

查看操作系统命令：

[root@AOI ~]# cat /etc/redhat-release

CentOS Linux release 7.5.1804 (Core)

基本的分区有/, /var, /home, /boot和swap。

分区需要注意：系统数据库会安装在/var，所以需要数据库的空间。

### 开机自动网络连接

1. 查看网卡信息

[root@AOI network-scripts]# ifconfig

ens1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 10.41.95.122 netmask 255.255.255.0 broadcast 10.41.95.255

inet6 fe80::ca49:8ed4:f40d:487b prefixlen 64 scopeid 0x20<link>

ether 24:8a:07:88:56:72 txqueuelen 1000 (Ethernet)

RX packets 2452320 bytes 520316770 (496.2 MiB)

RX errors 10 dropped 0 overruns 0 frame 10

TX packets 32867 bytes 2846636 (2.7 MiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

知道了网卡的名字是ens1。

1. 修改网卡的配置ONBOOT的值。

[root@AOI ~]# cd /etc/sysconfig/network-scripts/

把ifcfg-ens1文件中的ONBOOT的值改为yes

ONBOOT=yes

1. 重开机。

### 手动编译安装Python 3.64

目前使用的Python3版本源码包是Python-3.6.4.tgz。

下载路径：<https://www.python.org/downloads/release/python-364/>

#### 安装epel release

安装epel release，可以获取到比较新的软件包。

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

#### 安装Development Tools

安装开发环境的相关工具，可以提前安装编译Python3需要的工具，比如gcc。

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

如果在安装Development Tools出现以下失败信息：

Maybe run yum groups mark install

No packages in any requested group availabe to install or update

安装失败解决步骤如下：

1. 先执行以下两条命令：

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

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

1. 再安装Development Tools

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

#### 编译Python3.64依赖库

必须的：

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

#### 编译安装Python 3.64

使用root用户进行这一步的操作。

1. 解压Python3压缩包

[root@localhost Python]# tar -xvf Python-3.6.4.tgz

1. 进入到解压目录

[root@localhost Python]# cd Python-3.6.4/

1. 创建Python3安装路径

[root@localhost Python-3.6.4]# mkdir /usr/python3.64

1. 执行配置

[root@localhost Python-3.6.4]# ./configure --prefix=/usr/python3.64 --enable-optimizations

1. 编译

[root@localhost Python-3.6.4]# make

1. 安装

[root@localhost Python-3.6.4]# make install

### 安装PostgreSQL

我们该项目选择使用PostgreSQL 10。

#### 安装PostgreSQL仓库

[root@localhost Python-3.6.4]# yum install https://download.postgresql.org/pub/repos/yum/10/redhat/rhel-7-x86\_64/pgdg-centos10-10-2.noarch.rpm

我执行不了这句话，用的下面这句

yum install <https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat->repo-latest.noarch.rpm

#### 安装PostgreSQL客户端程序包

[root@localhost Python-3.6.4]# yum install postgresql10 -y

#### 安装PostgreSQL服务器端程序包

[root@localhost Python-3.6.4]# yum install postgresql10-server -y

#### 初始化数据库和开机自启动

[root@localhost Python-3.6.4]# /usr/pgsql-10/bin/postgresql-10-setup initdb

[root@localhost Python-3.6.4]# systemctl enable postgresql-10

[root@localhost Python-3.6.4]# systemctl start postgresql-10

我在这里遇到了端口被占用的问题，一直报错，就是下面这个

systemctl start postgresql-10

Job for postgresql-10.service failed because the control process exited with error code. See "systemctl status postgresql-10.service" and "journalctl -xe" for details.

接着我就去 /var/lib/pgsql/10/data/postgresql.conf这个文件里面，把post改成别的，

此时systemctl start postgresql-10就可以用了

#### 设置PostgreSQL超级用户密码

Web项目会使用PostgreSQL的超级用户postgres连接数据库，且密码为1234qwer!@#$QWER，端口为5432。

PostgreSQL超级用户是postgres，不过是没有密码的，需要给它设置密码。

1. 设置密码的操作：

[root@honor\_10-6abcf333af37da1d ~]# su – postgres

-bash-4.2$ psql

psql (9.2.23, server 10.9)

WARNING: psql version 9.2, server version 10.0.

Some psql features might not work.

