



# ボックスツーボックス冗長性の設定

この章では、同じように設定された 2 台のコンテンツ サービス スイッチ (CSS) 間に冗長性を設定する方法を説明します。この章の内容は、特に指定のない限り、すべての CSS モデルに適用されます。

この章の主な内容は次のとおりです。

- [CSS 冗長性の概要](#)
- [冗長プロトコルの概要](#)
- [冗長性設定のクイック スタート](#)
- [冗長 CSS のケーブル接続](#)
- [冗長性の設定](#)
- [冗長性設定の同期化](#)
- [redundancy force-master コマンドの使用方法](#)
- [複数の冗長アップリンク サービスの設定](#)
- [redundancy-phy コマンドの使用方法](#)
- [ステートレス冗長フェールオーバーの設定](#)
- [冗長性設定の表示](#)

## CSS 冗長性の概要

冗長性によって、次の点がより確実になります。

- ネットワーク アプリケーションの高可用性
- シングル ポイント障害による、長期のネットワーク遅延またはブラックホールの回避

CSS は次の 3 つのタイプの冗長性設定を備えています。

- 仮想 IP (VIP) および仮想インターフェイスの冗長性: フェイト シェアリングとサーバのデフォルト ゲートウェイのための、冗長 VIP アドレスと冗長仮想インターフェイスを提供します。詳細については、[第 1 章「VIP および仮想インターフェイスの冗長性の設定」](#)を参照してください。
- Adaptive Session Redundancy (ASR; 適用型セッション冗長性): セッションレベルの冗長性 (ステートフル フェールオーバー) を実現します。マスター CSS がバックアップ CSS にフェールオーバーしても、アクティブフローは中断しません。詳細については、[第 2 章「適応型セッションの冗長性」](#)を参照してください。
- ボックスツーボックス冗長性: 2 台の同一に設定された CSS 間のシャーシレベルの冗長性を実現します。詳細については、この章の説明を参照してください。

ここでは、このさまざまな冗長性のタイプに応じた使用形態について説明します。

## VIP 冗長性および仮想インターフェイス冗長性を使用する環境

通常 VIP 冗長性は、サーバファームの前方に設置された CSS ピアの公衆側に設定します。仮想インターフェイス冗長性は、サーバの前方のレイヤ 2 装置に接続された私設側インターフェイスに設定します。

次のような場合に VIP 冗長性を設定します。

- 仮想インターフェイス冗長性により、フェイト シェアリングを実現する場合
- VIP が存在する 2 台の CSS 間に、共通のサブネットがある場合
- ASR 設定の前提条件として必要な場合 (アクティブ - バックアップ VIP 冗長性が必要)
- アクティブ - アクティブ CSS 動作 (両方の CSS がフローを処理) の実現に必要な場合

次のような場合にインターフェイス冗長性を設定します。

- VIP インターフェイス冗長性により、フェイトシェアリングを実現する場合
- バックエンドサーバにデフォルトゲートウェイが必要な場合
- VIP がアップリンクのサブネットと違うサブネット上にある場合に、CSS のクライアント側の VIP 冗長性の代替とする場合

## ASR を使用する環境

ASR は、マスター CSS がバックアップ CSS にフェールオーバーしたときにも、アクティブフロー（TCP と UDP を含む）の中断が許されないような用途において、セッションレベルの冗長性を実現します。

次のような場合に ASR を設定します。

- ミッションクリティカルなアプリケーション（たとえば、企業アプリケーション、HTTP または FTP ファイル転送のような長寿命フロー、および e- コマースなど）に、ステートフルフェールオーバーが必要な場合
- アクティブ-バックアップVIP と仮想インターフェイス冗長性を初めて設定した場合

冗長 CSS ピア間に レイヤ 2 装置のある Inter-Switch Communication（ISC; スイッチ間通信）リンクは設定しないでください。

## ボックスツーマックス冗長性を使用する環境

次のような場合にボックスツーマックス冗長性を設定します。

- CSS の動作をアクティブ/スタンバイ（マスター CSS のみがフローを処理）にする場合
- CSS 間に Virtual Router Redundancy Protocol (VRRP; 仮想ルータ冗長プロトコル) ハートビートのための専用 Fast Ethernet (FE) リンクを設定できる場合

