

Basic CSS Load–Balancing Configuration

Document ID: 12557

Introduction

Prerequisites

Requirements

Components Used

Conventions

Background Theory

Configure

Bridge VLANs to Ethernet Interfaces

Number the Circuit VLANs

Create a Service and Keepalive

Create a Content Owner

Create a Content Rule and Virtual IP Address

Add Services to a Content Rule

Sample Configuration

Verify

Basic Troubleshooting

Related Information

Introduction

This document walks you through a basic Content Services Switch (CSS) load–balancing configuration and explains the functions of each configuration element. This document does not describe in detail all of the features of this appliance. For more information on any feature reviewed in this document, refer to the Related Information section.

The example presented in this document defines one Virtual IP address (VIP) and binds three real servers, or services, behind that VIP to be load balanced.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these software and hardware versions:

- WebNS version 4.10, 5, 6, 7 build 17s
- CSS 11150

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Background Theory

When you create a load-balancing configuration, the two most important CSS-specific configuration elements to understand are content rules and services.

Content rules are where the CSS's load-balancing features are customized, virtual IP address are defined, and where the actual servers (called services) are bound to that virtual IP address. Content rules allow you to specify load-balancing types, sticky methods, port specific Virtual IP addresses, and a host of other features. Once a request hits a circuit VLAN on the CSS, that request is interrogated against all its content rules for a match. Once a match has been made between particular elements of the request and a content rule, the CSS load balances that request to all the services bound to that content rule. Each content rule must be configured under an owner.

Services are the actual individual servers that the CSS load balances. Services are configured individually and can be customized with a variety of keepalive types, Internet Control Message Protocol (ICMP) by default. A group of services can also be referred to as a 'server farm.'

Configure

In this section, you are presented with the information to configure the features described in this document.

- Bridge VLANs to Ethernet Interfaces
- Number the Circuit VLANs
- Create Services and Keepalive
- Create a Content Owner
- Create a Content Rule and Virtual IP Address
- Add Services to a Content Rule

Note: To find additional information on the commands used in this document, use the IOS Command Lookup tool.

Bridge VLANs to Ethernet Interfaces

On the CSS, it is necessary to bridge the circuit VLAN to an interface before you number that VLAN. Since VLAN1 is always the default VLAN for any interface, in this configuration it is only necessary to manually bridge VLAN10.

```
CSS150#config
CSS150(config)#interface e12
CSS150(config-if[e12])#bridge vlan 10
```

Number the Circuit VLANs

Here, you define an IP address for the VLAN interface from which the CSS will receive traffic. In this configuration, the IP address is on the same subnet as the router's LAN-side interface. CSS commands are case sensitive; circuit VLANs are referred to in all caps.

```
CSS150#config
CSS150(config)#circuit VLAN1
CSS150(config-circuit[VLAN1])#ip address 192.168.1.50 255.255.255.0
```

```

CSS150(config-circuit-ip[VLAN1-192.168.1.50])#exit
CSS150(config-circuit[VLAN1])#exit
CSS150(config)#circuit VLAN10
CSS150(config-circuit[VLAN10])#ip address 10.10.10.50 255.255.255.0
CSS150(config-circuit-ip[VLAN10-10.10.10.50])#

```

Create a Service and Keepalive

If no keepalive type is specified when you configure services, the keepalive type is set to the default of ICMP. All services must be manually activated once created.

```

CSS150(config)#service www-server1
Create service <www-server1>,[y/n]:y
CSS150(config-service[www-server1])#ip address 10.10.10.1
CSS150(config-service[www-server1])#keepalive type ?
ftp                Use FTP Keepalives for this service
http               ;   Use HTTP Keepalives for this service
icmp               ;   Use ICMP Keepalives for this service ( default )
named              &nbs p; Name of keepalive to use.
none               ;   Disable keepalive for this service
script             &nbs sp; Use SCRIPT Keepalives for this service
tcp                Use TCP Keepalives for this service
CSS150(config-service[www-server1])#keepalive type http
CSS150(config-service[www-server1])#active

```

Create a Content Owner

Owners exist for administrative flexibility. Multiple content rules can be grouped together under different owners for ease of management. All content rules must exist under an owner.

```

CSS150(config)#owner Sample
Create owner <Sample>,[y/n]:y
CSS150(config-owner[Sample])#