Type "help" for help.

postgres=# alter user postgres with password ‘1234qwer!@#$QWER’;

ALTER ROLE

postgres=# \q

-bash-4.2$ exit

logout

这里我执行psql的时候出现了这么一个问题

psql: could not connect to server: No such file or directory

        Is the server running locally and accepting

        connections on Unix domain socket "/var/run/postgresql/.s.PGSQL.5432"?

究其原因也是端口被占用了，这时候就得改一下

像这么写 psql -p 5435（这是我在postgresql.conf里面写的端口号）就可以进入数据库了

1. 配置本地需要密码

需要把/var/lib/pgsql/10/data/pg\_hba.conf中，下面的内容：

# TYPE DATABASE USER ADDRESS METHOD

# "local" is for Unix domain socket connections only

local all all peer

改成：

# "local" is for Unix domain socket connections only

local all all md5

"local" 的设置在PostgreSQL安装的服务器上使用psql命令时有效。

1. 修改配置生效

[root@localhost Python-3.6.4]# systemctl restart postgresql-10

#### 连接服务器数据库配置

* 如果连接服务器数据库的项目系统，跟数据库服务器在同一个操作系统，做一下配置：

1. 配置pg\_hba.conf

把/var/lib/pgsql/10/data/pg\_hba.conf中下面的内容：

# TYPE DATABASE USER ADDRESS METHOD

# IPv4 local connections:

host all all 127.0.0.1/32 ident

改成：

# IPv4 local connections:

host all all 127.0.0.1/32 md5

"host" 的设置使用IP去连接服务器上安装的PostgreSQL数据库时有效。

1. 配置postgresql.conf

把/var/lib/pgsql/10/data/postgresql.conf中，listen\_addresses和port设置如下：

listen\_addresses = ‘localhost’

port = 5432

1. 修改配置生效

[root@localhost Python-3.6.4]# systemctl restart postgresql-10

* 如果连接服务器数据库的项目系统，跟数据库服务器不在同一个操作系统，做一下配置：

1. 配置pg\_hba.conf

# TYPE DATABASE USER ADDRESS METHOD

host all all all md5

ADDRESS可以是项目系统服务器的IP。例如：192.168.1.1/32。

1. 配置postgresql.conf

把/var/lib/pgsql/10/data/postgresql.conf中，listen\_addresses和port设置如下：

listen\_addresses = ‘\*’

port = 5432

1. 修改配置生效

[root@localhost Python-3.6.4]# systemctl restart postgresql-10

1. 设置防火墙允许项目系统服务器IP访问该数据库服务器

[root@zhs-w90003690 data]# firewall-cmd --permanent --add-rich-rule="rule family="ipv4" source address="项目系统服务器IP" port protocol="tcp" port="5432" accept"

[root@zhs-w90003690 data]# systemctl restart firewalld.service

### 安装代理服务器Nginx

#### 安装EPTL源

[root@localhost Python-3.6.4]# yum install epel-release

#### 安装NGINX

[root@localhost Python-3.6.4]# yum install nginx

#### 自动启动NGINX

[root@localhost Python-3.6.4]# systemctl enable nginx.service

[root@localhost Python-3.6.4]# systemctl start nginx.service

### 配置防火墙

因为是web服务器，添加service http和https。

[root@localhost example]# firewall-cmd --permanent --zone=public --add-service=http

[root@localhost example]# firewall-cmd --permanent --zone=public --add-service=https

[root@localhost example]# firewall-cmd --reload

Firewall启动和关闭

systemctl start firewalld

systemctl stop firewalld

### 关闭SELinux

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

设置如下：

"/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

## 搭建系统

预设环境部分准备好了，就可以进行系统搭建了。

### 创建操作系统用户

需要给Web项目新增一个专有的操作系统用户及密码。

用户：hugin

密码：1234qwer!@#$QWER

需要在root用户下才能新增成功，命令如下：

[root@localhost example]# useradd hugin

[root@localhost example]# passwd hugin

### 创建Python3 venv

在新增的操作系统用户hugin下，创建Python3 venv。

#### 在用户主目录创建项目目录

使用用户hugin在用户主目录/home/hugin下创建项目目录hugin。命令如下：

[hugin@localhost ~]$ mkdir hugin

#### 创建Python3 venv

在项目目录hugin下，创建Python3 venv。

[hugin@localhost ~]$ cd hugin

[hugin@localhost hugin]$ /usr/python3.64/bin/apython3.6 -m venv venv

source venv/bin/activate 进入虚拟环境

### 布置Web Service

#### 复制Web Service的发布版

首先，把发布的Web Service整个目录hugin\_webservice拷到项目目录/home/hugin/hugin下。在项目目录/home/hugin/hugin下，查看hugin\_webservice目录的文件目录结构如下所示：

.

