https://blog.csdn.net/huwh\_/article/details/80497790

**Supervisor的使用**

2018年05月29日 16:05:16 [胡伟煌](https://me.csdn.net/huwh_) 阅读数 7319

 版权声明：本文为博主原创文章，未经博主允许不得转载。如需转载请联系本人，并标明作者和出处。 https://blog.csdn.net/huwh\_/article/details/80497790

本文个人博客地址：<http://www.huweihuang.com/article/linux/supervisor-usage/>

**1. Supervisor简介**

Supervisord 是用 Python 实现的一款的进程管理工具，supervisord 要求管理的程序是非 daemon 程序，supervisord 会帮你把它转成 daemon 程序，因此如果用 supervisord 来管理进程，进程需要以非daemon的方式启动。

例如：管理nginx 的话，必须在 nginx 的配置文件里添加一行设置 daemon off 让 nginx 以非 daemon 方式启动。

**2. Supervisor安装**

以centos系统为例，以下两种方式选择其一。

# yum install 的方式

yum install -y supervisor

# easy\_install的方式

yum install -y python-setuptools

easy\_install supervisor

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

* 1
* 2
* 3
* 4
* 5
* 6
* 7

**3. Supervisor的配置**

**3.1. supervisord.conf的配置**

如果使用yum install -y supervisor的命令安装，会生成默认配置/etc/supervisord.conf和目录/etc/supervisord.d，如果没有则自行创建。

在/etc/supervisord.d的目录下创建conf和log两个目录，conf用于存放管理进程的配置，log用于存放管理进程的日志。

cd /etc/supervisord.d

mkdir conf log

* 1
* 2

修改/etc/supervisord.conf的[include]部分，即载入/etc/supervisord.d/conf目录下的所有配置。

vi /etc/supervisord.conf

...

[include]

files = supervisord.d/conf/\*.conf

...

* 1
* 2
* 3
* 4
* 5

也可以修改supervisor应用日志的目录，默认日志路径为/var/log/supervisor/supervisord.log。

vi /etc/supervisord.conf

...

[supervisord]

logfile=/var/log/supervisor/supervisord.log ; (main log file;default $CWD/supervisord.log)

logfile\_maxbytes=50MB ; (max main logfile bytes b4 rotation;default 50MB)

logfile\_backups=10 ; (num of main logfile rotation backups;default 10)

loglevel=info ; (log level;default info; others: debug,warn,trace)

pidfile=/var/run/supervisord.pid ; (supervisord pidfile;default supervisord.pid)

...

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9

**3.2. 管理应用的配置**

进入到/etc/supervisord.d/conf目录，创建管理应用的配置，可以创建多个应用配置。

例如，创建confd.conf配置。

[program:confd]

directory = /usr/local/bin ; 程序的启动目录

command = /usr/local/bin/confd -config-file /etc/confd/confd.toml ; 启动命令，与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 = root ; 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 = /etc/supervisord.d/log/confd.log ;日志统一放在log目录下

*; 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*

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15

**4. Surpervisor的启动**

# supervisord二进制启动

supervisord -c /etc/supervisord.conf

# 检查进程

ps aux | grep supervisord

* 1
* 2
* 3
* 4

或者以systemd的方式管理

vi /etc/rc.d/init.d/supervisord

#!/bin/sh

*#*

*# /etc/rc.d/init.d/supervisord*

*#*

*# Supervisor is a client/server system that*

*# allows its users to monitor and control a*

*# number of processes on UNIX-like operating*

*# systems.*

*#*

*# chkconfig: - 64 36*

*# description: Supervisor Server*

*# processname: supervisord*

*# Source init functions*

. /etc/rc.d/init.d/functions

prog="supervisord"

prefix="/usr"

exec\_prefix="${prefix}"

prog\_bin="${exec\_prefix}/bin/supervisord"

PIDFILE="/var/run/$prog.pid"

start()

{

echo -n $"Starting $prog: "

daemon $prog\_bin --pidfile $PIDFILE -c /etc/supervisord.conf

[ -f $PIDFILE ] && success $"$prog startup" || failure $"$prog startup"

echo

}

stop()

{

echo -n $"Shutting down $prog: "

[ -f $PIDFILE ] && killproc $prog || success $"$prog shutdown"

echo

}

case "$1" in

start)

start

;;

stop)

stop

;;

status)

status $prog

;;

restart)

stop

start

;;

\*)

echo "Usage: $0 {start|stop|restart|status}"

