

# CSS HTTP Keepalives

## Содержание

[Введение](#)

[Перед началом работы](#)

[Условные обозначения](#)

[Предварительные условия](#)

[Используемые компоненты](#)

[Настройка](#)

[Схема сети](#)

[!--- конфигурацию](#)

[Устранение неполадок](#)

[Конфигурирования метода элементов оптимизации соединений HTTP](#)

[Инструкции по использованию](#)

[Дополнительные сведения](#)

## **Введение**

Cisco Content Services Switch (CSS) опрашивает активность серверов с помощью других протоколов для проверки целостности серверов на Уровнях 3 к 5. Опрос может быть сделан с помощью ICMP, HTTP (ГОЛОВКА, Контрольная сумма Страницы, URL), TCP и FTP. За исключением Системы доменных имен (DNS), потому что CSS является знающее Приложение DNS, Cisco не поддерживает пакеты Keepalive Протокола UDP использования. Используя HTTP, с помощью опросов можно проверить, была ли веб-страница обслужена или изменена. Опрос может быть выполнен на уровнях отдельного сервиса или связан к связанному сервису.

## **Перед началом работы**

### **Условные обозначения**

[Дополнительные сведения об условных обозначениях см. в документе Технические рекомендации Cisco. Условные обозначения.](#)

### **Предварительные условия**

Для данного документа отсутствуют предварительные условия.

### **Используемые компоненты**

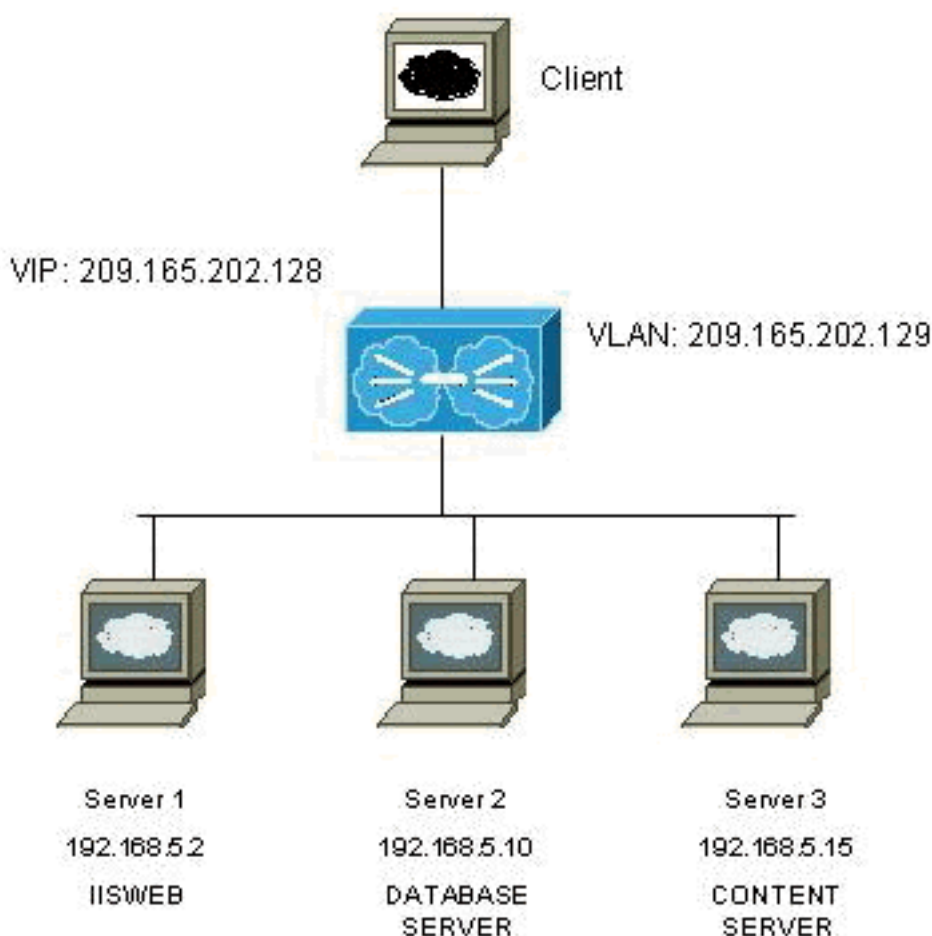
Сведения в этом документе основываются на Cisco CSS 11000 и коммутаторах контент-сервисов серии 11500 и всех Версиях WebNS Cisco.