└── hugin\_webservice

   ├── hugin\_webservice

   │   ├── database

   │   ├── db.sqlite3

   │   ├── hugin\_webservice

   │   ├── manage.py

   │   ├── static

   │   └── webservice

   ├── hugin\_webservice\_nginx.conf

   ├── hugin\_webservice\_uwsgi.ini

   ├── hugin\_webservice.uwsgi.service

└── requirements.txt

然后，为了确保所属用户和用户群组是hugin。执行以下命令：

[root@localhost ~]# cd /home/hugin/hugin

[root@localhost ~]# chown –R hugin:hugin hugin

#### 安装依赖包

Web Service发布时，会附带requirements.txt，该文件内容是Web Service能正常运行需要安装的依赖包列表。

首先，确保所属用户和用户群组是hugin。命令如下：

[root@localhost ~]# cd /home/hugin/hugin/hugin\_webservice

[root@localhost hugin\_webservice]# chown –R hugin:hugin requirements.txt

然后，必须在用户hugin下，进入到项目hugin的Python3 venv环境中，下载requirements.txt中列出的依赖包。命令如下：

[root@localhost ~]# su - hugin

[hugin@localhost ~]$ cd /home/hugin/hugin

[hugin@localhost hugin]$ source venv/bin/activate

(venv) [hugin@localhost hugin]$ cd /home/hugin/hugin/hugin\_webservice

(venv) [hugin@localhost hugin\_webservice]$ requirements=`cat requirements.txt | tr "\n" " "`

(venv) [hugin@localhost hugin\_webservice]$ pip install $requirements

(venv) [hugin@localhost hugin\_webservice]$ deactivate

#### 创建数据库

##### 查看Web项目系统的数据库连接配置

在目录/home/hugin/hugin/hugin\_webservice/hugin\_webservice/hugin\_webservice下的settings.py文件中，数据库连接配置，如下：

DATABASES = {

'default': {

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

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

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

'NAME': 'hugin',

'USER': 'postgres',

'PASSWORD': '1234qwer!@#$QWER',

'HOST': '127.0.0.1',

'PORT': '5432',

}

}

数据库的名字是hugin。

##### 创建数据库

已经知道数据库的名字是hugin，需要创建数据库。命令如下：

[root@honor\_10-6abcf333af37da1d ~]# su – postgres

-bash-4.2$ createdb hugin

创建数据库的时候，需要输入用户postgres的密码。

#### 数据表迁移

必须在用户hugin下，进入到项目hugin的Python3 venv环境中，操作数据表数据结构迁移。命令如下：

[hugin@localhost ~]$ cd /home/hugin/hugin

[hugin@localhost hugin]$ source venv/bin/activate

(venv) [hugin@localhost hugin]$ cd hugin\_webservice/hugin\_webservice

(venv) [hugin@localhost hugin\_webservice]$ python manage.py makemigrations

(venv) [hugin@localhost hugin\_webservice]$ python manage.py migrate

(venv) [hugin@localhost hugin\_webservice]$ deactivate

#### 新建Web项目系统admin账号

要新建hugin项目系统管理数据表的admin账号为：hugin，密码：hugin。

必须在用户hugin下，进入到项目hugin的Python3 venv环境中，新建数据表管理账号。命令如下：

[hugin@localhost ~]$ cd /home/hugin/hugin

[hugin@localhost hugin]$ source venv/bin/activate

(venv) [hugin@localhost hugin]$ cd hugin\_webservice/hugin\_webservice

(venv) [hugin@localhost hugin\_webservice]$ python manage.py createsuperuser

(venv) [hugin@localhost hugin\_webservice]$ deactivate

#### 设定Nginx服务器

配置文件/etc/nginx/nginx.conf。

##### 使用root用户权限

修改文件nginx.conf：

把user nginx; 改为 user root;。

否则会有**Permission denied问题。**

##### 添加服务器

* 首先，添加server。

请注意了，确认好要nginx服务器的端口。命令如下：

[root@localhost ~]# netstat -lnt

使用端口81，如果nginx.conf中已经预设了81端口的nginx服务器，需要先删除掉这个服务器。

然后在nginx.conf的http {}内的尾部追加hugin的服务器，内容如下：

server {

listen 81 default\_server;

listen [::]:81 default\_server;

server\_name \_;

root /usr/share/nginx/html;