次のような場合は、ボックスツーマックス冗長性を設定できません。

- CSS の動作をアクティブ-アクティブ（両方の CSS がフローを処理）にする場合。この場合、VIP 冗長性を使用してください。
- CSS 間に専用 FE リンクが設定できない場合
- 冗長 CSS ピア間にレイヤ 2 装置の冗長性が必要な場合

## 冗長プロトコルの概要

CSS の冗長プロトコルは、2 台の CSS 間にシャーシレベルの冗長性を提供します。この機能を使用することでネットワークが保護され、ユーザはサーバとコンテンツに確実かつ継続的にアクセスできるようになります。

冗長プロトコル CLI コマンドを使用して、2 台の CSS をマスターおよびバックアップとして冗長設定できます。冗長設定では、マスター CSS は冗長プロトコル メッセージ（ハートビート）を每秒送信して、マスターがオンラインであることをバックアップ CSS に知らせます。

マスター CSS に障害が発生し、3 秒以内にメッセージの送信を行わない場合、バックアップ CSS では次の処理が行われます。

- マスター CSS への移行
- 冗長プロトコル メッセージの送信開始
- 近隣ノードの Address Resolution Protocol (ARP; アドレス解決プロトコル) テーブルおよび接続されているブリッジング デバイス (たとえばレイヤ 2 スイッチ) の転送テーブルをアップデートするために Gratuitous ARP メッセージを送信する。CSS は 1 つの ARP 要求パケットを送信し、gratuitous ARP の呼び出しごとに 1 つの ARP 応答パケットを送信します。

元のマスター CSS が再びオンラインに戻っても、設定時にマスター CSS として明示的に指定していない限り、その CSS はマスター CSS からのメッセージを受け取り次第、自動的にバックアップ CSS になります。IP 冗長性の詳細については、この章で後述する「[IP 冗長性の設定](#)」を参照してください。

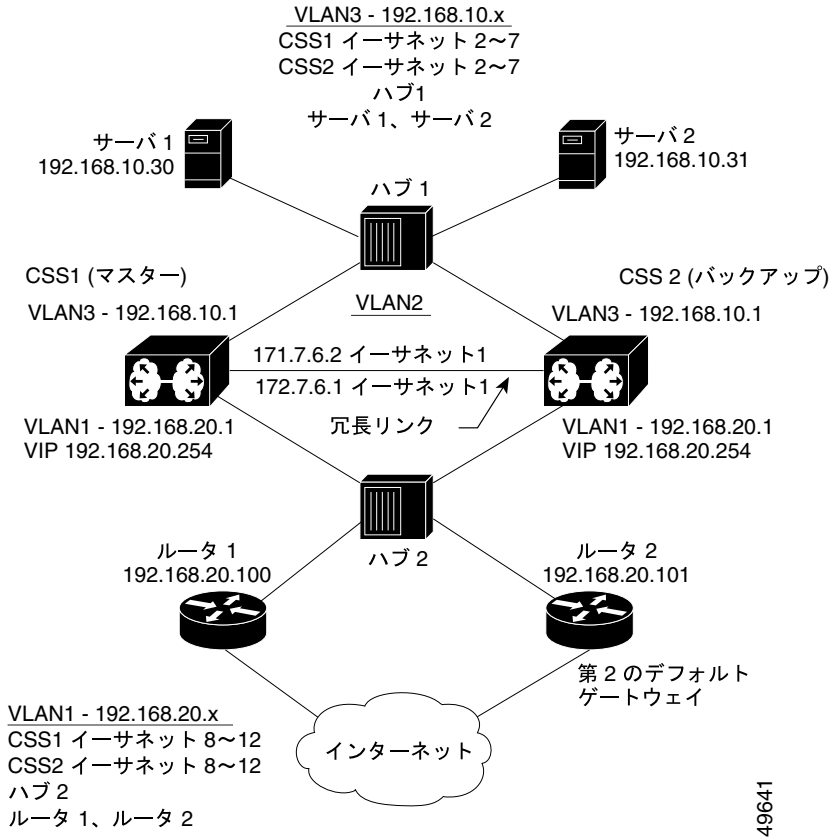


### 注意

冗長性設定で Access Control List (ACL; アクセス コントロール リスト) を使用する場合は、マスター CSS とバックアップ CSS 間の冗長回線で、すべてのトラフィックの搬送を許可するようにしてください。ACL の詳細については、『*Cisco Content Services Switch Security Configuration Guide*』を参照してください。

