链滴

Supervisor 实战

作者：lucianolixin

原文链接：https://ld246.com/article/1572752081620

来源网站：链滴

前言

当我们部署线上服务的时候，特别是像Golang这种常驻进程，难免因为一些问题导致程序异常，引发anic。这时就需要一个脚本或者工具，去重新拉起进程，当服务少、语言单一的时候可以写一个守护程，或者脚本来监测对应的程序是否退出，当部署的程序比较多，实现的语言比较多样的时候，用Supervisor这样的工具更适合。

Supervisor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（守护进）。可以用他来管理自己的“服务程序”。

安装

Supervisor 以来Python，Mac下自带。

安装Supervisor

```
sudo pip install supervisor
```

默认安装在 /usr/local/bin 目录下

生成配置到指定目录

```
sudo echo_supervisord_conf > /usr/local/etc/supervisord.conf
```

修改最后两行

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

- 注意 **[inlcude]标签的注释** 是要被**放开**的，坑在这里很久。

启动并载入配置

```
./user/local/bin/supervisord -c /usr/local/etc/supervisord.conf
```

查看supervisor进程状态

```
ps -ef|grep supervisord
```

简单http服务

```
package main

import (
    "fmt"
    "log"
    "net/http"
)

// w表示response对象，返回给客户端的内容都在对象里处理
// r表示客户端请求对象，包含了请求头，请求参数等等
func index(w http.ResponseWriter, r *http.Request) {
    // 往w里写入内容，就会在浏览器里输出
    fmt.Fprintf(w, "Hello golang http!")
}

func main() {
    // 设置路由，如果访问/，则调用index方法
    http.HandleFunc("/", index)

    // 启动web服务，监听9090端口
    err := http.ListenAndServe(":9090", nil)
    if err != nil {
        log.Fatal("ListenAndServe: ", err)
    }
}
```

后台启动程序

```
./simpleHttp
```

在浏览器中访问 http://localhost:9090 就可以看到Hello golang http!输出了。

Supervisor监控Http服务

创建配置

```
[program:simpleHttp]
;程序启动参数，这个比较简单
command=/Users/Luciano/work/go/src/simpleHttp/simpleHttp
;是否跟随supervisord的启动而启动，我们设置了true是
autostart=true
;程序退出后自动重启，选择true是
autorestart=true
```

```
;进程被杀死时，是否向这个进程组发送stop信号，包括子进程，选择true是
stopasgroup=true
;向进程组发送kill信号，包括子进程，选择true是
killasgroup=true
;下面这几行是日志文件和日志大小和备份个数
stdout_logfile=/var/log/simpleHttp.std.log
stdout_logfile_maxbytes = 50MB
stdout_logfile_backups  = 10
stderr_logfile=/var/log/simpleHttp.err.log
stderr_logfile_maxbytes=50MB
stderr_logfile_backups=10
[supervisord]
```

注意: **配置被[program:xxx] ... [supervisord]包裹**

重启 Supervisor服务

./user/local/bin/supervisorctl -c /usr/local/etc/supervisord.conf restart

验证管理效果

- ps -ef|grep simpleHttp
- kill pid
- ps -ef|grep simpleHttp

检查pid的变化，如果变化，说明程序被自动拉起了，验证成功!

Supervisor 命令

- 启动supervisor服务

install_path/supervisord -c supervisord.conf

- 停止supervisor服务

supervisorctl [-c supervisord.conf] shutdown

- 查看supervisor管理的program列表及状态

supervisorctl [-c supervisord.conf] status

- 管理单个或多个进程

supervisorctl [-c supervisord.conf] [start|status|stop|restart] program_name_1 [program_name_n...]

- 管理进程组

supervisorctl [-c supervisord.conf] [start|status|stop|restart] groupworker:group_name

- 启动新配置的program、重启配置有改动的program

supervisorctl [-c supervisord.conf] update

- 停止 **原所有进程**，并按新的配置启动program

supervisorctl [-c supervisord.conf] reload

- 停止所有program

supervisorctl [-c supervisord.conf] stop all

- 进入supervisorctl交互式环境

supervisorctl [-c supervisord.conf]