Supervisord Cheat Sheet

by misterrabinhalder via cheatography.com/35799/cs/19115/

Running supervisord

-c FILE	path to supervisord configuration file
-n	run supervisord in the foreground
-u USER	UNIX username or numeric user id
-l FILE	filename path to use as the supervisord activity log
-e LEVEL	Valid levels are trace, debug, info, warn, error, and critical
-j FILE	filename to which supervisord should write its pid file

Running supervisorctl

-c FILE	configuration file path
-u USER	username to use for authentication with server
-p PASSWORD	password to use for authentication with server
-r	keep a readline history

Supervisord Signal

SIGTERM	supervisord and all its subprocesses will shut down
SIGINT	supervisord and all its subprocesses will shut down
SIGQUIT	supervisord and all its subprocesses will shut down
SIGHUP	supervisord will stop all processes, reload the configuration from the first config file it finds, and start all processes
SIGUSR2	supervisord will close and reopen the main activity log and all child log files

Installing supervisord

Installing With Pip

```
pip install supervisor
```

Install On Debian

```
apt-get install supervisor
```

Install On Redhat

```
yum install supervisor
```

Install On Archlinux

```
pacman -S supervisor
```

Using Environment Variables

```
[program:example]
comman d=/ usr /bi n/e xample --
logl eve l=% (EN V_L OGL EVEL)s
```

Environment variables that are present in the environment at the time that supervisord is started can be used in the configuration file using the Python string expression syntax %(ENV_X)s

Using UNIX domain socket

```
[unix_http_server]
file = /tmp/s upe rvi sor.sock
chmod = 0777
chown= nobody :no group
username = user
password = 123
```

HTTP server that listens on a UNIX domain socket

Using process group

```
[group:foo]
progra ms= bar,baz
priori ty=999
```

Group “homogeneous” process groups (aka “programs”) together into a “heterogeneous” process group

Using TCP domain socket

```
[inet_http_server]
port = 127.0.0.1 :9001
username = user
password = 123
```

HTTP server that listens on a TCP (internet) socket

Using supervisord

```
[supervisord]
logfile = /tmp/s upe rvi sor -
d.log
logfil e_m axbytes = 50MB
logfil e_b ack ups=10
loglevel = info
pidfile = /tmp/s upe rvi sor -
d.pid
nodaemon = false
minfds = 1024
minprocs = 200
umask = 022
user = chrism
identifier = supervisor
directory = /tmp
nocleanup = true
childl ogdir = /tmp
strip_ansi = false
enviro nment = KEY1="v alu -
e1", KEY 2="v alu e2"
```

Global settings related to the supervisord

Using supervisorctl

```
[supervisorctl]
serverurl = unix:/ //t mp/ sup -
erv iso r.sock
username = chris
password = 123
prompt = mysupe rvisor
```

Supervisorctl interactive shell program

By **misterrabinhalder**

cheatography.com/misterrabinhalder/

Published 28th March, 2019.
Last updated 27th March, 2019.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Using program

```
[program:cat]
comman d=/ bin/cat
proces s_n ame =%( pro gra m_n -
ame)s
numprocs=1
direct ory =/tmp
umask=022
priori ty=999
autost art =true
autore sta rt= une xpected
starts ecs=10
startr etr ies=3
exitco des=0,2
stopsi gna l=TERM
stopwa its ecs=10
stopas gro up= false
killas gro up= false
user=c hrism
redire ct_ std err =false
stdout _lo gfi le= /a/path
stdout _lo gfi le_ max byt -
es=1MB
stdout _lo gfi le_ bac kups=10
stdout _ca ptu re_ max byt -
es=1MB
stdout _ev ent s_e nab led -
=false
stderr _lo gfi le= /a/path
stderr _lo gfi le_ max byt -
es=1MB
stderr _lo gfi le_ bac kups=10
stderr _ca ptu re_ max byt -
es=1MB
stderr _ev ent s_e nab led -
=false
enviro nme nt= A="1 " ,B= " 2"
server url =AUTO
```

Supervisord to know which programs it should start and control

Using configuration directory

```
[include]
files = superv iso rd.d /*.conf
```

Include configuration files from drop in directory

Log Level

critical	CRIT
error	ERRO
warn	WARN
info	INFO
debug	DEBG
trace	TRAC
blather	BLAT

Supervisorctl actions

help	print a list of available actions
help <action>	print help for <action>
add <name> [...]	activates any updates in config for process/group
remove <name> [...]	removes process/group from active config
update	reload config and then add and remove as necessary
clear <name>	clear a process' log files
clear <name> <name>	clear multiple process' log files
clear all	clear all process' log files
fg <process>	connect to a process in foreground mode Press Ctrl+C to exit foreground

Supervisorctl actions (cont)

pid	get the PID of supervisord
pid <name>	get the PID of a single child process by name
pid all	get the PID of every child process, one per line
reload	restarts the remote supervisord
reread	reload the daemon's configuration files, without add/remove (no restarts)
restart <name>	restart a process
restart <gname>:*	restart all processes in a group
restart <name> <name>	restart multiple processes
restart all	restart all processes
signal	No help on signal
start <name>	start a process
start <gname>:*	start all processes in a group
start <name> <name>	start multiple processes
start all	start all processes
status	get all process status info
status <name>	get status on a single process by name
status <name> <name>	get status on multiple named processes
stop <name>	stop a process
stop <gname>:*	stop all processes in a group

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Supervisorctl actions (cont)	
stop <name> <name>	stop multiple processes or groups
stop all	stop all processes
tail [-f] <name> [stdout|stderr]	output the last part of process logs

By **misterrabinhalder**

cheatography.com/misterrabinhalder/

Published 28th March, 2019.
Last updated 27th March, 2019.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com