# 热启动

一般有k8s的那套, 不需要热重启的, 因为他会先断流量 再重启 在加入到服务组.

本质上是需要fork一个子进程出来,子进程出来后,子进程拿到父进程的listenr即可.

注意你的一个listene其实是一个server socket, 那么一个socket肯定是会对应linux系统里面的一个file描述符的, 这玩意是一个整数.其实就是子进程拿到这个file描述符,然后通过net.listernter(file) 这个函数,就可以通过这个描述符,获取到listener这个网络的东西就可以用了.

l, \_ := net.Listen("tcp", "127.0.0.1:8081")  
tcpln := l.(\*net.TCPListener)  
f, err := tcpln.File()  
if err != nil {  
 panic(err)  
}  
fmt.Println(fmt.Sprintf("%d", f.Fd()))

这样就可以获取一个listener的文件描述符了.

注意net.listener的参数是一个地址, 而net.FileListener的参数是一个file描述符.

package main

import (

"fmt"

"io/ioutil"

"log"

"net"

"os"

"strconv"

"sync"

"syscall"

"time"

)

const envRestart = "RESTART"

const envListenFD = "LISTENFD"

func main() {

v := os.Getenv(envRestart)

if v != "1" {

ln, err := net.Listen("tcp", "localhost:8888")

if err != nil {

panic(err)

}

wg := sync.WaitGroup{}

wg.Add(1)

go func() {

defer wg.Done()

for {

ln.Accept()

}

}()

tcpln := ln.(\*net.TCPListener)

f, err := tcpln.File()

if err != nil {

panic(err)

}

os.Setenv(envRestart, "1")

os.Setenv(envListenFD, fmt.Sprintf("%d", f.Fd()))

\_, err = syscall.ForkExec(os.Args[0], os.Args, &syscall.ProcAttr{

Env: os.Environ(),

Files: []uintptr{os.Stdin.Fd(), os.Stdout.Fd(), os.Stderr.Fd(), f.Fd()},

Sys: nil,

})

if err != nil {

panic(err)

}

log.Print("parent pid:", os.Getpid(), ", pass fd:", f.Fd())

f.Close()

wg.Wait()

} else {

v := os.Getenv(envListenFD)

fd, err := strconv.ParseInt(v, 10, 64)

if err != nil {

panic(err)

}

log.Print("child pid:", os.Getpid(), ", recv fd:", fd)

// case1: 理解上面提及的file descriptor、file description的关系

// 这里子进程继承了父进程中传递过来的一些fd，但是fd数值与父进程中可能是不同的

//

// 取消注释来测试...

//ff := os.NewFile(uintptr(fd), "")

//if ff != nil {

// \_, err := ff.Stat()

// if err != nil {

// log.Println(err)

// }

//}

// case2: 假定父进程中共享了fd 0\1\2\listenfd给子进程，那再子进程中可以预测到listenfd=3

ff := os.NewFile(uintptr(3), "")

fmt.Println("fd:", ff.Fd())

if ff != nil {

\_, err := ff.Stat()

if err != nil {

panic(err)

}

// 这里pause, 运行命令lsof -P -p $pid，检查下有没有listenfd传过来，除了0，1，2，应该有看到3

// ctrl+d to continue

ioutil.ReadAll(os.Stdin)

fmt.Println("....")

\_, err = net.FileListener(ff)

if err != nil {

panic(err)

}

// 这里pause, 运行命令lsof -P -p $pid, 会发现有两个listenfd,

// 因为前面调用了ff.FD() dup2了一个，如果这里不显示关闭，listener将无法关闭

ff.Close()

time.Sleep(time.Minute)

}

time.Sleep(time.Minute)

}

}