図 3-1 に、複数の VLAN で構成される冗長性設定の例を示します。表 3-1 では、この図を使用して冗長プロトコルを設定するために必要なコマンドの例を定義します。

図 3-1 冗長性設定の例



## 冗長性設定のクイック スタート

表 3-1 に、図 3-1 で示した冗長性設定を行うのに必要な手順を示します。それぞれの手順には、作業を完了するために必要な CLI コマンドも示しています。各機能の詳細については、表 3-1 の後に続く各項を参照してください。

冗長性を設定するための基本手順を次に示します。

1. CSS の電源を入れる前に、1 本のクロス ケーブルだけを使用してマスター CSS と冗長 CSS を接続します。



### 注意

ケーブルを接続する前に CSS の電源を入れると、どちらの装置もマスター CSS として起動し、ネットワーク障害が発生します。冗長設定ではクロス ケーブルを使用してください。クロス ケーブルを外すとどちらの CSS もマスターになります。



(注) 1 本のクロス ケーブルだけを使って、冗長 CSS 上の Gigabit Ethernet (GE; ギガビット イーサネット) ポート (ソフトウェア バージョン 7.10.1.02 以降)、または Fast Ethernet (FE; ファースト イーサネット) ポートを直接接続する必要があります。冗長リンクの 2 つの CSS の間では、レイヤ 2 デバイスは使用しないでください。

2. 各サーバのデフォルト ゲートウェイを CSS 回線の VLAN IP アドレスとして設定します。
3. 既存のマスター CSS で冗長性を設定し、実行設定 (running-config) を起動設定 (startup-config) にコピーします。
4. FTP を使用して起動設定を PC に転送します。ファイルを編集して、バックアップ CSS 回線の VLAN IP アドレスを含めます。
5. FTP を使用して起動設定をバックアップ CSS に転送します。バックアップ CSS を新しい起動設定で再度ブートします。

CLI コマンドを使用して、冗長プロトコルを含むバックアップ CSS に必要なすべての設定を手動で行うこともできます。



(注) マスター CSS の起動設定を変更する場合は、バックアップ CSS の起動設定にも同じ変更を加える必要があります。マスター CSS とバックアップ CSS の実行設定を同期化する方法については、この章で後述する「[冗長性設定の同期化](#)」を参照してください。

表 3-1 冗長性設定のクイック スタート

作業とコマンドの例

1. CSS の電源を入れる前にクロス ケーブルを接続します。CSS 間は必ずカテゴリ 5 クロス ケーブルで接続します。この表では、マスター CSS、バックアップ CSS のどちらも、ポート 1/1 を使用しています。
2. 各サーバのデフォルト ゲートウェイを CSS 回線の VLAN IP アドレスとして設定します。
3. マスター CSS で **ip redundancy** コマンドを入力し、CSS 間の冗長性を有効にします。  

```
(config)# ip redundancy
```
4. バックアップ CSS へ冗長接続するためのインターフェイスをマスター CSS で設定します。インターフェイスの設定の詳細については、『*Cisco Content Services Switch Routing and Bridging Configuration Guide*』を参照してください。  

```
(config)# interface 1/1
```
5. マスター CSS のインターフェイスを冗長接続の VLAN に割り当てます。インターフェイスの VLAN へのブリッジの詳細については、『*Cisco Content Services Switch Routing and Bridging Configuration Guide*』を参照してください。  

```
(config-if[e1])# bridge vlan 2
```
6. 冗長 VLAN の回線モードに入ります。回線の設定の詳細については、『*Cisco Content Services Switch Routing and Bridging Configuration Guide*』を参照してください。  

```
(config-if[e1])# circuit VLAN2  
(config-circuit[VLAN2])#
```

