
Tue, Jun 9, 2015
python , development , tools
Comments

# 使用 supervisor 管理进程

Supervisor (http://supervisord.org) 是一个用 Python 写的进程管理工具，可以很方便的用来启动、重启、关闭进程（不仅仅是 Python 进程）。除了对单个进程的控制，还可以同时启动、关闭多个进程，比如很不幸的服务器出问题导致所有应用程序都被杀死，此时可以用 supervisor 同时启动所有应用程序而不是一个一个地敲命令启动。

## 安装

Supervisor 可以运行在 Linux、Mac OS X 上。如前所述，supervisor 是 Python 编写的，所以安装起来也很方便，可以直接用 pip：

```
sudo pip install supervisor
```

如果是 Ubuntu 系统，还可以使用 apt-get 安装。

## supervisord 配置

Supervisor 相当强大，提供了很丰富的功能，不过我们可能只需要用到其中一小部分。安装完成之后，可以编写配置文件，来满足自己的需求。为了方便，我们把配置分成两部分：supervisord（supervisor 是一个 C/S 模型的程序，这是 server 端，对应的有 client 端：supervisorctl）和应用程序（即我们要管理的程序）。

首先来看 supervisord 的配置文件。安装完 supervisor 之后，可以运行`echo_supervisord_conf` 命令输出默认的配置项，也可以重定向到一个配置文件里：

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

去除里面大部分注释和“不相关”的部分，我们可以先看这些配置：

```
[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_
;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
```

我们把上面这部分配置保存到 /etc/supervisord.conf（或其他任意有权限访问的文件），然后启动 supervisord（通过 -c 选项指定配置文件路径，如果不指定会按照这个顺序查找配置文件：$CWD/supervisord.conf, $CWD/etc/supervisord.conf, /etc/supervisord.conf）：

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

查看 supervisord 是否在运行：

```
ps aux | grep supervisord
```

# program 配置

上面我们已经把 supervisrod 运行起来了，现在可以添加我们要管理的进程的配置文件。这些配置可以都写到 supervisord.conf 文件里，如果应用程序很多，最好通过 include 的方式把不同的程序（组）写到不同的配置文件里。

为了举例，我们新建一个目录 /etc/supervisor/ 用于存放这些配置文件，相应的，把 /etc/supervisord.conf 里 include 部分的的配置修改一下：

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

假设有个用 Flask 开发的用户系统 usercenter, 生产环境使用 gunicorn 运行。项目代码位于 `/home/leon/projects/usercenter`，WSGI 对象位于 wsgi.py。在命令行启动的方式是这样的：

```
cd /home/leon/projects/usercenter
gunicorn -w 8 -b 0.0.0.0:17510 wsgi:app
```

对应的配置文件可能是：

```
[program:usercenter]
directory = /home/leon/projects/usercenter ; 程序的启动目录
command = gunicorn -w 8 -b 0.0.0.0:17510 wsgi:app  ; 启动命令
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
stdout_logfile = /data/logs/usercenter_stdout.log
```

其中 `[program:usercenter]` 中的 `usercenter` 是应用程序的唯一标识，不能重复。对该程序的所有操作（start, restart 等）都通过名字来实现。

### Tips 1: Python 环境

有两种方式指定程序使用的 Python 环境：

1. `command` 使用绝对路径。假设使用 pyenv 来管理 Python 环境，上面例子中的 gunicorn 路径可以替换为 `/home/leon/.pyenv/versions/usercenter/bin/gunicorn`. 这种方式一目了然，推荐。
2. 通过 `environment` 配置 `PYTHONPATH`.
   `environment=PYTHONPATH=$PYTHONPATH:/home/leon/.pyenv/versions/usercenter/bin/.`
   `environment` 这个配置项非常有用，可以用来给程序传入环境变量。

### Tips 2: 后台进程

Supervisor 只能管理在前台运行的程序，所以如果应用程序有后台运行的选项，需要关闭。

### Tips 3: 子进程

有时候用 Supervisor 托管的程序还会有子进程（如 Tornado），如果只杀死主进程，子进程就可能变成孤儿进程。通过这两项配置来确保所有子进程都能正确停止：

```
stopasgroup=true
killasgroup=true
```

## 使用 supervisorctl

Supervisorctl 是 supervisord 的一个命令行客户端工具，启动时需要指定与 supervisord 使用同一份配置文件，否则与 supervisord 一样按照顺序查找配置文件。

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

上面这个命令会进入 supervisorctl 的 shell 界面，然后可以执行不同的命令了：

```
> status     # 查看程序状态
> stop usercenter    # 关闭 usercenter 程序
> start usercenter   # 启动 usercenter 程序
> restart usercenter     # 重启 usercenter 程序
> reread     #  读取有更新（增加）的配置文件，不会启动新添加的程序
> update     #  重启配置文件修改过的程序
```

上面这些命令都有相应的输出，除了进入 supervisorctl 的 shell 界面，也可以直接在 bash 终端运行：

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

# 其它

除了 supervisorctl 之外，还可以配置 supervisrod 启动 web 管理界面，这个 web 后台使用 Basic Auth 的方式进行身份认证。

除了单个进程的控制，还可以配置 group，进行分组管理。

经常查看日志文件，包括 supervisord 的日志和各个 pragram 的日志文件，程序 crash 或抛出异常的信息一半会输出到 stderr，可以查看相应的日志文件来查找问题。

Supervisor 有很丰富的功能，还有其他很多项配置，可以在官方文档获取更多信息：
http://supervisord.org/index.html

# Comments