

# Señales de mantenimiento HTTP de CSS

## Contenido

[Introducción](#)

[Antes de comenzar](#)

[Convenciones](#)

[prerrequisitos](#)

[Componentes Utilizados](#)

[Configurar](#)

[Diagrama de la red](#)

[Configuración](#)

[Troubleshooting](#)

[Configuración del método Keepalive de HTTP](#)

[Pautas de uso](#)

[Información Relacionada](#)

## [Introducción](#)

El Switch del Cisco Content Services (CSS) sondea la actividad de los servidores usando diversos protocolos para marcar la integridad de los servidores en las capas 3 a 5. La interrogación se puede hacer usando el ICMP, HTTP (PISTA, suma de comprobación de la página, URL), el TCP, y el FTP. A excepción del Domain Name System (DNS), porque el CSS es aplicación DNS enterada, Cisco no soporta usando el Keepalives del User Datagram Protocol (UDP). Con HTTP, las consultas pueden verificar si una página Web está en servicio y si ha sido modificada. La interrogación se puede realizar en los niveles del servicio individual o atar al servicio relacionado.

## [Antes de comenzar](#)

### [Convenciones](#)

Para obtener más información sobre las convenciones del documento, consulte [Convenciones de Consejos Técnicos de Cisco](#).

### [prerrequisitos](#)

No hay requisitos previos específicos para este documento.

### [Componentes Utilizados](#)

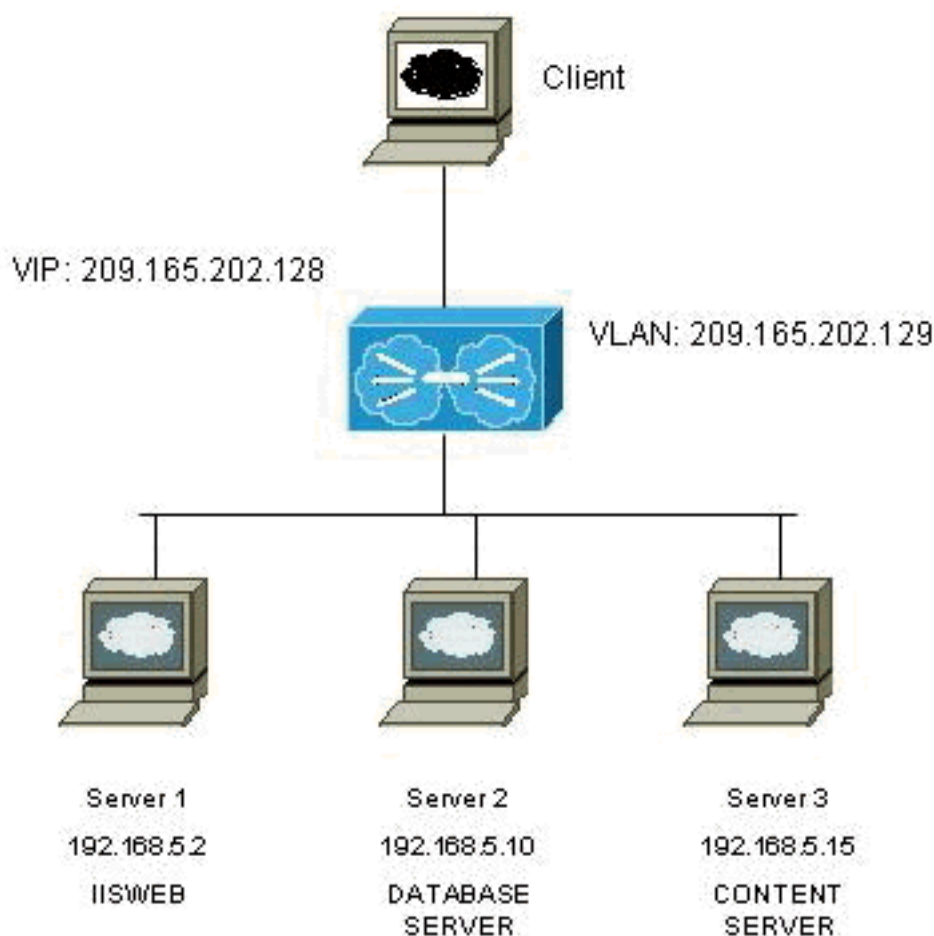
La información en este documento se basa en los Content Services Switches del Cisco CSS 11000 y de las 11500 Series y todas las WebNS versiones de Cisco.