表 3-1 冗長性設定のクイック スタート (続き)

## 作業とコマンドの例

7. VLAN2 の回線に IP アドレスおよびサブネット マスクを割り当てます。回線の IP アドレスの設定の詳細については、『*Cisco Content Services Switch Routing and Bridging Configuration Guide*』を参照してください。

```
(config-circuit[VLAN2])# ip address 172.7.6.1/24
```

8. 冗長 IP インターフェイスの冗長プロトコルを有効にします。

```
(config-circuit-ip[VLAN2-172.7.6.1])# redundancy-protocol
(config-circuit-ip[VLAN2-172.7.6.1])# exit
```

9. 他の回線を冗長回線として定義します。

```
(config-if[1/1])# circuit VLAN1
(config-circuit[VLAN1])# redundancy
(config-if[1/1])# circuit VLAN3
(config-circuit[VLAN3])# redundancy
```

10. スーパーユーザ モードでマスター CSS の実行設定を起動設定にコピーします。

```
# copy running-config startup-config
```

11. 起動設定を編集するため、FTP を使用して PC に転送します。

12. テキスト エディタを使用して起動設定ファイルを編集し、冗長接続用のバックアップ CSS 回線の VLAN IP アドレス (図 3-1 では、回線は VLAN2、IP アドレスは 172.7.6.2/24) を追加します。バックアップ CSS の VIP は変更しないでください。マスター CSS とバックアップ CSS の VIP は同一にする必要があります。VIP が複数ある場合は、マスターとバックアップの両方の CSS にそれらの VIP を設定する必要があります。

13. FTP を使用して新しいファイルをバックアップ CSS へ転送し、必要に応じてファイル名を **startup-config** に変更します。

14. バックアップ CSS を再度ブートします。

15. **show redundancy** コマンドを実行して冗長性設定を表示し、バックアップ CSS が適切に設定されていることを確認します。

16. すべてのハブ (またはスイッチ) をバックアップ CSS にケーブルで接続します。



次の running-config の例は、表 3-1 のコマンドを実行した結果を示します。

```
!***** GLOBAL *****
ip redundancy

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

interface 2/1
  bridge vlan 3

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

  ip address 172.27.16.23 255.255.255.0

circuit VLAN2

  ip address 172.7.6.1 255.255.255.0
  redundancy-protocol

circuit VLAN3
  redundancy
```

## 冗長 CSS のケーブル接続

冗長性設定を行う場合、カテゴリ 5 のクロス ケーブル 1 本だけを 2 台の CSS に直接接続してマスター インターフェイスとバックアップ インターフェイスを接続します。2 台の CSS 間での複数の冗長リンクはサポートしていません。



### 注意

ケーブルを接続する前に CSS の電源を入れると、どちらの CSS もマスター CSS として起動し、ネットワーク障害が発生します。冗長性設定では必ずクロス ケーブルを使用してください。クロス ケーブルを外すとどちらの CSS もマスターになります。



### (注)

冗長 CSS の電源を入れる前に、ギガビット イーサネット (GE) ポート (ソフトウェア バージョン 7.10.1.02 以降) またはファースト イーサネット (FE) ポートをクロス ケーブルで直接接続する必要があります。冗長リンクの 2 つの CSS の間では、レイヤ 2 デバイスは使用しないでください。

表 3-2 に、CSS ファースト イーサネット コネクタおよびクロス ケーブルのピン配置を示します。

**表 3-2 RJ-45 ファースト イーサネット コネクタのピン配置**

シグナル名	ピン番号	クロス ケーブルのピン配置
RX +	1	3
RX -	2	6
TX +	3	1
接続なし	4	4
接続なし	5	5
TX -	6	2
接続なし	7	7
接続なし	8	8

