

# Configuration du CSS en vue de l'équilibrage de charge des serveurs DNS et de l'utilisation des Keepalives scriptés par DNS

## Contenu

[Introduction](#)

[Diagramme du réseau](#)

[Utilisant les keepalives à base de script de DN](#)

[Script d'AP-kal-dn CSS](#)

[Configuration](#)

[Construction s'exécutante 8 CSS 11150 WebNS 4.01](#)

[Sorties de la commande show](#)

[affichez la keepalive](#)

[affichez le keepalive-résumé](#)

[résumé de show service](#)

[Informations connexes](#)

## Introduction

Avec les versions de logiciel 4.0 et ultérieures de Cisco WebNS, les clients ont la capacité d'utiliser des keepalives à base de script pour des services non standard ou spécialisés, tels que le Système de noms de domaine (DNS).

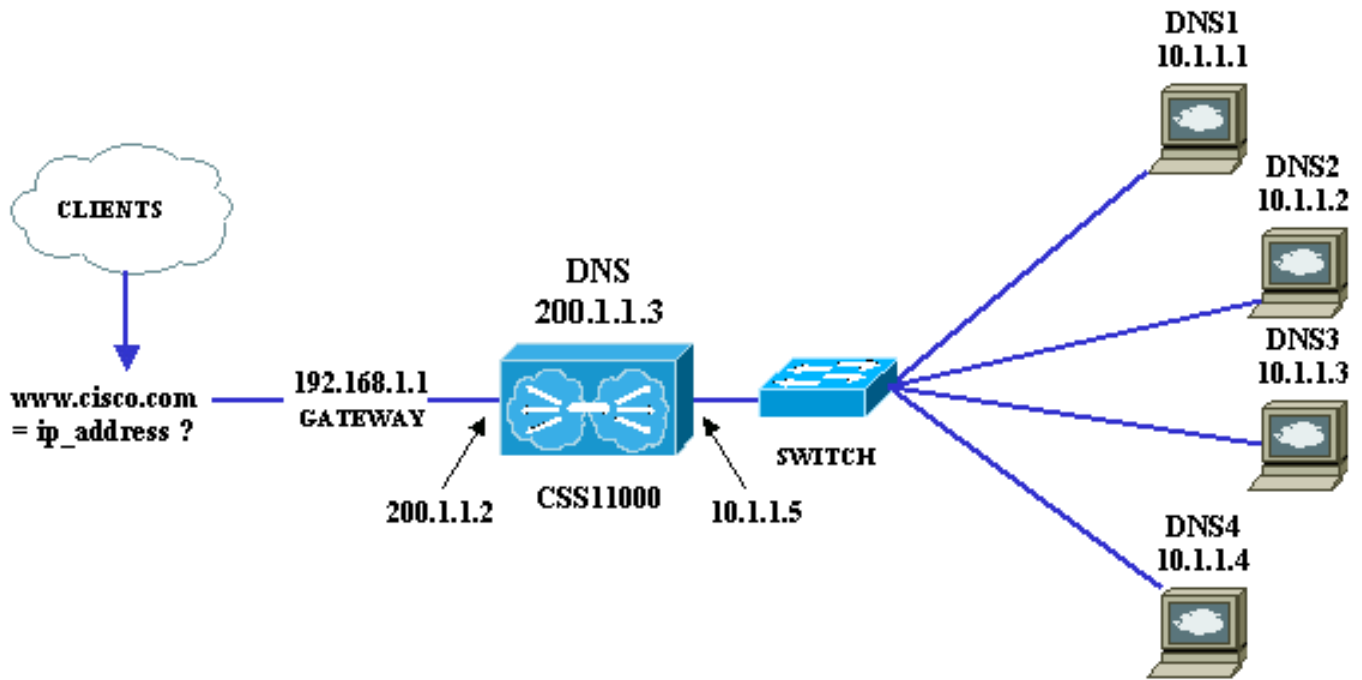
Puisque les requêtes DNS sont Protocole UDP (User Datagram Protocol) basé sur, vous devez configurer un groupe source sur le Commutateur de services de contenu (CSS) de sorte que les réponses du serveur DNS semblent provenir la même adresse dont les requêtes ont été initialement envoyées (très probablement l'adresse virtuelle IP (VIP)). Quand vous lancez ce groupe source, les keepalives à base de script de DN ne fonctionnent pas parce que la réponse à la requête de keepalive passe par le Traduction d'adresses de réseau (NAT), entraînant le CSS recevoir une réponse d'une adresse IP différente que celle à laquelle vous avez initialement envoyé la demande.

