

随笔 - 187, 文章 - 0, 评论 - 0, 引用 - 0

supervisor使用详解

1.什么是supervisor

supervisor是用python写的一个进程管理工具，用来启动，重启，关闭进程。

2.supervisor的安装

```
pip install supervisor
```

3.supervisor的配置文件

supervisor安装完毕后，会有一个配置文件supervisord.conf运行

```
echo_supervisord_conf
```

命令，可输出文件详细

我们使用重定向运算符将配置文件定向到/etc路径下（方便管理）

```
echo_supervisord_conf>/etc/supervisord.conf
```

4.配置文件详解



```
[unix_http_server]
file=/tmp/supervisor.sock ; UNIX socket 文件, supervisorctl 会使用
;chmod=0700 ; socket 文件的 mode, 默认是 0700
;chown=nobody:nogroup ; socket 文件的 owner, 格式: uid:gid

[inet_http_server] ; HTTP 服务器, 提供 web 管理界面
;port=127.0.0.1:9001 ; Web 管理后台运行的 IP 和端口, 如果开放到公网, 需要注意安全性
;username=user ; 登录管理后台的用户名
;password=123 ; 登录管理后台的密码

[supervisord]
logfile=/tmp/supervisord.log ; 日志文件, 默认是 $CWD/supervisord.log
logfile_maxbytes=50MB ; 日志文件大小, 超出会 rotate, 默认 50MB
logfile_backups=10 ; 日志文件保留备份数量默认 10
loglevel=info ; 日志级别, 默认 info, 其它: debug, warn, trace
pidfile=/tmp/supervisord.pid ; pid 文件
nodaemon=false ; 是否在前台启动, 默认是 false, 即以 daemon 的方式启动
minfds=1024 ; 可以打开的文件描述符的最小值, 默认 1024
minprocs=200 ; 可以打开的进程数的最小值, 默认 200

; the below section must remain in the config file for RPC
; (supervisorctl/web interface) to work, additional interfaces may be
; added by defining them in separate rpcinterface: sections
[rpcinterface:supervisor]
supervisor.rpcinterface_factory = supervisor.rpcinterface:make_main_rpcinterface

[supervisorctl]
serverurl=unix:///tmp/supervisor.sock ; 通过 UNIX socket 连接 supervisord, 路径与
unix_http_server 部分的 file 一致
;serverurl=http://127.0.0.1:9001 ; 通过 HTTP 的方式连接 supervisord

; 包含其他的配置文件
[include]
files = relative/directory/*.ini ; 可以是 *.conf 或 *.ini
```



5.进程Program配置



```
[program:usercenter]
directory = /home/leon/projects/usercenter ; 程序的启动目录
command = gunicorn -c gunicorn.py wsgi:app ; 启动命令, 可以看出与手动在命令行启动的命令是一样的
autostart = true ; 在 supervisord 启动的时候也自动启动
startsecs = 5 ; 启动 5 秒后没有异常退出, 就当作已经正常启动了
autorestart = true ; 程序异常退出后自动重启
startretries = 3 ; 启动失败自动重试次数, 默认是 3
user = leon ; 用哪个用户启动
redirect_stderr = true ; 把 stderr 重定向到 stdout, 默认 false
stdout_logfile_maxbytes = 20MB ; stdout 日志文件大小, 默认 50MB
stdout_logfile_backups = 20 ; stdout 日志文件备份数
; stdout 日志文件, 需要注意当指定目录不存在时无法正常启动, 所以需要手动创建目录 (supervisord 会自动创建日志文件)
stdout_logfile = /data/logs/usercenter_stdout.log

; 可以通过 environment 来添加需要的环境变量, 一种常见的用法是修改 PYTHONPATH
; environment=PYTHONPATH=$PYTHONPATH:/path/to/somewhere
```



将进程配置添加到supervisord.conf文件中

推荐将各个进程配置放到单独的文件中, 然后通过[include]配置项引用

```
[include]
files = /etc/supervisor/*.conf
```

6.supervisor服务启动

运行命令

```
supervisord -c /etc/supervisord.conf
```

启动服务

若服务已启动会报以下错误：

```
[root@51dev ~]# supervisord -c /etc/supervisord.conf
Error: Another program is already listening on a port that one of our HTTP servers is configured to use. Shut this program down first before starting supervisord.
```

查找服务进程, kill, 重新运行

```
[root@51dev ~]# ps -ef | grep super
root      25478      1   0 17:52 ?           00:00:00 /usr/bin/python /usr/bin/supervisord -c /etc/supervisord.conf
root      26343  25096   0 18:20 pts/0    00:00:00 grep --color=auto super
```

7.supervisor服务客户端管理

可以使用supervisord的命令行客户端管理工具supervisorctl, 来管理进程。



```
$ supervisorctl status
$ supervisorctl stop usercenter
$ supervisorctl start usercenter
$ supervisorctl restart usercenter
$ supervisorctl reread
$ supervisorctl update
```



8.supervisor web管理界面

```
:[inet_http_server] ; HTTP 服务器, 提供 web 管理界面
;port=127.0.0.1:9001 ; Web 管理后台运行的 IP 和端口, 如果开放到公网, 需要注意安全性
;username=user ; 登录管理后台的用户名
;password=123 ; 登录管理后台的密码
```

在浏览器中输入http://127.0.0.1:9001, 可进入web管理界面