# 一、安装/配置

1. **收集所有包**

pip freeze >plist.txt

1. **安装所有包**

pip install -r plist.txt

**3、supervisor多进程管理**

pip install supervisor

**4、生成配置文件**

echo\_supervisor\_conf > supervisor.conf

**5、修改supervisor.conf文件**

**；找到**

[include]

files = /etc/supervisor/\*.ini

**；修改为**

[include]

files = 自己定义的目录/\*.ini

[unix\_http\_server]

;file=/tmp/supervisor.sock #修改为特定目录

[supervisord]

logfile=/tmp/supervisord.log

pidfile=/tmp/supervisord.pid

[supervisorctl]

;serverurl=unix:///tmp/supervisor.sock ; use a unix:// URL for a unix socket

1. **复制文件到etc、或其他目录(自定义目录）**
2. **在自定义目录创建tornado.ini**
3. **修改tornado.ini**

[group:tornadoes]

programs=tornado-8000,....# 自行添加

[program:tornado-8001] #程序：程序名

# ~/.virtualenvs/虚拟目录名/bin/python

command=虚拟环境中的python文件 [项目目录] --port=8000

directory=程序目录

autorestart=true ;自动重启

redirect\_stderr=true ;开启日志

stdout\_logfile=自定义目录/tornodo.log ;日志位置

loglevel=info ;错误等级

# 二 、运行

## 命令

supervisord -c 自定义目录/supervisord.conf

## 查看进程

ps aux | grep supervisord

## 3、进入终端

supervisorctl -c 自定义目录/supervisord.conf

----------命令----------------------------

1. status ：查看进程运行状态

* 正确启动可以看到程序条目
* 状态为空，检查配置文件是否有；或没开启任务

1. stop tornadoes:\* ：停止所有
2. retstart tornadoes:\* ：重启
3. update ：不停止更新
4. exit：退出

## 关闭supervisord

-c 指定配置文件

supervisorctl -c supervisord.conf shutdown

## 错误信息

**entered FATAL state, too many start retries too quickly**

**查看错误日志，有可能是程序有错误**

# nginx配置

## 配置静态文件路径

server:

location /statice {

root /var/www/a # 静态文件

if ( $query\_string ){

expires max; # 缓存最大

}

}

## 配置tornado

// 配置多台应用服务器

**// 在server外**

**http:**

upstream tornadoes {

server 127.0.0.1:8000

server 127.0.0.1:8001

server 127.0.0.1:8002

}

// 配置server，只配置一次

**server:**

location / {

# 传递请求的server header

proxy\_pass\_header Server

proxy\_redirect off # 重定向关

proxy\_set\_header Host $http\_host

# 传送用户真实IP

proxy\_set\_header x-Real-IP $remote\_addr

# 用户请求的协议

proxy\_set\_header x-Scheme $scheme

# tornadoes是设置的一个名字，踏实一个负载均衡的组

proxy\_pass <http://tornadoes;>

}

## 配置websocket

**http:**

upstream websocket {

server 127.0.0.1:8000

}

**server:**

location /chat {

// websocket指定是nginx中的一个组

proxy\_pass <http://websocket/chat;>

proxy\_http\_version 1.1

proxy\_set\_header Upgrade $http\_upgrade

proxy\_set\_header connection “upgrade”

}