;;

esac

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62

设置开机启动及systemd方式启动。

sudo chmod +x /etc/rc.d/init.d/supervisord

sudo chkconfig --add supervisord

sudo chkconfig supervisord on

sudo service supervisord start

* 1
* 2
* 3
* 4

**5. supervisorctl&supervisord**

Supervisord 安装完成后有两个可用的命令行 supervisord 和 supervisorctl，命令使用解释如下：

**5.1. supervisorctl**

* supervisorctl stop programxxx，停止某一个进程(programxxx)，programxxx 为 [program:beepkg] 里配置的值，这个示例就是 beepkg。
* supervisorctl start programxxx，启动某个进程。
* supervisorctl restart programxxx，重启某个进程。
* supervisorctl status，查看进程状态。
* supervisorctl stop groupworker ，重启所有属于名为 groupworker 这个分组的进程(start,restart 同理)。
* supervisorctl stop all，停止全部进程，注：start、restart、stop 都不会载入最新的配置文件。
* supervisorctl reload，载入最新的配置文件，停止原有进程并按新的配置启动、管理所有进程。
* supervisorctl update，根据最新的配置文件，启动新配置或有改动的进程，配置没有改动的进程不会受影响而重启。

更多参考：

$ supervisorctl --help

supervisorctl -- control applications run by supervisord from the cmd line.

Usage: /usr/bin/supervisorctl [options] [action [arguments]]

Options:

-c/--configuration -- configuration file path (default /etc/supervisord.conf)

-h/--help -- print usage message and exit

-i/--interactive -- start an interactive shell after executing commands

-s/--serverurl URL -- URL on which supervisord server is listening

(default "http://localhost:9001").

-u/--username -- username to use for authentication with server

-p/--password -- password to use for authentication with server

-r/--history-file -- keep a readline history (if readline is available)

action [arguments] -- see below

Actions are commands like "tail" or "stop". If -i is specified or no action is

specified on the command line, a "shell" interpreting actions typed

interactively is started. Use the action "help" to find out about available

actions.

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21

例如：

# supervisorctl status

confd RUNNING pid 31256, uptime 0:11:24

twemproxy RUNNING pid 31255, uptime 0:11:24

* 1
* 2
* 3

**5.2. supervisord**

* supervisord，初始启动 Supervisord，启动、管理配置中设置的进程。

$ supervisord --help

supervisord -- run a set of applications as daemons.

Usage: /usr/bin/supervisord [options]

Options:

-c/--configuration FILENAME -- configuration file

-n/--nodaemon -- run in the foreground (same as 'nodaemon true' in config file)

-h/--help -- print this usage message and exit

-v/--version -- print supervisord version number and exit

-u/--user USER -- run supervisord as this user (or numeric uid)

-m/--umask UMASK -- use this umask for daemon subprocess (default is 022)

-d/--directory DIRECTORY -- directory to chdir to when daemonized

-l/--logfile FILENAME -- use FILENAME as logfile path

-y/--logfile\_maxbytes BYTES -- use BYTES to limit the max size of logfile

-z/--logfile\_backups NUM -- number of backups to keep when max bytes reached

-e/--loglevel LEVEL -- use LEVEL as log level (debug,info,warn,error,critical)

-j/--pidfile FILENAME -- write a pid file for the daemon process to FILENAME

-i/--identifier STR -- identifier used for this instance of supervisord

-q/--childlogdir DIRECTORY -- the log directory for child process logs

-k/--nocleanup -- prevent the process from performing cleanup (removal of

old automatic child log files) at startup.

-a/--minfds NUM -- the minimum number of file descriptors for start success

-t/--strip\_ansi -- strip ansi escape codes from process output

--minprocs NUM -- the minimum number of processes available for start success

--profile\_options OPTIONS -- run supervisord under profiler and output

results based on OPTIONS, which is a comma-sep'd

list of 'cumulative', 'calls', and/or 'callers',

e.g. 'cumulative,callers')