Cette configuration a été développée et testée utilisant le logiciel et les versions de matériel ci-dessous.

- Toutes les Plateformes CSS (CSS11000 et CSS11500)
- Versions de logiciel 5.0 et ultérieures de Cisco WebNS (des keepalives à base de script n'ont pas été ajoutées jusqu'à la version 4.0)

Les informations contenues dans ce document ont été créées à partir des périphériques dans un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si vous fonctionnez dans un réseau vivant, assurez-vous que vous comprenez l'impact potentiel de n'importe quelle commande en fonction avant de l'utiliser.

## Diagramme du réseau



## Utilisant les keepalives à base de script de DN

Conseils pour l'usage des keepalives à base de script de DN :

- Access aux serveurs racine de système de noms de domaine d'Internet est exigé pour l'implémentation réussie des keepalives à base de script de DN.
- L'adresse IP de service a non portant sur le service DNS du tout. N'importe quelle adresse peut être introduite et elle n'effectue pas l'état, cependant, une adresse doit être introduite ou une erreur de « mauvaise adresse IP » apparaît.
- L'argument pour le script doit être l'adresse IP ou l'adresse Internet du serveur DNS que vous voulez vérifier. C'est typiquement l'adresse IP configurée au service.
- Le script est codé en dur de résoudre [www.cisco.com](http://www.cisco.com). Il n'importe pas si le serveur DNS peut résoudre cette adresse ou pas, tant que une réponse de DN revient que le service est actif. Cette keepalive teste seulement si un serveur DNS peut répondre à une requête, pas si elle peut résoudre un nom spécifique. Les requêtes de script du CSS aux serveurs internes de DN maintiennent la Disponibilité des serveurs DNS.

## Script d'AP-kal-dn CSS

```
!no echo
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Filename: ap-kal-dns
! Parameters: DNS_Server
!
! Description: !--- This script resolves a domain name from a specific DNS
!--- server. This builds a UDP packet based on RFC 1035 . !
! Failure Upon: !--- Not resolving the hosts's IP from the domain name.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! if ${ARGS}[#] "NEQ" "1" echo
"Usage: ap-kal-dns \'Hostname\'" exit script 1 endbranch ! Defines: set HostName "${ARGS}[1]" !
Connect to the remote host set EXIT_MSG "Connection failed" socket connect host ${HostName} port
53 udp !--- This may require a little explanation. Since we just want to see
```

```

!--- if the DNS server is alive, we send a simple DNS Query. This
!--- query is hard-coded in hexadecimal and sent raw to the DNS server.
!--- The DNS request has a 12-byte header (as seen for the first 12 bytes
!--- of hex) and then a DNS name (for example, www.cisco.com).
!--- Lastly, it follows with some null termination and a few bytes
!--- representing query type. !--- See RFC 1035 for more information. set EXIT_MSG "Send:
failure" socket send ${SOCKET} "000201000001000000000000003777777057961686f6f03636f6d00000 10001"
raw !--- Receive an unexplained response, but it is not important because
!--- an unstable DNS server or a non-existent one would probably not send
!--- back any data at all. set EXIT_MSG "Receive: Failed to receive data" socket waitfor
${SOCKET} "cisco" 4000 no set EXIT_MSG socket disconnect ${SOCKET} exit script 0

```

Si les serveurs DNS ne peuvent pas répondre à une requête, alors vous devez créer un ACL avec cette logique :

- Quelque chose originaire d'un des serveurs DNS destinés pour le CSS adresse IP virtuelle de RÉSEAU LOCAL de circuit ((VLAN)) saute tous les règles de contenu et groupes sources.
- Autre trafic originaire des serveurs DNS passe par le groupe source configuré.

## Configuration

### Construction s'exécutante 8 CSS 11150 WebNS 4.01

```

!***** GLOBAL *****

ip redundancy
no restrict xml
username predictive des-password xecchdhdhnglhueig5csfbe4fievhjg
username admin des-password uezfqq6eoeic3e2d superuser
acl enable
ip route 0.0.0.0 0.0.0.0 200.1.1.1 1

!***** INTERFACE *****

interface ethernet-1
bridge vlan 2
phy 100Mbps-FD
interface ethernet-2
bridge vlan 3
phy 100Mbps-FD

!***** CIRCUIT *****

circuit VLAN2
redundancy
ip address 10.1.1.5 255.255.255.0
circuit VLAN3
redundancy
ip address 200.1.1.2 255.255.255.0

!***** SERVICE *****

service DNS1
ip address 10.1.1.1
keepalive type script ap-kal-dns "10.1.1.1"
active
service DNS2
ip address 10.1.1.2
keepalive type script ap-kal-dns "10.1.1.2"
active