## 冗長性の設定

冗長プロトコルを設定するには、次の作業を行う必要があります。

- マスター CSS とバックアップ CSS を設定して、冗長性を確立する (**ip redundancy** コマンドを使用)。
- マスターおよびバックアップ回線 VLAN で冗長プロトコルを有効にする (**redundancy-protocol** コマンドを使用)。
- 回線 VLAN の冗長性を有効にする (**redundancy** コマンドを使用)。



(注) CSS は IP 冗長性と VIP 冗長性を同時にはサポートしません。VIP 冗長性の詳細については、第1章「VIP および仮想インターフェイスの冗長性の設定」を参照してください。

## IP 冗長性の設定

**ip redundancy** コマンドを使用して、クロス ケーブルで接続された 2 台の CSS の冗長性を有効にします。デフォルトでは、このコマンドを両方の CSS で実行するまで、冗長性は有効になりません。

このコマンドで **master** オプションを指定すると、マスター CSS として使用する CSS を指定できます。マスター CSS とバックアップ CSS は、クロス ケーブルで接続された 2 台の CSS を起動したときに初期設定されます。先に起動する方がマスター CSS になります。2 台の CSS が同時に起動した場合は、IP アドレスの数字が大きい方がマスターになります。



(注) **ip redundancy master** コマンドは、**type redundancy-up** コマンド (冗長アップリンク サービス) や **redundancy-phy** コマンド (物理リンク冗長性) とは併用できません。**ip redundancy master** コマンドを使用する前に、必要に応じて **no type redundancy-up** コマンドや **no redundancy-phy** コマンドを実行し、対応するコマンドを無効にしてください。

CSS で **ip redundancy master** コマンドを実行すると、その CSS がマスター CSS になります。このコマンドは、現在のマスターとバックアップのどちらでも実行できます。このオプションをバックアップ CSS で実行すると、バックアップは自動的にマスターになり、マスターはバックアップになります。

マスター CSS を指定すると、一度オフラインになっても、再びオンラインになったときにその CSS はマスターのステータスを取り戻します。たとえば、マスター CSS がオフラインになるとバックアップ CSS がマスターになりますが、マスターとして指定されていた CSS がオンラインに戻ると、その CSS が再びマスターになります。

両方の CSS が起動しており、マスター CSS を指定する必要がない場合は、マスター CSS の冗長性を有効にする際に **master** オプションを指定しないでください。指定しない場合、マスター CSS がオフラインになると、バックアップ CSS がマスターになります。元のマスター CSS がオンラインに戻ると、その CSS はバックアップになります。

このグローバル設定モードのコマンドのシンタックスは次のとおりです。

- **ip redundancy** : バックアップ CSS で CSS 間の冗長性を有効にする。マスターにする CSS がどちらでもかまわない場合、このコマンドは冗長設定する両方の CSS で実行してください。
- **ip redundancy master** : マスターとして指定する CSS で CSS 間の冗長性を有効にする (バックアップ CSS では、必ず **ip redundancy** を実行してください)。**ip redundancy master** は、次の場合に実行できます。
  - マスターとバックアップのどちらで起動したかに関係なく、特定の CSS をマスターに指定する場合。このオプションをバックアップ CSS で実行すると、自動的にバックアップはマスターになり、マスターはバックアップになります。
  - CSS 間の冗長性がすでに有効になっている場合

たとえば、次のように入力します。

```
(config)# ip redundancy
```



注意

**ip redundancy master** コマンドは、マスター CSS とバックアップ CSS の両方で実行しないでください。両方の CSS でこのコマンドを実行すると、重大なネットワーク障害が発生する場合があります。ネットワークでの IP アドレスの重複を避けるため、冗長性を無効にする際には、事前にすべての冗長回線の接続を切るか、無効化してください。

CSS 間の冗長性を無効にするには、次のように入力します。

```
(config)# no ip redundancy
```



(注)

**no ip redundancy** コマンドを実行すると、CSS の実行設定から **redundancy** および **redundancy-protocol** コマンドが削除されます。

CSS は冗長性を無効化する前に、次のメッセージを表示します。

```
WARNING: Disabling redundancy may result in duplicate
IP addresses on the network.
Be sure you disconnect or disable all redundant circuits before you
disable redundancy.
Do you want to disable redundancy? [y/n]:
```

