

Script del keepalive de la muestra para hacer ping los serverAddress

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Secuencia de comandos de ejemplo](#)

[Información Relacionada](#)

[Introducción](#)

Este script hace ping los serverAddress. Si no responde el direccionamiento, el script sale y señala el servicio por medio de una bandera como la muerte/abajo. Si el servidor está para arriba pero todos los puertos están abajo, el servidor se marca como muerte/abajo. Cada uno de los puertos virtuales del servidor se prueba. Si ningunos fallan, se registra un mensaje, pero el servicio no se marca como abajo. Este documento también dirige la implementación de los keepalives en secuencia de comandos. Este método de scripting está el más estrechamente vinculado a las funciones, que está presente en los clientes de marcación manual del Remote Access Server (RAS), los programas para terminal, y las utilidades generales del scripting. Esta característica utiliza el lenguaje de la secuenciación de comandos de los ricos de WebNS.

Complete con un Application Program Interface del socket simple (API) (conecte/desconexión/envían/reciben), un keepalive en secuencia de comandos da a usuario la capacidad de adaptar su propio protocolo, o escribió su propia la secuencia de pasos para proporcionar un `vivo` o a un `estado inactivo` confiable de un servicio. Sin las funciones del keepalive en secuencia de comandos, le limitan actualmente al FTP, al HTTP, al ICMP, y al TCP. Sin embargo, con los keepalives en secuencia de comandos usted puede permanecer encima de los protocolos actuales cuando usted escribe sus propios scripts. Por ejemplo, usted puede desarrollar un script entonado específicamente para conectar con un servidor POP3 sin requerir WebNS construir un tipo de keepalive POP3. Esta característica permite que los clientes creen su propio Keepalives de encargo para adaptarse a sus requisitos específicos. Aunque éste sea un componente del Content Services Switch (CSS), las secuencias de comandos personalizadas no son soportadas por el Soporte técnico de Cisco.

Los keepalives en secuencia de comandos en este documento no son soportados oficialmente por el Soporte técnico de Cisco, sino se han probado, y están disponibles para el uso en su propia discreción.

[prerrequisitos](#)

Requisitos

Familiaridad con el lenguaje de la secuenciación de comandos de los ricos de WebNS.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- WebNS versión 3.x y posteriores
- 11x00 Series CSS

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Convenciones

Consulte [Convenciones de Consejos TécnicosCisco](#) para obtener más información sobre las convenciones del documento.

Secuencia de comandos de ejemplo

Este script se puede utilizar para hacer ping los serverAddress.

```
!--- No echo. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !--- Filename:
ap-kal-virtual !--- Parameters: serverAddress serverPort1 [serverPort2...] ! !--- Description:
!--- This script pings the serverAddress. If the address !--- does not respond, the script exits
and flags the !--- service as dying/down. !--- If the server is up but all the ports are down,
the !--- server is marked as dying/down. !--- Each of the server's virtual ports are tested. If
any !--- fail, a message is logged, but the service is not !--- marked as down. ! !--- 1.0 -
03/20/02 KGS: Initial version (Tested) !--- 1.1 - 03/27/02 KGS: Added check for out of sockets
condition !--- Added die on all ports down (tested). !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !--- Make sure there are enough
arguments. if ${ARGS}[#] "LT" "2" echo "Usage: ap-kal-virtual \'servAddress1 servPort1
[servPort2 servPort3...]\'" exit script 1 endbranch !--- Defines. set NO_SOCKETS "785" set
NO_CONNECT "774" !--- Set up variables. set serverAddress "${ARGS}[1]" var-shift ARGS set
portList "${ARGS}" !--- Test the server. set EXIT_MSG "Ping failure of ${serverAddress}" ping
${serverAddress} no set EXIT_MSG set anyPortUp "0" while ${portList}[#] "GT" "0" !--- Test the
ports that the virtual servers are on. set CONTINUE_ON_ERROR "1" socket connect host
${serverAddress} port ${portList}[1] tcp set Result "${STATUS}" socket disconnect ${SOCKET} set
CONTINUE_ON_ERROR "0" !--- If out of sockets, warn the operator. if Result "==" "${NO_SOCKETS}"
cliLogMessage subsystem netman "KAL Scripts: Out of sockets. Try increasing KAL frequency."
level warning-4 endbranch !--- If a port does not open, log it. if Result "==" "${NO_CONNECT}"
cliLogMessage subsystem netman "Virtual server on ${serverAddress}:${portList}[1] did not
respond." level critical-2 endbranch !--- Check to make sure at least one port is up. if Result
"==" "0" set anyPortUp "1" endbranch var-shift portList endbranch !--- If no ports are up, the
service should die. if anyPortUp "==" "0" set EXIT_MSG "Virtual server on ${serverAddress} was
up, but no ports were alive." exit script 1 endbranch exit script 0
```

Información Relacionada

- [Soporte del hardware de los CSS 11000 Series Content Services Switch](#)

- [Soporte del hardware de los CSS 11500 Series Content Services Switch](#)
- [Descarga del software para el CSS11500](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)