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29

**6. Supervisor控制台**

在/etc/supervisord.conf中修改[inet\_http\_server]的参数，具体如下：

[inet\_http\_server] ; inet (TCP) server disabled by default

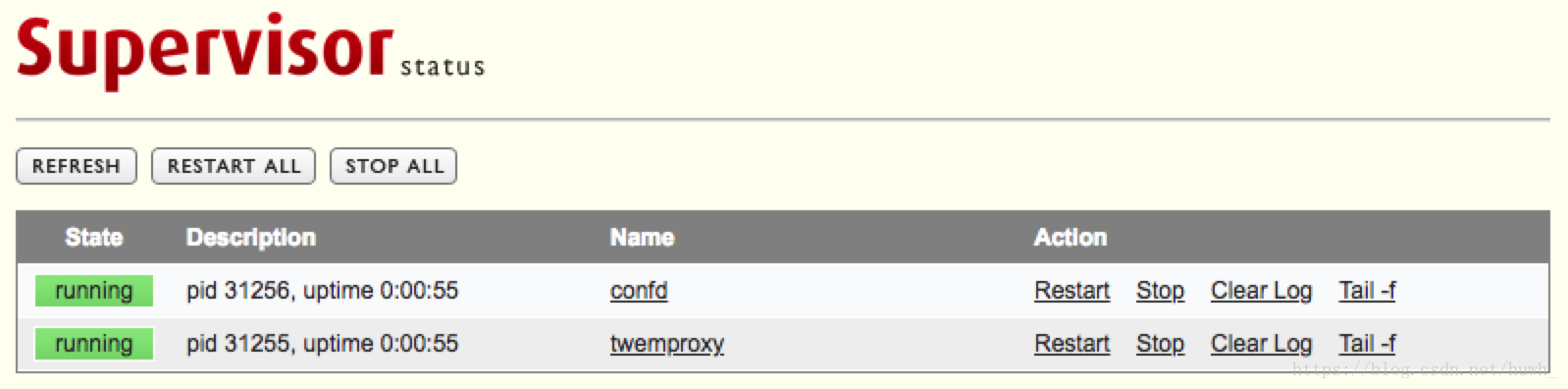
port=\*:9001 ; ip\_address:port specifier, \*:port for all iface

username=root ; default is no username (open server)

password=xxxx ; default is no password (open server)

* 1
* 2
* 3
* 4

修改后重启supervisor进程，在浏览器访问 http://<host-ip>:9001。

具体如下：   


**7. supervisor.conf详细配置**

cat /etc/supervisord.conf

; Sample supervisor config file.

[unix\_http\_server]

file=/var/run/supervisor/supervisor.sock ; (the path to the socket file)

;chmod=0700 ; sockef file mode (default 0700)

;chown=nobody:nogroup ; socket file uid:gid owner

;username=user ; (default is no username (open server))

;password=123 ; (default is no password (open server))

;[inet\_http\_server] ; inet (TCP) server disabled by default

;port=127.0.0.1:9001 ; (ip\_address:port specifier, \*:port for all iface)

;username=user ; (default is no username (open server))

;password=123 ; (default is no password (open server))

[supervisord]

logfile=/var/log/supervisor/supervisord.log ; (main log file;default $CWD/supervisord.log)

logfile\_maxbytes=50MB ; (max main logfile bytes b4 rotation;default 50MB)

logfile\_backups=10 ; (num of main logfile rotation backups;default 10)

loglevel=info ; (log level;default info; others: debug,warn,trace)

pidfile=/var/run/supervisord.pid ; (supervisord pidfile;default supervisord.pid)

nodaemon=false ; (start in foreground if true;default false)

minfds=1024 ; (min. avail startup file descriptors;default 1024)

minprocs=200 ; (min. avail process descriptors;default 200)

;umask=022 ; (process file creation umask;default 022)

;user=chrism ; (default is current user, required if root)

;identifier=supervisor ; (supervisord identifier, default is 'supervisor')

;directory=/tmp ; (default is not to cd during start)

;nocleanup=true ; (don't clean up tempfiles at start;default false)

;childlogdir=/tmp ; ('AUTO' child log dir, default $TEMP)

;environment=KEY=value ; (key value pairs to add to environment)

;strip\_ansi=false ; (strip ansi escape codes in logs; def. false)

; 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:///var/run/supervisor/supervisor.sock ; use a unix:// URL for a unix socket

;serverurl=http://127.0.0.1:9001 ; use an http:// url to specify an inet socket

;username=chris ; should be same as http\_username if set

;password=123 ; should be same as http\_password if set

;prompt=mysupervisor ; cmd line prompt (default "supervisor")

;history\_file=~/.sc\_history ; use readline history if available

; The below sample program section shows all possible program subsection values,

; create one or more 'real' program: sections to be able to control them under

; supervisor.

;[program:theprogramname]

;command=/bin/cat ; the program (relative uses PATH, can take args)

;process\_name=%(program\_name)s ; process\_name expr (default %(program\_name)s)

;numprocs=1 ; number of processes copies to start (def 1)

;directory=/tmp ; directory to cwd to before exec (def no cwd)

;umask=022 ; umask for process (default None)

;priority=999 ; the relative start priority (default 999)

;autostart=true ; start at supervisord start (default: true)

;autorestart=true ; retstart at unexpected quit (default: true)

;startsecs=10 ; number of secs prog must stay running (def. 1)

;startretries=3 ; max # of serial start failures (default 3)

;exitcodes=0,2 ; 'expected' exit codes for process (default 0,2)

;stopsignal=QUIT ; signal used to kill process (default TERM)