冗長性を無効にするには、**y** を入力します。冗長性の無効化をキャンセルするには、**n** を入力します。

マスター CSS の割り当てを解除するには、次のコマンドを入力します。

```
(config)# no ip redundancy master
```



(注)

**no ip redundancy master** コマンドを実行しても、CSS 間の冗長性は無効になりません。

## 冗長回線の設定

回線を冗長回線として設定するには、**redundancy** コマンドを使用します。**redundancy** コマンドは回線設定モードで使用できます。



(注) **redundancy** コマンドを使用すると、指定した VLAN はバックアップ モード時には休止状態になります。

冗長性を設定する回線 (VLAN) は、冗長 CSS によって共有されているネットワーク IP アドレスを含む回線 (図 3-1 では VLAN1 および VLAN3) です。冗長性通信のために特別に設定されている回線 (図 3-1 では、VLAN2) では、冗長性の設定を行わないでください。

次に示す例では、共有のネットワーク IP アドレス (図 3-1 では 192.168.20.1) を含む VLAN1 を冗長回線として設定しています。

たとえば、次のように入力します。

```
(config-circuit [VLAN1])# redundancy
```

冗長性設定から回線を削除するには、次のコマンドを入力します。

```
(config-circuit [VLAN1])# no redundancy
```

## 冗長プロトコルの設定

2 台の CSS を接続する回線 (VLAN) に冗長プロトコルを設定するには、IP インターフェイス設定モードで **redundancy-protocol** コマンドを実行します。このコマンドは、冗長性通信のために特別に設定した回線 (図 3-1 では VLAN2) を対象として実行します。



(注) CSS ボックスツーボックス冗長プロトコルは、ソフトウェア バージョンが 7.10.1.02 以降である CSS のギガビットイーサネット (GE) ポートでサポートされています。

たとえば、次のように入力します。

```
(config-circuit-ip[VLAN2-172.7.6.1])# redundancy-protocol
```

インターフェイスでの冗長プロトコルの実行を停止するには、次のコマンドを入力します。

```
(config-circuit-ip[VLAN2-172.7.6.1])# no redundancy-protocol
```

## VRRP バックアップ タイマーの設定

**vrrp-backup-timer** コマンドを両方の冗長 CSS 上で設定して、マスター CSS がダウンしてからバックアップ CSS がマスターになるまでの待機時間を秒数で指定します。このタイマーの値をデフォルトの 3 秒を超える値にすると、フェールオーバー時間が長くなるので、このコマンドは、CSS の CPU 利用率が 100 % に近い環境でだけ使用してください。タイマーの値を設定した後、新しいタイマーを有効にするには、CSS 間の冗長回線に対して **redundancy-protocol** コマンドを再実行する必要があります。冗長プロトコルの設定の詳細については、この章で前述した「[冗長プロトコルの設定](#)」を参照してください。



(注)

**commit\_redundancy** スクリプトを使用して冗長性設定を同期化する場合は、**script play** コマンドに引数 **-a** を指定して、タイマーの設定がマスター CSS からバックアップ CSS に確実にコピーされるようにします。冗長性設定の同期化の詳細については、この章で後述する「[冗長性設定の同期化](#)」を参照してください。

このグローバル設定モードのコマンドのシンタックスは次のとおりです。

```
vrrp-backup-timer wait_time
```

このコマンドの変数は *wait\_time* です。3 ~ 120 (秒) の整数を入力します。デフォルトは 3 秒です。

たとえば、次のように入力します。

```
(config)# vrrp-backup-timer 15
```

タイマーをデフォルトの 3 秒にリセットするには、次のように入力します。

```
(config)# no vrrp-backup-timer
```

## 冗長性設定の同期化

マスター CSS で障害が発生した場合に、バックアップ CSS にマスター CSS と同じタスクを実行させるには、バックアップの実行設定をマスターの実行設定と同一にする必要があります。この設定同期のプロセスは、マスター CSS の実行設定をバックアップ CSS の実行設定にコピーするスクリプト (`commit_redundancy`) を、マスター CSS で実行することによって自動化できます。