## Настройка

У клиента есть три сервера, которые распределяют основное содержание. Второй сервер также обрабатывает транзакции базы данных, и третий сервер обрабатывает определенное содержание. Если вспомогательная база данных перестает работать, второй сервер следует отключить; аналогично, если сервер содержания выключается, он должен быть взят вне обслуживания также.

**Примечание:** [Поиск дополнительной информации о командах в данном документе можно выполнить с помощью средства "Command Lookup" \(Поиск команд\) \(только для зарегистрированных клиентов\).](#)

## Схема сети



## !--- конфигурацию

### Сиднейский 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

```

## Устранение неполадок

В этом разделе содержатся сведения, которые помогают убедиться в надлежащей работе конфигурации.

**Команда `sh keepalive`** предоставляет завершённую сводку всех пакетов Keepalive на CSS. Если вы видите `State: Alive`, вы знаете, что тот сервис был настроен, и CSS видит устройство. Эта команда также предоставляет сведения о том, какая поддержка активности это и специфические особенности того, что сделает сервисный сбой. Когда вы добавляете сервис, сервисы начиная с `AUTO_` являются сервисами по умолчанию, генерируемыми CSS; это включает шлюз по умолчанию. Если сервис показывают как `State: Down`, необходимо проверить соединение от CSS до реального сервера (по умолчанию, требуется три сбоя отдельной поддержки активности для маркировки сервиса как вниз). `State:Dying` означает, что CSS еще не имел контакта с реальным сервером.

```
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
```

Если у вас есть большое число сервисов, возможно перечислить только выбранный сервис путем ввода от имени сервиса, которым вы интересуетесь.

```
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

## Конфигурирования метода элементов оптимизации соединений HTTP

Выполните команду **метода поддержки активности** для определения метода поддержки активности HTTP для сервиса. Синтаксис и опции для этой команды сервисного режима:

- **метод добирается** — CSS выполняет метод GET HTTP к сервису, вычисляет значение хеш-функции на странице и хранит значение хеш-функции как ссылочный хэш. Последующие GET требуют статус 200 ОК (команда HTTP завершилось ответом ОК) и равенства значения хэша значению опорного хэша. Если состояние 200 ОК не возвращается или если возвращается состояние 200 ОК, но хеш-значение отличается от контрольного хеш-значения, CSS полагает, что обслуживание прекращено. При определении информации о содержании Унифицированного идентификатора ресурса (URI) HTTP для поддержки активности HTTP CSS вычисляет значение хеш-функции для содержания. Если сведения содержимого изменяются, значение хэша не совпадают с исходным и CSS считают службу отключенной. Чтобы препятствовать тому, чтобы CSS предположил, что сервис не работает из-за несоответствия значения хеш-функции, задайте метод поддержки активности как головку.
- **головка метода** (по умолчанию) — CSS выполняет метод HTTP-HEAD к сервису, и требуются 200 статусов ОК. CSS не вычисляет опорное значение хэша для данного типа поддержания активности. Если 200 статусов ОК не возвращены, CSS рассматривает сервис вниз. Например, `enter`:

При изменении метода поддержки активности на активном сервисе удостоверьтесь, что вы приостанавливаете и повторно активируете сервис для изменения для вступления в силу.

### Инструкции по использованию

При определении Идентификатора Uniform Resource(?) (URI) для поддержки активности HTTP CSS вычисляет значение хеш-функции для веб-страницы, заданной в URI. Если веб-страница изменяется, значение хеш-функции больше не совпадает с первоначальным значением хэш - функции, и CSS предполагает, что сервис мертв. Чтобы препятствовать тому, чтобы CSS предположил, что сервис является заблокированной из-за к несоответствию значения хеш-функции, определите **метод поддержки активности** как **головку**. CSS не вычисляет значение хеш-функции для этого типа поддержки активности.

Если указан URI веб-страницы с изменяемым контентом, но не указан метод кеерalive в заголовке, при каждом изменении этой веб-страницы необходимо приостанавливать и снова активировать службу.

### Максимумы 512 пакетов Кеерalive (255 типов проверки активности сценария) с Версией WebNS 5.0 Cisco и далее

CSS 11000 или 11500 конфигураций могут иметь до 255 пакетов Кеерalive любого типа с в

общей сложности всеми типами, являющимися меньше чем или равным 512.

## Пример

Клиент example.com должен сбалансировать нагрузку 500 Web-серверов.

```
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
```

## Дополнительные сведения

- [Руководство по базовой конфигурации CSS](#)
- [Cisco CSS 11000 Series Content Services Switches](#)
- [Команды активного режима конфигурации \(keepalive\)](#)
- [Техническая поддержка - Cisco Systems](#)