

CSS HTTP Keepalives

Índice

[Introdução](#)

[Antes de Começar](#)

[Convenções](#)

[Pré-requisitos](#)

[Componentes Utilizados](#)

[Configurar](#)

[Diagrama de Rede](#)

[Configuração](#)

[Troubleshooting](#)

[Configurando o método de manutenção de atividade HTTP](#)

[Diretrizes de uso](#)

[Informações Relacionadas](#)

[Introdução](#)

O interruptor do Cisco Content Services (CSS) vota a atividade de servidores usando protocolos diferentes para verificar a integridade dos server nas camadas 3 ao 5. A votação pode ser feita usando o ICMP, o HTTP (CABEÇA, soma de verificação da página, URL), o TCP, e o FTP. À exceção do Domain Name System (DNS), porque o CSS é aplicativo DNS ciente, Cisco não apoia usando o Keepalives do User Datagram Protocol (UDP). Usando HTTP, o poll pode verificar se uma página da Web está sendo atendida e se ela foi modificada. A votação pode ser executada a níveis do serviço individual ou ser amarrada ao serviço relacionado.

[Antes de Começar](#)

[Convenções](#)

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

[Pré-requisitos](#)

Não existem requisitos específicos para este documento.

[Componentes Utilizados](#)

A informação neste documento é baseada em Content Services Switch do Cisco CSS 11000 e do 11500 Series e em todas as versões webns de Cisco.

Configurar

O cliente tem três server que distribuem o índice básico. O segundo server igualmente segura transações de base de dados, e o terceiro server segura o índice específico. Se o banco de dados de backend tornar-se inativo, o segundo servidor deverá ser colocado fora de serviço; igualmente, se o server satisfeito vai para baixo, deve ser tomado fora de serviço também.

Nota: Para localizar informações adicionais sobre os comandos usados neste documento, utilize a Ferramenta Command Lookup (somente clientes [registrados](#)).

Diagrama de Rede

Configuração

Sydney CSS

```
!Generated on 05/01/2004 02:57:42
!Active version: sg0710102

configure

!***** GLOBAL
*****
ip route 0.0.0.0 0.0.0.0 209.165.202.130 1

!***** INTERFACE
*****
interface 1/2
bridge vlan 2

interface 2/2
bridge vlan 2

!***** CIRCUIT
*****
circuit VLAN1

ip address 209.165.202.128 255.255.255.224

circuit VLAN2

ip address 192.168.5.1 255.255.255.0

!***** KEEPALIVE
*****
keepalive ContentServer
!--- Poll the database server using HTTP. Use the
default HEAD method, which looks for !--- a 200 OK
response from the real server. type http uri
"/content.html" ip address 192.168.5.15 active keepalive
DatabaseServer type http !--- Use a GET and compare the
checksum of the login page. If there is a problem, it is
!--- displayed on the login page. method get ip address
192.168.5.10 uri "/dblogin.html" active
!***** SERVICE
*****
!--- Service rule for the
real servers. The CSS polls the server using the
```

```

specified named !--- keepalive. service ContentServer ip
address 192.168.5.15 keepalive type named ContentServer
active service DatabaseServer ip address 192.168.5.10
keepalive type named DatabaseServer active !--- Service
rule for the real servers. The CSS polls the server
using the default HTTP !--- keepalive type. service
IISWEB ip address 192.168.5.2 keepalive type http active
|***** OWNER
***** owner Website !--- Specify a
Layer 7 content rule. The CSS matches the most specific
rule first. content CONTENT protocol tcp vip address
209.165.202.129 port 80 url "/content.html" add service
ContentServer active content DATABASE protocol tcp vip
address 209.165.202.129 port 80 url "/dblogin.html" add
service DatabaseServer active content WWW protocol tcp
add service IISWEB vip address 209.165.202.129 port 80
active !--- Specify a group only if you have real
servers with private IP addresses; this will NAT !---
the packet on return to the client.
|***** GROUP
***** group WWW vip address
209.165.202.129 add destination service IISWEB active

```

Troubleshooting

Esta seção fornece informações que você pode usar para confirmar se sua configuração está funcionando adequadamente.