## Configurar

El cliente tiene tres servidores que distribuyan el contenido básico. El segundo servidor también maneja las transacciones de la base de datos, y el tercer servidor maneja el contenido del específico. Si la base de datos backend (segundo plano) se cae, el segundo servidor debería hacerse cargo del servicio: asimismo, si va el servidor contenido abajo, debe ser tomado el Out Of Service también.

**Nota:** Para obtener información adicional sobre los comandos que se utilizan en este documento, use la Command Lookup Tool (solo para clientes [registrados](#)).

## Diagrama de la red



## Configuración

### 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

En esta sección encontrará información que puede utilizar para confirmar que su configuración esté funcionando correctamente.

El comando **keepalive sh** proporciona un resumen completo de todo el Keepalives en el CSS. Si usted ve el `estado: Vivo`, usted sabe que se ha configurado ese servicio, y el CSS puede ver el dispositivo. Este comando también proporciona la información sobre qué tipo de keepalive es y los específicos en lo que hará el fall del servicio. Los servicios que comienzan con `AUTO_` son servicios predeterminados generados por el CSS cuando usted agrega un servicio; esto incluye el default gateway. Si un servicio se muestra como `estado: Abajo`, usted necesita marcar su conexión del CSS a su servidor real (por abandono toma tres errores de un keepalive determinada marcar un servicio como abajo). `Estado: La muerte` significa que el CSS todavía no ha tenido contacto con el 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
```

Si usted tiene un gran número de servicios, es posible enumerar solamente un servicio especificado tecleando en nombre del servicio que usted está interesado adentro.

```
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

## Configuración del método Keepalive de HTTP

Publique el comando del **método de keepalive** de especificar el método del keepalive HTTP para un servicio. El sintaxis y las opciones para este comando mode del servicio son:

- **el método consigue** — El CSS publica un método HTTP GET al servicio, computa un valor de troceo en la página, y salva el valor de troceo como hash de la referencia. Los GET posteriores requieren un estado OK de 200 (el comando HTTP completó la respuesta de OK) y que el valor hash sea igual al valor de troceo de referencia. Si el estado 200 OK no regresa, o si el estado 200 OK regresa pero el valor de troceo es diferente al valor de troceo de la referencia, el CSS considera que el servicio está caído. Cuando usted especifica la información contenta de un Identificador de recursos uniformes (URI) HTTP para un keepalive HTTP, el CSS calcula un valor de troceo por el contenido. Si cambia la información de contenidos, el valor de troceo ya no coincide con el valor troceo original y el CSS supone que el servicio está desconectado. Para evitar que el CSS si se asume que un servicio está abajo de debido a una discordancia del valor de troceo, especifique el método de keepalive como pista.
- **pista del método** (valor por defecto) — El CSS publica un método HTTP-HEAD al servicio y se requiere un estatus de 200 AUTORIZACIONES. El CSS no calcula un valor hash de referencia para este tipo de señal de mantenimiento. Si el estatus de 200 AUTORIZACIONES no se vuelve, el CSS considera el servicio abajo. Por ejemplo, *ingrese*:

Si usted cambia el método de keepalive en un servicio activo, asegúrese que usted suspende y reactiva el servicio para que el cambio tome el efecto.

### Pautas de uso

Cuando usted especifica un Identificador de recurso uniforme (?) (URI) para un keepalive HTTP, el CSS calcula un valor de troceo para la página web especificada en URI. Si la página web cambia, el valor de troceo hace juego no más el valor hash original y el CSS asume que el servicio está muerto. Para evitar que el CSS si se asume que un servicio es sin funcionamiento debido a una discordancia del valor de troceo, defina el **método de keepalive** como **pista**. El CSS no computa un valor de troceo para este tipo de keepalive.

Si especifica el URI de una página Web con contenido cambiante y no especifica el método principal de la señal de mantenimiento, deberá suspender y reactivar el servicio cada vez que cambia la página Web.

### Máximos 512 Keepalives (255 tipos de keepalive del script) con la WebNS versión 5.0 de Cisco hacia adelante

Un CSS11000 o una configuración 11500 puede tener hasta 255 Keepalives de cualquier un tipo con un total de todos los tipos que son inferior o igual 512.

## Ejemplo:

El cliente example.com necesita cargar a los servidores Web de la balanza 500.

```
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
```

## Información Relacionada

- [Guía de configuración básica de CSS](#)
- [Switches de servicios de contenido Cisco CSS de la serie 11000](#)
- [Comandos del modo de configuración Keepalive](#)
- [Soporte Técnico - Cisco Systems](#)