系统：centos7.X

1．首先安装python3

centos7原本就安装了Python2，而且这个Python2不能被删除，因为有很多系统命令，比如yum都要用到。

安装步骤：

yum -y install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel gdbm-devel db4-devel libpcap-devel xz-devel

wget <https://www.python.org/ftp/python/3.6.5/Python-3.6.5.tar.xz>

mkdir /usr/local/python3

tar -xvf Python-3.6.5.tar.xz

cd Python-3.6.5

./configure --prefix=/usr/local/python3

make

make install

创建软链接

ln -s /usr/local/python3/bin/python3 /usr/bin/python3

ln -s /usr/local/python3/bin/pip3 /usr/bin/pip3

在命令行中输入python3测试

2.安装mysql， 这里安装的是mariadb，其实就是mysql另外一个版本，使用时是相同的

1. 安装

sudo yum install mariadb-server

2. 启动， 重启

sudo systemctl start mariadb

sudo systemctl restart mariadb

3. 设置bind-ip

vim /etc/my.cnf

在 [mysqld]:

下面加一行

bind-address = 0.0.0.0

4. 设置外部ip可以访问

先进入mysql才能运行下面命令:

mysql 直接进入就行

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY 'Ys6688' WITH GRANT OPTION;

FLUSH PRIVILEGES;

5. 安装mysqlclient出问题

centos 7：

yum install python-devel mariadb-devel -y

ubuntu：

sudo apt-get install libmysqlclient-dev

然后：pip install mysqlclient

3.安装nginx

yum install epel-release

yum install nginx

systemctl start nginx

nginx配置文件，在nginx/conf目录下新建xx.conf文件

# the upstream component nginx needs to connect to

upstream django {

# server unix:///path/to/your/mysite/mysite.sock; # for a file socket

server 127.0.0.1:8000; # for a web port socket (we'll use this first)

}

# configuration of the server

server {

# the port your site will be served on

listen 80;

# the domain name it will serve for

server\_name 你的ip地址 ; # substitute your machine's IP address or FQDN

charset utf-8;

# max upload size

client\_max\_body\_size 75M; # adjust to taste

# Django media

location /media {

alias 你的目录/Mxonline/media; # 指向django的media目录

}

location /static {

alias 你的目录/Mxonline/static; # 指向django的static目录

}

# Finally, send all non-media requests to the Django server.

location / {

uwsgi\_pass django;

include uwsgi\_params; # the uwsgi\_params file you installed

}

}

4.安装uwsgi, pip3 install uwsgi

项目下新建uwsgi.ini文件

# mysite\_uwsgi.ini file

[uwsgi]

# Django-related settings

# the base directory (full path)

chdir = /home/xxx/projects/myblog

# Django's wsgi file

module = myblog.wsgi

# the virtualenv (full path)

# process-related settings

# master

master = true

# maximum number of worker processes

processes = 10

# the socket (use the full path to be safe

socket = 127.0.0.1:8000

# ... with appropriate permissions - may be needed

# chmod-socket = 664

# clear environment on exit

vacuum = true

virtualenv = /home/xxx/.virtualenvs/myblog

daemonize = /home/xxx/projects/xxx/log/uwsgi.log

注：

chdir： 表示需要操作的目录，也就是项目的目录

module： wsgi文件的路径

processes： 进程数

virtualenv：虚拟环境的目录

进入python3安装目录bin文件夹下，复制uwsgi到/usr/bin下，即可在项目中使用uwsgi -i uwsgi.ini启动项目。

5. 使用supervisor来管理进程

pip install supervisor

echo\_supervisord\_conf > /etc/supervisord.conf

vim /etc/supervisord.conf

在最后加上需要管理的进程，配置示例

[program:名字]

command=uwsgi –i uwsgi.ini

directory=/项目path

startsecs=0

stopwaitsecs=0

autostart=true

autorestart=true

配置完成后，启动supervisor

supervisord -c /etc/supervisord.conf

supervisorctl -c /etc/supervisord.conf [start|stop|restart] [program-name|all]

股票配置

#the upstream component nginx needs to connect to

upstream django {

# server unix:///path/to/your/mysite/mysite.sock; # for a file socket

server 127.0.0.1:8000; # for a web port socket (we'll use this first)

}

# configuration of the server

map $http\_upgrade $connection\_upgrade {

default upgrade;

'' close;

}

server {

# the port your site will be served on

listen 80;

# the domain name it will serve for

server\_name 39.100.109.254 zt.jinguxun.net; # substitute your machine's IP address or FQDN

charset utf-8;

# max upload size

client\_max\_body\_size 100M; # adjust to taste

client\_body\_temp\_path /root/project/stock-strong/tmp;

# Django media

#location /media {

# alias /media; # 指向django的media目录

#}

location /static {

alias /root/project/stock-strong/static; # 指向django的static目录

}

location /socketLink {

proxy\_pass http://127.0.0.1:8001;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection $connection\_upgrade;

}

# Finally, send all non-media requests to the Django server.

location / {

uwsgi\_pass django;

include uwsgi\_params; # the uwsgi\_params file you installed

# proxy\_pass http://127.0.0.1:8000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

}

}

# mysite\_uwsgi.ini file

[uwsgi]

# Django-related settings

# the base directory (full path)

chdir = /root/project/stock-strong

# Django's wsgi file

module = stock.wsgi

# the virtualenv (full path)

# process-related settings

# master

master = true

# maximum number of worker processes

processes = 10

# the socket (use the full path to be safe

socket = 127.0.0.1:8000

# ... with appropriate permissions - may be needed

# chmod-socket = 664

# clear environment on exit

vacuum = true

#virtualenv = /root/project/stock-strong

daemonize = /root/project/stock-strong/uwsgi.log

[program:uwsgi]

command=uwsgi -i uwsgi.ini

directory=/root/project/stock-strong

startsecs=0

stopwaitsecs=0

autostart=true

autorestart=true

[program:websocket\_server]

command=python3 websocket\_server.py

directory=/root/project/stock-strong

startsecs=0

stopwaitsecs=0

autostart=true

autorestart=true

[program:ztb]

command=python3 ztb.py

directory=/root/project/stock-strong

startsecs=0

stopwaitsecs=0

autostart=true

autorestart=true