# Load configuration files for the default server block.

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

location / {

include uwsgi\_params;

uwsgi\_pass 127.0.0.1:8100;

}

location /static {

alias /home/hugin/hugin/hugin\_webservice/hugin\_webservice/static;

}

error\_page 404 /404.html;

location = /40x.html {

}

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

location = /50x.html {

}

}

说明：由于用户nginx服务器的端口设为81，所以映射uwsgi的端口8100为nginx服务器的端口接上00；同时会使用8101端口为nginx服务器的端口接上01，用于uwsgi的状态服务。

* 防火墙开放端口

对外使用的端口是81，需要防火墙开放这个端口。命令如下：

[root@localhost ~]# firewall-cmd --permanent --zone=public --add-port=81/tcp

[root@localhost ~]# systemctl --reload

* 然后，重启Nginx服务。

[root@localhost ~]# systemctl restart nginx

#### 设定uwsgi服务器

##### 新建uwsgi ini文件

首先，在/home/hugin/hugin目录中新增hugin\_uwsgi.ini文件。内容如下：

[uwsgi]

; project dir

chdir = /home/hugin/hugin/hugin/

; sock file path

socket = 127.0.0.1:8100

wsgi-file = hugin/wsgi.py

processes = 2

threads = 1

stats = 127.0.0.1:8101

; 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 = hugin\_uwsgi.pid

daemonize = hugin\_uwsgi.log

uid = hugin

gid = hugin

说明：

服务器uwsgi需要两个端口8100,8101。这个两个端口的格式是服务器nginx的端口81加上00，01。由此格式的取法，服务器uwsgi第一个端口是8100，第二个端口8101。

注意，确保所属用户和用户群组是hugin。命令如下：

[root@localhost ~]# cd /home/hugin/hugin

[root@localhost hugin]# chown –R hugin:hugin hugin\_uwsgi.ini

##### 新增uwsgi服务

* 首先，在/etc/systemd/system目录中新增hugin.uwsgi.service文件。内容如下：

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

[Unit]

Description=uWSGI instance to serve hugin

After=syslog.target

[Service]

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

User=hugin

Group=hugin

RemainAfterExit=yes

[Install]

WantedBy=multi-user.target

* 然后，启动uwsgi服务。

[root@localhost ~]# systemctl start hugin.uwsgi.service

[root@localhost ~]# systemctl enable hugin.uwsgi.service

### 布置发布的Web项目

#### 复制发布的Web项目

发布的Djang Web项目目录是hugin，该目录里会有manage.py。把整个目录拷到项目目录/home/hugin/hugin下。

为了确保所属用户和用户群组是hugin。执行以下命令：

[root@localhost ~]# cd /home/hugin/hugin

[root@localhost ~]# chown –R hugin:hugin hugin

#### 安装Web项目依赖包

Web项目发布时，会附带requirements.txt，该文件内容是Web项目可以正常需要安装的依赖包列表。

首先，把requirements.txt拷贝到跟目录venv同一目录中。确保所属用户和用户群组是hugin。命令如下：

[root@localhost ~]# cd /home/hugin/hugin

[root@localhost hugin]# chown –R hugin:hugin requirements.txt

必须在用户hugin下，进入到项目hugin的Python3 venv环境中，下载requirements.txt中列出的依赖包。命令如下：

[root@localhost ~]# su - hugin

[hugin@localhost ~]$ cd /home/hugin/hugin

[hugin@localhost hugin]$ source venv/bin/activate

(venv) [hugin@localhost hugin]$ requirements=`cat requirements.txt | tr "\n" " "`