```

```

service DNS3
ip address 10.1.1.3
keepalive type script ap-kal-dns "10.1.1.3"
active
service DNS4
ip address 10.1.1.4
keepalive type script ap-kal-dns "10.1.1.4"
active
service Router1
ip address 200.1.1.1
type redundancy-up
active

!***** OWNER *****

owner L3_Owner
content L3_Rule
vip address 200.1.1.3
add service DNS1
add service DNS2
add service DNS3
add service DNS4
active

!***** GROUP *****

group dns
vip address 200.1.1.3
active

!***** ACL *****

acl 20
clause 10 permit any any destination any
apply circuit-(VLAN3)
acl 10
clause 10 bypass any 10.1.1.1 255.255.255.255 destination 10.1.1.5 255.255.255.255
clause 20 bypass any 10.1.1.2 255.255.255.255 destination 10.1.1.5 255.255.255.255
clause 30 bypass any 10.1.1.3 255.255.255.255 destination 10.1.1.5 255.255.255.255
clause 40 bypass any 10.1.1.4 255.255.255.255 destination 10.1.1 5 255.255.255.255
clause 50 permit any 10.1.1.0 255.255.255.0 destination any sourcegroup dns
clause 60 permit any any destination any
apply circuit-(VLAN2)

```

## Sorties de la commande show

Échantillons de sortie de commande show :

### **affichez la keepalive**

```

52-css150-4# show keepalive Keepalives: Name: AUTO_nexthop00002 Index: 0 State: Alive
Description: Auto generated for service nexthop00002 Address: 200.1.1.1 Port: Any Type: ICMP
Frequency: 5 Max Failures: 3 Retry Frequency: 5 Dependent Services: nexthop00002 Name: AUTO_DNS1
Index: 1 State: Down Description: Auto generated for service DNS1 Address: 10.1.1.1 Port: Any
Type: SCRIPT ap-kal-dns Script Arguments: "10.1.1.1" Script Error: Script error in line: 41 !---
Note: This service has no access to root servers, which causes the error.
Script Run Time: 4 seconds Frequency: 5 Max Failures: 3 Retry Frequency: 5 Dependent Services:
DNS1 Name: AUTO_DNS3 Index: 2 State: Alive Description: Auto generated for service DNS3 Address:
10.1.1.3 Port: Any Type: SCRIPT ap-kal-dns Script Arguments: "10.1.1.3" Script Error: None !---
Note: This service has access to Internet root servers. Script Run Time: 0 seconds Frequency: 5
Max Failures: 3 Retry Frequency: 5 Dependent Services: DNS3 Name: AUTO_DNS4 Index: 3 State:

```

Alive Description: Auto generated for service DNS4 Address: 10.1.1.4 Port: Any Type: SCRIPT ap-kal-dns Script Arguments: "10.1.1.4" Script Error: None Script Run Time: 0 seconds Frequency: 5 Max Failures: 3 Retry Frequency: 5 Dependent Services: DNS4 Name: AUTO\_Router1 Index: 4 State: Down Description: Auto generated for service Router1 Address: 200.1.1.1 Port: Any Type: ICMP Frequency: 5 Max Failures: 3 Retry Frequency: 5 Dependent Services: Router1 Name: AUTO\_DNS2 Index: 5 State: Down Description: Auto generated for service DNS2 Address: 10.1.1.2 Port: Any Type: SCRIPT ap-kal-dns Script Arguments: "10.1.1.2" Script Error: Script error in line: 41 Script Run Time: 4 seconds Frequency: 5 Max Failures: 3 Retry Frequency: 5 Dependent Services: DNS2 52-css150-4#

#### **affichez le keepalive-résumé**

```
52-css150-4# show keepalive-summary Keepalives: AUTO_nexthop00002 State: Alive 200.1.1.1
AUTO_DNS1 State: Down 10.1.1.1 AUTO_DNS3 State: Alive 10.1.1.3 AUTO_DNS4 State: Alive 10.1.1.4
AUTO_Router1 State: Down 200.1.1.1 AUTO_DNS2 State: Down 10.1.1.2
```

#### **résumé de show service**

```
52-css150# show service summary Service Name State Conn Weight Avg State Load Transitions
AUTO_DNS1 Down 0 1 2 0 AUTO_DNS3 Alive 0 1 2 1 AUTO_DNS4 Alive 0 1 255 1 AUTO_DNS2 Down 0 1 255
0
```

## [Informations connexes](#)

- [Support produit de la Distribution de vidéo et de contenu](#)
- [Support et documentation techniques - Cisco Systems](#)