

健康检查脚本类型配置帮助

脚本使用基本规则

提供 3 种基本数据类型

- int, string, bool
- 定义一个 int 变量, 并赋值
 - var code int
 - code = 10
- 定义一个 string 变量
 - var helloworld string
 - helloworld = "helloworld"
- 定义一个 bool 变量
 - var isTrue bool
 - isTrue = true

提供以下函数使用

- httpReq
 - 用途: 请求一个 url
 - 参数
 - url -- string 请求 url
 - method -- string 请求方法
 - body -- string 请求体(没有请求体时可以传入一个空字符串 "")
 - 返回值
 - int 函数结果状态码
 - int http 请求响应状态码
- tcpConnect
 - 用途: 进行一次 tcp 连接
 - 参数
 - ip -- string ip 地址
 - port -- int 端口号
 - 返回值
 - int 函数结果状态码
- icmpConnect
 - 用途: ping 一个 ip
 - 参数
 - ip -- string ip 地址
 - 返回值
 - int 函数结果状态码
- getNodeStatus
 - 用途: 获取一个服务器池中的节点的状态
 - 参数
 - poolId -- int 服务器池 id

- nodeIp -- string	服务器池节点 ip
- nodePort -- int	服务器池节点 port
- 返回值	
- int	函数结果状态码
- string	node 健康状态

函数结果状态码表

状态码	说明
0	函数执行成功
1001	创建 http 请求失败, 请检查参数
1002	执行 http 请求失败, 请检查参数
2001	Tcp 连接失败
3001	创建 ICMP 请求失败, 请检查参数
3002	执行 ICMP 请求失败, 请确定地址存在

脚本全局变量

脚本存在 2 个全局变量, 便于用户获取当前被检查的服务器节点的信息, 用户可以在脚本的任意地方使用

变量名	变量类型	说明
curIp	string	当前被检查的节点的 ip 地址
curPort	int	当前被检查的节点的端口号

脚本返回结果

- 脚本使用 `return` 关键字进行结束, 并且需要一个返回结果, 这个结果是一个 `bool` 类型的变量, 返回 `true` 代表脚本检查成功, 返回 `false` 代表脚本检查失败。

简单脚本示例

场景 1: 对当前服务器节点的 `url` 发起一次 `http` 调用

```
var codeHttp int
var statusCode int
var url string

// 拼接 url

url = "http://" + curIp + ":" + intToString(curPort) + "/"

// 对指定 url 发送 get 请求

codeHttp, statusCode = httpReq(url, "GET", "")
if codeHttp == 0 && statusCode == 200 {
    return true
}
```

```
}

return false
```

场景 2: 对当前的服务器节点地址发起一次 `tcp` 调用

```
var codeTcp int
codeTcp = tcpConnect(curIp, curPort)
if codeTcp == 0 {
    return true
}
return false
```

场景 3: 对当前的服务器节点 `ip` 发起一次 `icmp` 调用

```
var codeIcmp int

codeIcmp = icmpConnect(curIp)
if codeIcmp == 0 {
    return true
}
return false
```

场景 4: 服务器节点 `127.0.0.1: 80`, 根据【服务器池: 容量池 `10.133.7.250`】中的节点 `10.133.7.250:12000` 的状态来判断自己的状态; 服务器节点 `10.1.1.1:80` 根据【服务器池: 容量池 `10.138.5.250`】中的节点 `10.138.5.250:12000` 的状态来判断自己的状态, 其他节点均根据 `icmp` 结果来判断健康检测的结果。

```
var code int
var status string

if curIp == "127.0.0.1" && curPort == 80 {
    code, status = getNodeStatus("容量池 10.133.7.250", "10.133.7.250", 12000)
    if code == 0 && status == "health" {
        return true
    }
    return false
} else if curIp == "10.1.1.1" && curPort == 80 {
    code, status = getNodeStatus("容量池 10.138.5.250", "10.138.5.250", 12000)
    if code == 0 && status == "health" {
        return true
    }
    return false
} else {
```

```
code = icmpConnect(curIp)
if code == 0 {
    return true
}
return false
}
```