(venv) [hugin@localhost hugin]$ pip install $requirements

(venv) [hugin@localhost hugin]$ deactivate

#### 创建数据库

##### 查看Web项目系统的数据库连接配置

在目录/home/hugin/hugin/hugin下的settings.py文件中，数据库连接配置，如下：

DATABASES = {

'default': {

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

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

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

'NAME': 'hugin',

'USER': 'postgres',

'PASSWORD': '1234qwer!@#$QWER',

'HOST': '127.0.0.1',

'PORT': '5432',

}

}

数据库的名字是hugin。

##### 创建数据库

已经知道数据库的名字是hugin，需要创建数据库。命令如下：

[root@honor\_10-6abcf333af37da1d ~]# su – postgres

-bash-4.2$ createdb hugin

创建数据库的时候，需要输入用户postgres的密码。

#### 数据表迁移

必须在用户hugin下，进入到项目hugin的Python3 venv环境中，操作数据表数据结构迁移。命令如下：

[hugin@localhost ~]$ cd /home/hugin/hugin

[hugin@localhost hugin]$ source venv/bin/activate

(venv) [hugin@localhost hugin]$ cd hugin

(venv) [hugin@localhost hugin]$ python manage.py makemigrations

(venv) [hugin@localhost hugin]$ python manage.py migrate

(venv) [hugin@localhost hugin]$ deactivate

#### 新建Web项目系统admin账号

要新建Web项目系统admin账号为：hugin，密码：hugin。

必须在用户hugin下，进入到项目hugin的Python3 venv环境中，新建数据表管理账号。命令如下：

[hugin@localhost ~]$ cd /home/hugin/hugin

[hugin@localhost hugin]$ source venv/bin/activate

(venv) [hugin@localhost hugin]$ cd hugin

(venv) [hugin@localhost hugin]$ python manage.py createsuperuser

(venv) [hugin@localhost hugin]$ deactivate

#### 设定Nginx服务器

配置文件/etc/nginx/nginx.conf。

##### 使用root用户权限

修改文件nginx.conf：

把user nginx; 改为 user root;。

否则会有**Permission denied问题。**

##### 添加服务器

* 首先，添加server。

请注意了，确认好要nginx服务器的端口。命令如下：

[root@localhost ~]# netstat -lnt

如果要使用端口80，因为nginx.conf中已经预设了80端口的nginx服务器，需要先删除掉这个服务器。

假设端口80没有被使用，可以使用80端口。

然后在nginx.conf的http {}内的尾部追加hugin的服务器，内容如下：

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 10240m;

location / {

include uwsgi\_params;

uwsgi\_pass 127.0.0.1:8000;

}

location /static {

alias /home/hugin/hugin/hugin/static;

}

error\_page 404 /404.html;

location = /40x.html {

}

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

location = /50x.html {

}

}

说明：由于用户nginx服务器的端口设为80，所以映射uwsgi的端口8000为nginx服务器的端口接上00；同时会使用8001端口为nginx服务器的端口接上01，用于uwsgi的状态服务。

* 防火墙开放端口

对外使用的端口是80，需要防火墙开放这个端口。命令如下：

[root@localhost ~]# firewall-cmd --permanent --zone=public --add-port=80/tcp

[root@localhost ~]# systemctl --reload

* 然后，重启Nginx服务。

[root@localhost ~]# systemctl restart nginx

#### 设定uwsgi服务器

##### 新建uwsgi ini文件

首先，在/home/hugin/hugin目录中新增hugin\_uwsgi.ini文件。内容如下：

[uwsgi]

; project dir

chdir = /home/hugin/hugin/hugin/

; sock file path

socket = 127.0.0.1:8000

wsgi-file = hugin/wsgi.py

processes = 2

threads = 1

stats = 127.0.0.1:8001

; 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 = hugin\_uwsgi.pid

daemonize = hugin\_uwsgi.log

uid = hugin

gid = hugin

说明：

服务器uwsgi需要两个端口8000,8001。这个两个端口的格式是服务器nginx的端口80加上00，01。由此格式的取法，服务器uwsgi第一个端口是8000，第二个端口8001。

注意，确保所属用户和用户群组是hugin。命令如下：

[root@localhost ~]# cd /home/hugin/hugin

[root@localhost hugin]# chown –R hugin:hugin hugin\_uwsgi.ini

##### 新增uwsgi服务

* 首先，在/etc/systemd/system目录中新增hugin.uwsgi.service文件。内容如下：

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

[Unit]

Description=uWSGI instance to serve hugin

After=syslog.target

[Service]

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

User=hugin

Group=hugin

RemainAfterExit=yes

[Install]

WantedBy=multi-user.target

* 然后，启动uwsgi服务。

[root@localhost ~]# systemctl start hugin.uwsgi.service

[root@localhost ~]# systemctl enable hugin.uwsgi.service