

Script do Keepalive da amostra para sibilar serverAddress

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Exemplo de script](#)

[Informações Relacionadas](#)

[Introdução](#)

Este script sibila os serverAddress. Se o endereço não responde, o script retira e embandeira o serviço como a morte/para baixo. Se o server está acima mas todas as portas estão para baixo, o server está marcado como a morte/para baixo. Cada um das portas virtuais do server é testada. Se alguns falham, uma mensagem está registrada, mas o serviço não é marcado como para baixo. Este documento igualmente endereça a aplicação dos keepalive de script. Este método do script é o mais estreitamente relacionado à funcionalidade, que esta presente em clientes dialup do servidor de acesso remoto (RAS), em programas terminal, e em utilidades gerais do script. Esta característica utiliza o linguagem de script dos ricos de WebNS.

Termine com um Application Program Interface do soquete simples (API) (conecte/disconexão/enviam/recebem), um keepalive de script dá ao usuário a capacidade para costurar seu próprio protocolo, ou escrevem sua própria sequência das etapas a fim fornecer um `VIVO` ou um `estado inativo` seguro de um serviço. Sem a funcionalidade do keepalive de script, você é limitado atualmente ao FTP, ao HTTP, ao ICMP, e ao TCP. Contudo, com keepalive de script você pode permanecer sobre os protocolos atual quando você escreve seus próprios scripts. Por exemplo, você pode desenvolver um script tonificado especificamente para conectar a um server POP3 sem exigir WebNS construir um tipo keepalive POP3. Esta característica permite que os clientes criem seu próprio Keepalives feito sob encomenda a fim servir suas exigências específicas. Embora este seja um componente do Content Services Switch (CSS), os scripts personalizados não são apoiados pelo Suporte técnico de Cisco.

Os keepalive de script neste documento não são apoiados oficialmente pelo Suporte técnico de Cisco, mas foram testados, e estão disponíveis para o uso em sua própria discreção.

[Pré-requisitos](#)

[Requisitos](#)

Familiaridade com o linguagem de script dos ricos de WebNS.

[Componentes Utilizados](#)

As informações neste documento são baseadas nestas versões de software e hardware:

- WebNS versões 3.x e superiores
- 11x00 Series CSS

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

[Convenções](#)

Consulte as [Convenções de Dicas Técnicas da Cisco](#) para obter mais informações sobre convenções de documentos.

[Exemplo de script](#)

Este script pode ser usado para sibilhar os 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 ${ARGS}[#] "LT" "2" echo "Usage: ap-kal-virtual \serverAddress1 servPort1
[servPort2 servPort3...]" exit script 1 endbranch !--- Defines. set NO_SOCKETS "785" set
NO_CONNECT "774" !--- Set up variables. set serverAddress "${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
```

[Informações Relacionadas](#)

- [Suporte a hardware dos CSS 11000 Series Content Services Switch](#)
- [Suporte a hardware dos CSS 11500 Series Content Services Switch](#)
- [Download do software para o CSS11500](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)