

# Muestree el script del keepalive para marcar una lista de interfaces que los usuarios pasen encendido la línea de comando

## Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Secuencia de comandos de ejemplo](#)

[Información Relacionada](#)

## [Introducción](#)

Este script se diseña para marcar una lista de interfaces que el usuario pase encendido la línea de comando. Si de estos links falla, el servicio será declarado abajo. Esto se puede utilizar como servicio crítico con la Redundancia VIP/interface para proporcionar la capacidad de monitoreo del vínculo físico. 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 rico de WebNS.

Complete con un Application Program Interface del socket simple (API) (conecte/desconexión/envían/reciben), un dará al usuario del keepalive en secuencia de comandos la capacidad de adaptar su propio protocolo, o escribió su propia la secuencia de para proporcionar un estado ALIVE o DOWN confiable de algún servicio de los pasos. Sin las funciones del keepalive en secuencia de comandos, le limitan actualmente al FTP, al HTTP, al ICMP, y al TCP. Con los keepalives en secuencia de comandos, sin embargo, usted puede permanecer encima de los protocolos actuales escribiendo 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 Centro de Asistencia Técnica de Cisco (TAC de Cisco).

Los keepalives en secuencia de comandos abajo no son soportados oficialmente por TAC, 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 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.

## Secuencia de comandos de ejemplo

El script abajo se puede utilizar para marcar una lista de interfaces que los usuarios pasen encendido la línea de comando.

```
!--- No echo. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !--- Filename:
ap-kal-phy-check !--- Parameters: Phy1, Phy2, Phy3, and so on. For CSS 11x50, use e1, e2, e3, !-
-- and so on for the syntax. For CSS 11800, use slot/port 1/1, 1/2, 1/3, !--- and so on for the
syntax. !--- Description: !--- This script is designed to check a list of interfaces that the
user !--- passes on the command line. If any one of these links fails, the !--- service will be
declared down. This can be used as a critical service !--- with VIP/interface redundancy to
provide physical link monitoring capability. ! !--- Failure Upon: !--- 1. Any interface in the
down state. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! if ${ARGS} [#] "LT"
"1" echo "Usage: ap-kal-phy-check \?phy1 phy2 phy3 ...\?" exit script 1 endbranch while
${ARGS} [#] "GT" "0" set Host "${ARGS}[1]" var-shift ARGS function Phycheck call "${Host}"
endbranch no set EXIT_MSG exit script 0 function Phycheck begin !--- Check the first physical.
show phy ${ARGS}[1] | grep -u Down if STATUS "NEQ" "0" exit script 1 endbranch function Phycheck
end
```

## 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](#)