Supervisor

A Process Control System

Sergej Kurakin

.NF3

Need for Long Running Scripts under Linux

- Job Queues in PHP (or any other language)
- Application Servers in PHP (or any other language)
- NodeJS Applications
- Selenium WebDriver + Xvfb
- You name it

Solutions

- Self-Made Daemons
- SysV / systemd / launchd
- Screen or tmux
- nohup
- Five star crons
- Daemontools
- Might be more possible solutions
- Supervisor
- Docker

Needed options

- ☐ Automatic start
- ☐ Automatic restart
- ☐ Execute under some user account
- ☐ Logs (stdout and stderr)
- ☐ Easy to use

Self-Made Daemons

- Wrote once by myself
- I will never do it twice
- Third-party is not always good or maintainable
- Lots of debugging
- All features must be implemented by yourself
- Depends on SysV/launchd/systemd
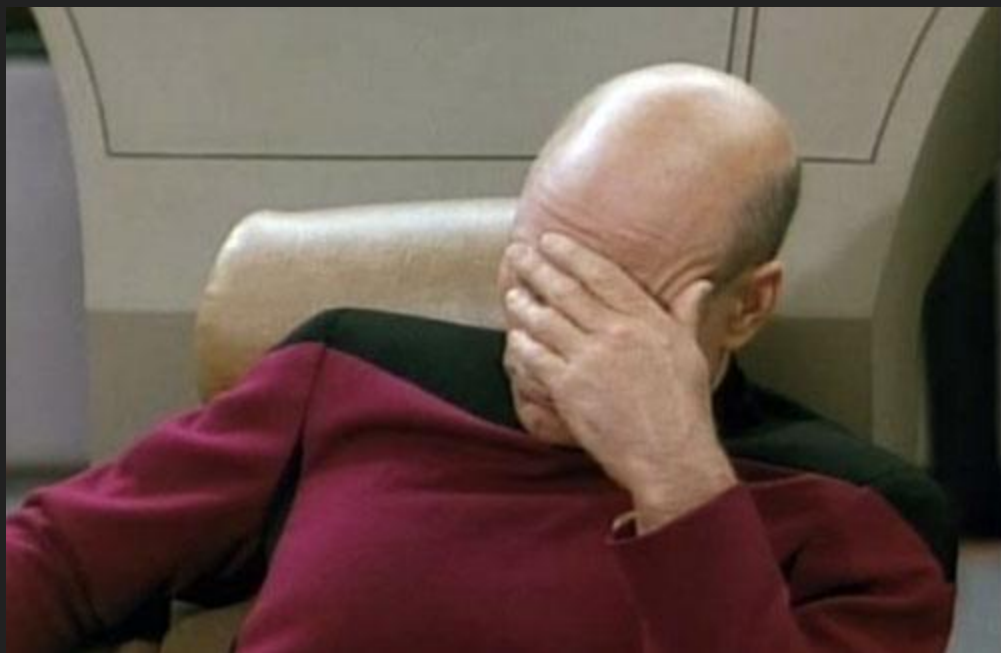
Screen / tmux or nohup

- No automatic start
- No automatic restart
- Logs...

daemontools

- Old (but not obsolete)
- Does not work under some virtualizations
- Might be hard to configure

Five star cron

- Still popular and usable
- Perfect for job queues (in some situations)

Docker

This topic deserves a separate talk from someone who has good experience with Docker in production environment.

Supervisor!

Supervisor?

Supervisor is a client/server system that allows its users to monitor and control a number of processes on UNIX-like operating systems.

Features

- Simple - INI-style files
- Centralized - one place to manage
- Efficient - fork, don't daemonize
- Extensible - events and XML-RPC interface
- Compatible - Linux, Mac OS X, Solaris and FreeBSD
- Proven

Components

supervisord

supervisorctl

Installiation

- apt-get install supervisor
- yum install supervisor

- easy_install supervisor
- pip install supervisor

Configuration files

/etc/supervisor/supervisord.conf

/etc/supervisor/conf.d/*.conf

/etc/supervisord.conf

/etc/supervisord.d/*.ini

Simplest process setup

[program:myprocess]

command=/usr/bin/php -f /home/user/long.php

Warning: you run this script as *root*! This is for demo only!!!

Control your processes

supervisorctl status

supervisorctl update

supervisorctl stop <name>

supervisorctl start <name>

supervisorctl restart <name>

What about logs?

See in /var/log/supervisor/

supervisorctl maintail -f

supervisorctl tail -f <name> stdout

supervisorctl tail -f <name> stderr

You gona like it (program section)!

numprocs

directory

umask

autostart

autorestart (false, unexpected, true)

user

environment

startretries

exitcodes

More on command line

clear <name>|all

fg <name>

pid

version

restart <name>

start <name>

stop <name>

reread

update

Many processes

[program:myother]

command=/usr/bin/php -f /home/user/other.php

user=user

numprocs=2

process_name=%(program_name)s_%(process_num)s

Groups!

[group:mygroup]

programs=myprocess,myother

priority=999

Control whole group

restart <group name>:*

start <group name>:*

stop <group name>:*

Control process in group

restart <group name>:<name>

start <group name>:<name>

stop <group name>:<name>

Needed options

- ✔ Automatic start
- ✔ Automatic restart
- ✔ Execute under some user account
- ✔ Logs (stdout and stderr)
- ✔ Easy to use

Not covered topics

- Logs Capture Mode
- Events and Event Listeners
- SupervisorD configuration
- Process monitoring (Munin, Zabbix, DataDog)
- Lots of details you must figure out

Some notes on Long Running Scripts

- Watch for resources (memory usage), GC or restart
- Be ready for connection timeouts (MySQL, Redis, etc.)
- Clear caches between calls (Doctrine ORM, Statcache, etc.)
- Handle exceptions better

Links

http://supervisord.org/

https://www.digitalocean.com/community/tutorials/how-to-install-and-manage-supervisor-on-ubuntu-and-debian-vps

https://code.tutsplus.com/tutorials/monitor-and-control-applications-using-supervisor-part-1--cms-23770

Demo files https://github.com/sergej-kurakin/supervisord-demo

Discussion

Sergej Kurakin

Work @mail: sergej.kurakin@nfq.lt

Personal @mail: sergej@kurakin.info

https://www.linkedin.com/in/sergejkurakin