O comando **keepalive sh** fornece um sumário completo de todo o Keepalives no CSS. Se você vê o estado: **Vivo**, você sabe que esse serviço esteve configurado, e o CSS pode ver o dispositivo. Este comando igualmente fornece a informação em que tipo de keepalive é e os específicos no que fará a falha do serviço. Os serviços que começam com **AUTO_** são serviços padrão gerados pelo CSS quando você adiciona um serviço; isto inclui o gateway padrão. Se um serviço é mostrado como o estado: **Para baixo**, você precisa de verificar sua conexão do CSS a seu servidor real (à revelia toma três falhas de um keepalive particular marcar um serviço como para baixo). Estado: **Morrer** significa que o CSS não teve ainda o contato com o servidor real.

```

SYDNEY# sh keepalive Keepalives: Name: AUTO_nexthop00004 Index: 0 State: Alive Description: Auto
generated for service nexthop00004 Address: 209.165.202.130 Port: Any Type: ICMP Frequency: 5
Max Failures: 3 Retry Frequency: 5 Dependent Services: nexthop00004 Name: AUTO_IISWEB Index: 1
State: Alive Description: Auto generated for service IISWEB Address: 192.168.5.2 Port: 80 Type:
HTTP:HEAD:/ Frequency: 5 Max Failures: 3 Retry Frequency: 5 Dependent Services: IISWEB Name:
DatabaseServer Index: 2 State: Alive Description: Address: 192.168.5.10 Port: 80 Type:
HTTP:GET:/dblogin.html Hash: 95bd5419e38977e967b399853729c86f Frequency: 5 Max Failures: 3 Retry
Frequency: 5 Dependent Services: DatabaseServer Name: ContentServer Index: 3 State: Alive
Description: Address: 192.168.5.15 Port: 80 Type: HTTP:HEAD:/content.html Frequency: 5 Max
Failures: 3 Retry Frequency: 5 Dependent Services: ContentServer

```

Se você tem um grande número serviços, é possível alistar somente um serviço especificado datilografando em nome do serviço que você está interessado dentro.

```

SYDNEY# sh keepalive ContentServer Name: ContentServer Index: 3 State: Alive Description:
Address: 192.168.5.15 Port: 80 Type: HTTP:HEAD:/content.html Frequency: 5 Max Failures: 3 Retry
Frequency: 5 Dependent Services: ContentServer

```

Configurando o método de manutenção de atividade HTTP

Emita o comando do **método de keepalive** especificar o método do keepalive de HTTP para um serviço. A sintaxe e as opções para este comando de modo do serviço são:

- **o método obtém** — O CSS emite um método HTTP GET ao serviço, computa um valor de hash na página, e armazena o valor de hash como uma mistura da referência. GETs subsequentes exigem um status de 200 OK (resposta OK concluída do comando HRRP) e o valor de hash para igualar o valor de hash de referência. Se o status 200 OK não for retornado ou se for retornado, mas o valor de hash for diferente do valor de hash de referência, o CSS não considerará o serviço como desativado. Quando você especifica a informação satisfeita de um identificador de recurso uniforme (URI) HTTP para um keepalive de HTTP, o CSS calcula um valor de hash para o índice. Se as informações de conteúdo forem alteradas, o valor de mistura não corresponde mais ao valor de mistura original, e o CSS supõe que o serviço está inativo. Para impedir que o CSS suponha que um serviço é abaixo de devido a uma má combinação do valor de hash, especifique o método de keepalive como a cabeça.
- **cabeça do método** (padrão) — O CSS emite um método HTTP-HEAD ao serviço e um estado de 200 APROVAÇÕES é exigido. O CSS não calcula um valor de referência de hash para esse tipo de manutenção de atividade. Se o estado de 200 APROVAÇÕES não é retornado, o CSS considera o serviço para baixo. Por exemplo, `entre:`

Se você muda o método de keepalive em um serviço ativo, certifique-se de que você suspende e reactivate o serviço para que a mudança tome o efeito.

[Diretrizes de uso](#)

Quando você especificar um identificador de recurso uniforme (?) (URI) para um keepalive de HTTP, o CSS calcula um valor de hash para o página da web especificado no URI. Se o página da web muda, o valor de hash já não combina o valor de hash original e o CSS supõe que o serviço está inoperante. Para impedir que o CSS suponha que um serviço é inoperante devido a uma má combinação do valor de hash, defina o **método de keepalive** como a **cabeça**. O CSS não computa um valor de hash para este tipo de keepalive.

Se você especificar o URI de uma página da Web com conteúdo alterável e não especificar o método de manutenção de atividade no cabeçalho, será necessário suspender e reativar o serviço toda vez que a página da Web for alterada.

[Máximos 512 Keepalives \(255 passam pelo processo de script tipos keepalive\) com versão webns 5.0 de Cisco avante](#)

Um CSS11000 ou uma configuração 11500 podem ter até 255 Keepalives de qualquer um tipo com um total de todos os tipos que são inferior ou igual a 512.

Exemplo

O cliente example.com precisa de carregar servidores de Web do equilíbrio 500.

```
250 TCP port 80 KeepAlives.  
250 HTTP KeepAlives.
```

```
-----  
Total: 500 Web Services
```

```
Content Rule Maximums: 64 Services per Content Rule
```

Content Rule Maximums: 250 Content Rules
(example with 2 Services per Content Rule).

Informações Relacionadas

- [CSS Basic Configuration Guide \(Manual de configuração básica CSS\)](#)
- [Cisco CSS 11000 Series Content Services Switches](#)
- [Comandos do modo de configuração de manutenção de atividade](#)
- [Suporte Técnico - Cisco Systems](#)