

Exemple de script Keepalive pour test ping sur serverAddress

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Exemple de script](#)

[Informations connexes](#)

[Introduction](#)

Ce script cingle les serverAddress. Si l'adresse ne répond pas, le script quitte et signale le service comme mourante/vers le bas. Si le serveur est en fonctionnement mais tous les ports sont vers le bas, le serveur est marqué en tant que mort/vers le bas. Chacun des ports virtuels du serveur est testé. Si en échouent, un message est enregistré, mais le service n'est pas marqué en tant que vers le bas. Ce document adresse également l'implémentation des keepalives à base de script. Cette méthode de script le plus étroitement est liée à la fonctionnalité, qui est présente dans des clients distants du Remote Access Server (RAS), des programmes de terminal, et des utilitaires généraux de script. Cette caractéristique utilise le langage de script de riches de WebNS.

Terminez-vous avec une interface de programmation simple de socket (API) (connectez/débranchement/send/receive), une keepalive à base de script donne à l'utilisateur la capacité de travailler leur propre protocole, ou écrit leur propre ordre des étapes afin de fournir un ACTIF ou un état d'indisponibilité fiable d'un service. Sans fonctionnalité de keepalive à base de script, vous êtes actuellement limité au FTP, au HTTP, à l'ICMP, et au TCP. Cependant, avec des keepalives à base de script vous pouvez rester sur les protocoles en cours quand vous écrivez vos propres scripts. Par exemple, vous pouvez développer un script spécifiquement modifié la tonalité pour se connecter à un serveur POP3 sans exiger de WebNS d'établir un type de keepalive POP3. Cette caractéristique permet à des clients pour créer leur propre Keepalives fait sur commande afin de satisfaire à leurs exigences spécifiques. Bien que ce soit un composant du Commutateur de services de contenu (CSS), des scripts personnalisés ne sont pas pris en charge par le support technique de Cisco.

Les keepalives à base de script dans ce document ne sont pas officiellement prises en charge par le support technique de Cisco, mais ont été testées, et sont disponibles pour l'usage à votre propre discrétion.

[Conditions préalables](#)

[Conditions requises](#)

Connaissance de langage de script de riches de WebNS.

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Versions 3.x et ultérieures de WebNS
- Gamme 11x00 CSS

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

Exemple de script

Ce script peut être utilisé pour cingler les 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
```

Informations connexes

- [Support matériel de Commutateurs de services satisfaits de gamme 11000 CSS](#)
- [Support matériel pour les commutateurs de services de contenu de la gamme CSS 11500](#)
- [Téléchargement logiciel pour CSS11500](#)

- [Support et documentation techniques - Cisco Systems](#)