supervisor是一个守护服务，当被守护的服务异常宕掉后，守护服务会自动将他重启。是一个非常重要的运维利器。尤其对于python，php这类运行时抛异常的服务尤为重要，在过往特定的环境下，我们曾经单机用supervisor守护300~500个python进程，很爽。

**本文将会聊一聊如何部署和使用，以及如何通过后台总览，特别是使用中的一些坑。**

**目录：**

**(1).supervisor安装**

1.安装supervisor

2.守护一个服务

3.常用命令

**(2).特别注意**

**(3).后台服务supervisor-monitor**

1.php7安装

2.supervisor开启管理端口

3.部署supervisor-monitor

4.部署nginx&配置supervisor-monitor

**(4).行百里者半九十**

**(5).参考资料**

**(1).supervisor安装**

**1.安装supervisor**

yum -y install python-setuptools

easy\_install supervisor 或者 pip install supervisor

查看supervisor版本号：

supervisord -v

mkdir /etc/supervisor

mkdir /data/xx/logs/supervisor

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

修改文件/etc/supervisord.conf：

logfile=/data/xx/logs/supervisor/supervisord.log

vim /etc/supervisord.conf

;修改include配置，去除注释

[include]

files = /etc/supervisor/\*.conf

启动supervisor：

/usr/bin/supervisord -c /etc/supervisord.conf

**2.守护一个服务**

在/etc/supervisor下建立conf文件，如：cdn-release-web-server.conf

[program:cdn-release-web-server]

user=hpy

command=sh /app/xx/cdn-release-web-server/start.sh

autostart=true

autorestart=true

stdout\_logfile=/data/xx/logs/supervisor/cdn-release-web-server.log ; stdout log path, NONE for none; default AUTO

stdout\_logfile\_maxbytes=100MB   ; max # logfile bytes b4 rotation (default 50MB)

stdout\_logfile\_backups=10     ; # of stdout logfile backups (0 means none, default 10)

stderr\_logfile=/data/xx/logs/supervisor/cdn-release-web-server.error.log        ; stderr log path, NONE for none; default AUTO

stderr\_logfile\_maxbytes=100MB   ; max # logfile bytes b4 rotation (default 50MB)

stderr\_logfile\_backups=10

priority=1

numprocs=1

startretries=100

stopwaitsecs=2

stopasgroup=true

重要参数说明：

stopasgroup：表示当stop掉守护进程后，是否杀掉其子进程；true表示会杀掉子进程。

然后执行如下命令让其生效，并且启动被守护的服务：

supervisorctl update

执行supervisorctl进入supervisor命令行：

可以执行status等命令进行操作。

**3.常用命令**

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。

**(2).特别注意**

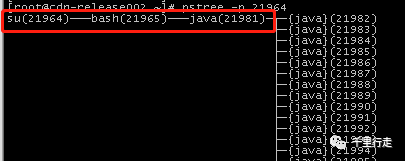
在开启stopasgroup=true的前提下，如果这么写：

command=su - hpy-c "/app/xx/cdn-release-web-server/start.sh"

是有问题的，start后你会发现有三个进程，比如：



使用pstree -p 21964查看进程树：



**此时如果在supervisorctl命令行中stop守护进程/或者被linux-os杀掉，只会把守护进程21964和其子进程21965杀掉，孙进程21981(实际服务)不会被杀掉，那么当你在start守护进程时，supervisor就会再新启动21981孙进程对应的服务，但是孙进程服务的端口号已经被占用，supervisor实际上是无法启动被守护服务的，就会反复尝试启动，通过监控你可能就会看到机器的cpu和memory的使用率情况非常的诡异。**

**(3).后台服务supervisor-monitor**

先看一个效果：



作为一个开源作品，页面还是挺赞的，我最爱的简约风格。

这是用php写的，所以需要安装php，nginx。

**1.php7安装**

安装epel-release

通过命令：

rpm -ivh http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

安装PHP7

终端再次运行如下命令：

rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm

成功获取PHP7的yum源，然后再执行：

yum install php70w

安装所有插件：这里主要用到php-fpm

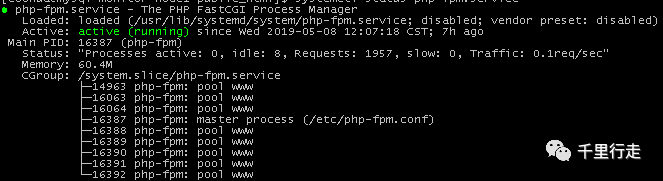
yum install php70w\*

启动php-fpm：

service php-fpm start

查看php-fpm服务状态：

systemctl status php-fpm.service



**2.supervisor开启管理端口**

要在每个机器的/etc/supervisord.conf配置文件中开启inet\_http\_server：

[inet\_http\_server]

port=\*:9001

username="yourusername"

password="yourpass"

暴露9001端口给supervisor-monitor获取数据和进行supervisor相关操作。

**然后需要重启supervisor。**

**3.部署supervisor-monitor**

下载代码：

git clone https://github.com/mlazarov/supervisord-monitor.git

copy配置文件：

cp supervisord-monitor/application/config/supervisor.php.example supervisord-monitor/application/config/supervisor.php

vim application/config/supervisor.php

修改用户名和密码：

$config['supervisor\_servers'] = array(

    'cdn-release-web-server-001' => array(

        'url' => 'http://supervisor-1/RPC2',

        'port' => '9001',

        'username' => 'yourusername',

        'password' => 'yourpass'

    ),

    'cdn-release-web-server-002' => array(

        'url' => 'http://supervisor-2/RPC2',

        'port' => '9001',

        'username' => 'yourusername',

        'password' => 'yourpass'

    ),

);

**4.部署nginx&配置supervisor-monitor**

部署过程略。

在nginx.conf中加入配置：

server {

    listen 80;

    server\_name  super.com;

    set $web\_root  /app/3rd/supervisord-monitor/public\_html;   #你的supervisor-monitor目录

    root $web\_root;

    index  index.php index.html index.htm;

    location / {

        try\_files $uri $uri/ /index.php;

    }

    location ~ /\*\.php$ {

        fastcgi\_index index.php;

        fastcgi\_param  SCRIPT\_FILENAME $web\_root$fastcgi\_script\_name;

        fastcgi\_param  SCHEME $scheme;

        include        fastcgi\_params;

        fastcgi\_pass 127.0.0.1:9000; #php-fpm默认端口是9000

    }

    access\_log  /data/xx/logs/nginx/supervisor-monitor.access.log;

    error\_log   /data/xx/logs/nginx/supervisor-monitor.error.log;

}

**为了安全，必须增加nginx的访问密码，主要是权限收敛防止误操作(supervisor-monitor支持直接start/stop等操作)：**

htpasswd -c passwd supervisor-monitor

启动nginx，本地配置host即可访问，输入密码进入页面。



**(4).行百里者半九十**

**一个服务上线仅仅是刚刚开始，让他高效稳定的24\*7运行需要太多太多的工作去处理。**

**而supervisor守护服务给我们提供了一个相对保障，允许我们有时间进行服务的优化&保障工作，这是非常重要的。**

**个人认为，supervisor守护服务在防守的同时，更是为了提供更好的进攻。**

**(5).参考资料**

1.supervisor官方资料：

http://supervisord.org/installing.html

2.supervisor-monitor官方git：

https://github.com/hepyu/supervisord-monitor