

采样Keepalive脚本打开和关闭的一个在用户指定的端口上插槽

目录

[简介](#)

[开始使用前](#)

[规则](#)

[先决条件](#)

[使用的组件](#)

[示例脚本](#)

[相关信息](#)

简介

本文档说明了脚本 keepalives 的实施。此脚本将在用户指定的端口上打开和关闭插槽。关闭的将是 FIN 而不是 RST。如果其中一个端口失败，则会将服务声明为 down。此脚本方法是对功能最密切相关的，存在于远程接入服务器(RAS)拨号客户端、终端程序和通用脚本实用程序。此功能使用 WebNS的丰富的脚本语言。

完成与一简单的插槽Application Program Interface (API) (connect或断开或送信/收信)，脚本 Keepalive将产生用户能力剪裁他们自己的协议，或者请写步骤他们自己的顺序提供服务的一可靠的 ALIVE或DOWN状态。没有脚本Keepalive功能，您当前被限制到FTP、HTTP、ICMP和TCP。然而，有了脚本Keepalive，您能通过写您自己的脚本保持在当前协议顶部。例如，您能开发特别的脚本，以连接到POP3服务器，无要求WebNS建立保活类型的POP3。此功能允许客户创建他们自己自定义的Keepalive匹配他们的特定需求。虽然这是内容服务交换机(CSS)的一部分，但是 Cisco技术支持中心不支持自定义脚本(Cisco TAC)。

TAC不正式支持下面的脚本Keepalive，然而已被测试，并且是可用的，请您谨慎使用。

[开始使用前](#)

[规则](#)

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

[先决条件](#)

本文档没有任何特定的前提条件。

[使用的组件](#)

本文档中的信息基于以下软件和硬件版本。

- WebNS版本3.x和更高
- Cisco CSS 11000系列

本文档中的信息都是基于特定实验室环境中的设备创建的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您是在真实网络上操作，请确保您在使用任何命令前已经了解其潜在影响。

[示例脚本](#)

下面脚本可以用于打开和结束在用户指定的端口的一socket。

```
!--- No echo. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !--- Filename:
ap-kal-tcp-ports !--- Parameters: Service Address, TCP Port(s) ! !--- Description: !--- This
script will open and close a socket on the user specified ports. !--- The close will be a FIN
rather than a RST. If one of the ports fails, !--- the service will be declared down ! !---
Failure Upon: !--- 1. Not establishing a connection with the host on one of the specified ports.
! !--- Note: Does not use output. !--- Will handle out of sockets scenario. ! !--- Tested: KGS
12/18/01 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! set OUT-OF-SOCKETS
"785" set NO-CONNECT "774" !--- Make sure the user has a qualified number of arguments. if
${ARGS}[#] "LT" "2" echo "Usage: ap-kal-tcp-ports \'ipAddress tcpPort1 [tcpPort2 tcpPort3...]\'"
exit script 1 endbranch set SERVICE "${ARGS}[1]" !--- echo "SERVICE = ${ARGS}[1]" var-shift ARGS
while ${ARGS}[#] "GT" "0" set TCP-PORT "${ARGS}[1]" var-shift ARGS function SOCKET_CONNECT call
!--- If out of sockets, exit, and look for sockets on the next KAL interval. if RETURN "=="
"${OUT-OF-SOCKETS}" set EXIT_MSG "Exceeded number of available sockets, skipping until next
interval." exit script 0 endbranch !--- Valid connection, look to see if it was good. if RETURN
"==" "${NO-CONNECT}" set EXIT_MSG "Connect: Failed to connect to ${SERVICE}:${TCP-PORT}" exit
script 1 endbranch endbranch no set EXIT_MSG exit script 0 function SOCKET_CONNECT begin set
CONTINUE_ON_ERROR "1" socket connect host ${SERVICE} port ${TCP-PORT} tcp 2000 set SOCKET-STAT
"${STATUS}" set CONTINUE_ON_ERROR "0" socket disconnect ${SOCKET} graceful function
SOCKET_CONNECT return "${SOCKET-STAT}" function SOCKET_CONNECT end
```

[相关信息](#)

- [CSS 11000系列内容服务交换机产品支持](#)
- [CSS 11500系列内容服务交换机产品支持](#)
- [下载CSS 11000软件\(仅限注册用户\)](#)
- [下载CSS11500软件\(仅限注册用户\)](#)
- [技术支持和文档 - Cisco Systems](#)