;stopwaitsecs=10 ; max num secs to wait b4 SIGKILL (default 10)

;user=chrism ; setuid to this UNIX account to run the program

;redirect\_stderr=true ; redirect proc stderr to stdout (default false)

;stdout\_logfile=/a/path ; stdout log path, NONE for none; default AUTO

;stdout\_logfile\_maxbytes=1MB ; max # logfile bytes b4 rotation (default 50MB)

;stdout\_logfile\_backups=10 ; # of stdout logfile backups (default 10)

;stdout\_capture\_maxbytes=1MB ; number of bytes in 'capturemode' (default 0)

;stdout\_events\_enabled=false ; emit events on stdout writes (default false)

;stderr\_logfile=/a/path ; stderr log path, NONE for none; default AUTO

;stderr\_logfile\_maxbytes=1MB ; max # logfile bytes b4 rotation (default 50MB)

;stderr\_logfile\_backups=10 ; # of stderr logfile backups (default 10)

;stderr\_capture\_maxbytes=1MB ; number of bytes in 'capturemode' (default 0)

;stderr\_events\_enabled=false ; emit events on stderr writes (default false)

;environment=A=1,B=2 ; process environment additions (def no adds)

;serverurl=AUTO ; override serverurl computation (childutils)

; The below sample eventlistener section shows all possible

; eventlistener subsection values, create one or more 'real'

; eventlistener: sections to be able to handle event notifications

; sent by supervisor.

;[eventlistener:theeventlistenername]

;command=/bin/eventlistener ; the program (relative uses PATH, can take args)

;process\_name=%(program\_name)s ; process\_name expr (default %(program\_name)s)

;numprocs=1 ; number of processes copies to start (def 1)

;events=EVENT ; event notif. types to subscribe to (req'd)

;buffer\_size=10 ; event buffer queue size (default 10)

;directory=/tmp ; directory to cwd to before exec (def no cwd)

;umask=022 ; umask for process (default None)

;priority=-1 ; the relative start priority (default -1)

;autostart=true ; start at supervisord start (default: true)

;autorestart=unexpected ; restart at unexpected quit (default: unexpected)

;startsecs=10 ; number of secs prog must stay running (def. 1)

;startretries=3 ; max # of serial start failures (default 3)

;exitcodes=0,2 ; 'expected' exit codes for process (default 0,2)

;stopsignal=QUIT ; signal used to kill process (default TERM)

;stopwaitsecs=10 ; max num secs to wait b4 SIGKILL (default 10)

;user=chrism ; setuid to this UNIX account to run the program

;redirect\_stderr=true ; redirect proc stderr to stdout (default false)

;stdout\_logfile=/a/path ; stdout log path, NONE for none; default AUTO

;stdout\_logfile\_maxbytes=1MB ; max # logfile bytes b4 rotation (default 50MB)

;stdout\_logfile\_backups=10 ; # of stdout logfile backups (default 10)

;stdout\_events\_enabled=false ; emit events on stdout writes (default false)

;stderr\_logfile=/a/path ; stderr log path, NONE for none; default AUTO

;stderr\_logfile\_maxbytes=1MB ; max # logfile bytes b4 rotation (default 50MB)

;stderr\_logfile\_backups ; # of stderr logfile backups (default 10)

;stderr\_events\_enabled=false ; emit events on stderr writes (default false)

;environment=A=1,B=2 ; process environment additions

;serverurl=AUTO ; override serverurl computation (childutils)

; The below sample group section shows all possible group values,

; create one or more 'real' group: sections to create "heterogeneous"

; process groups.

;[group:thegroupname]

;programs=progname1,progname2 ; each refers to 'x' in [program:x] definitions

;priority=999 ; the relative start priority (default 999)

; The [include] section can just contain the "files" setting. This

; setting can list multiple files (separated by whitespace or

; newlines). It can also contain wildcards. The filenames are

; interpreted as relative to this file. Included files \*cannot\*

; include files themselves.

[include]

files = supervisord.d/conf*/\*.conf*

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68
* 69
* 70
* 71
* 72
* 73
* 74
* 75
* 76
* 77
* 78
* 79
* 80
* 81
* 82
* 83
* 84
* 85
* 86
* 87
* 88
* 89
* 90
* 91
* 92
* 93
* 94
* 95
* 96
* 97
* 98
* 99
* 100
* 101
* 102
* 103
* 104
* 105
* 106
* 107
* 108
* 109
* 110
* 111
* 112
* 113
* 114
* 115
* 116
* 117
* 118
* 119
* 120
* 121
* 122
* 123
* 124
* 125
* 126
* 127
* 128
* 129

参考：

<http://supervisord.org/>