```

Create a Content Rule and Virtual IP Address

Cisco suggests you specify a port under the content rule for two reasons: this allows more flexibility with the Domain Name System (DNS) as several port-specific content rules can be configured with the same IP address, and permitting only the traffic required by the service provides greater security for the services behind the content rule. Two content rules can coexist with the same virtual IP address if there are other criteria for the CSS to differentiate between the two, such as TCP/User Datagram Protocol (UDP) port, URL, etc.

```

CSS150(config)#owner Sample
CSS150(config-owner[Sample])#content web-servers
Create content <web-servers>,[y/n]:y
CSS150(config-owner-content[Sample-web-servers])#
CSS150(config-owner-content[Sample-web-servers])#ip address 192.168.1.1
CSS150(config-owner-content[Sample-web-servers])#port 80

```

Add Services to a Content Rule

This is where the services are bound to the content rule, and the real servers are associated with the virtual IP address. Once services have been added, content rules must be manually activated before the services start listening for requests.

```

CSS150(config)#owner Sample
CSS150(config-owner[Sample])#content web-servers
CSS150(config-owner-content[Sample-web-servers])#add service www-server1

```

```
CSS150(config-owner-content[Sample-web-servers])#add service www-server2
CSS150(config-owner-content[Sample-web-servers])#add service www-server3
CSS150(config-owner-content[Sample-web-servers])#add service www-server4
CSS150(config-owner-content[Sample-web-servers])#active
```

Sample Configuration

```
CSS150-4#show run
```

```
!Generated on 08/28/2001 18:40:54
!Active version: ap0410017s
```

```
configure
```

```
!***** GLOBAL *****
ip route 0.0.0.0 0.0.0.0 192.168.1.100 1

!***** INTERFACE *****
interface e12
  bridge vlan 10

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

  ip address 192.168.1.50 255.255.255.0

circuit VLAN10

  ip address 10.10.10.50 255.255.255.0

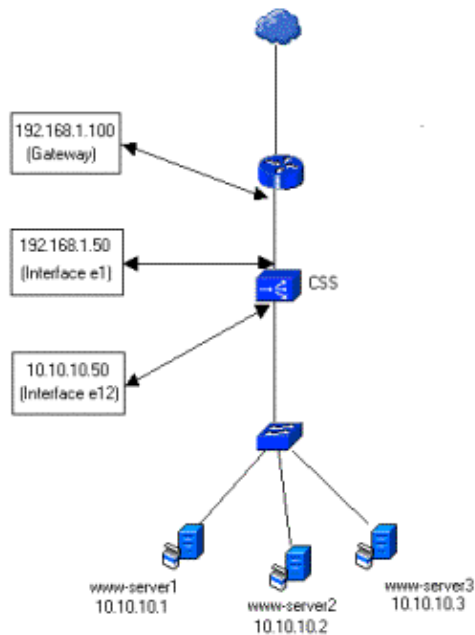
!***** SERVICE *****
service www-server1
  ip address 10.10.10.1
  keepalive type http
  active

service www-server2
  ip address 10.10.10.2
  keepalive type http
  active

service www-server3
  ip address 10.10.10.3
  keepalive type http
  active

!***** OWNER *****
owner Sample

  content web-servers
    protocol tcp
    port 80
    add service www-server1
    add service www-server2
    add service www-server3
    vip address 192.168.1.1
    active
```



Verify

There is currently no verification procedure available for this configuration.

Basic Troubleshooting

This section provides information you can use to troubleshoot your configuration.

Certain **show** commands are supported by the Output Interpreter tool, which allows you to view an analysis of **show** command output.

If traffic destined to the content rule times out, and pings from the client are successful to the circuit VLAN IP, it is a good idea to first verify the services are 'in service.' The **show service** command displays the status of the services determined by the individual service keepalives. This command helps diagnose server errors. A server only lists as being 'down' if the server has either failed to respond to its keepalive probes, or has not been manually made 'active.'

- **show summary** Displays the relationship between owners, content rules, and services.
- **show service summary** Displays summary information for all services, including the service state, connections, weight, and load.

Related Information

- [Catalyst 6500 Series SSL Services Module Installation and Verification Note](#)
- [Downloads – Catalyst 6500/6000 Module Software](#)
- [Configuring Content Rules on the CSS](#)
- [Configuring Services on the CSS](#)
- [Configuring Service Keepalives on the CSS](#)
- [Configuring Session Persistence, Remapping, and Redirection on the CSS](#)
- [CSS Basic Configuration Guide \(Software Version 7.20\)](#)
- [CSS Advanced Configuration Guide \(Software Version 7.20\)](#)
- [Technical Support & Documentation – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Jan 31, 2006

Document ID: 12557