設定の同期化には次の 2 つの種類があります。

- **設定完全同期化**：同一のシャーシを持つ CSS (同じ CSS モデル) で、マスター CSS の実行設定と完全に一致する実行設定をバックアップ CSS に作成します。
- **設定部分同期化 (デフォルト)**：互換性のない設定シンタックスをもつ CSS 間で、インターフェイス設定と回線設定を除くすべてのパラメータ値を同期します。たとえば、マスターが CSS 11506、バックアップが CSS 11501 の組み合わせで同期化スクリプトを実行すると、バックアップ側のインターフェイス設定と回線設定は自動的に維持されます。

## スクリプト実行前の作業

設定同期化スクリプトを実行する前に、2 台の CSS 間の冗長回線設定が完了しており、その回線で APP (Application Peering Protocol) が有効になっていることを確認してください。冗長回線の設定の詳細については、この章で前述した「[冗長性の設定](#)」を参照してください。APP の設定の詳細については、『*Cisco Content Services Switch Global Server Load-Balancing Configuration Guide*』を参照してください。

CSS 上で `restrict user-database` コマンドを設定する場合、管理者または技術者権限のあるユーザだけが `username` コマンドを設定することができます。`restrict user-database` コマンドとの一貫性をもたせるため、`commit_redundancy` スクリプトは、バックアップ CSS で管理者ユーザまたは技術者ユーザの権限レベルを変更しません。

`commit_redundancy` コマンドは、両方の CSS で同じ管理者権限または技術者権限を持っているユーザが操作している場合に限り成功します。マスター CSS 上でローカル ユーザになり (`username` コマンドを使用して設定)、バックアップ CSS 上で同じユーザ名を用いて管理者または技術者になることはできません。



**restrict user-database** コマンドの詳細については、『*Cisco Content Services Switch Security Configuration Guide*』を参照してください。管理者の設定 (**username\_offdm** コマンド) および技術者の設定 (**username\_technician** コマンド) についての詳細は、『*Cisco Content Services Switch Command Reference*』を参照してください。

## 設定同期化スクリプトの実行

設定同期化スクリプトを実行するには、**script play commit\_redundancy** コマンドを使用します。このコマンドのシンタックスは次のとおりです。

### **script play commit\_redundancy** “arguments”

また、次のコマンドを実行することにより、すべての CSS に付属する定義済みのエイリアスを使用して、設定同期化スクリプトを実行することもできます。

```
# commit_RedundConfig “arguments”
```

commit\_redundancy スクリプトの引数は次のとおりです。

- **ip address** : バックアップ CSS の IP アドレス。このスクリプトの唯一の必須引数です。IP アドレスの自動入力の詳細については、この章で後述する「[REMOTE\\_IP 変数の設定](#)」を参照してください。
- **-a** (All) : 設定を完全に同期化します。このオプションは、マスター CSS とバックアップ CSS のシャーシが同一の場合にだけ使用します (表 3-3 を参照)。
- **-d** (Debug) : commit\_redundancy スクリプトのデバッグスイッチ。スクリプトの進行に応じて、実行中のタスクを順次表示します。デバッグメッセージは、引数 **-s** を指定した場合でも表示されます。



### 注意

引数 **-f** を使用して設定同期化ロック ファイルを削除する前に、他のユーザが設定同期化スクリプトをその CSS 上で実行していないことを確認してください。他のユーザによるスクリプトの実行中にロック ファイルを削除し、スクリプトを再実行すると、生成される設定に不整合が生じるおそれがあります。

- **-f** : スクリプトが異常終了した場合にロック ファイルを削除し、スクリプトを再実行できる状態にします。この引数は、指定した他の引数をすべて無効にし、ロック ファイルを削除するとスクリプトがすぐに終了します。ロック ファイルの詳細については、この項で後述する「[REMOTE\\_IP 変数の設定](#)」を参照してください。
- **-int (Interface)** : リンクの停止を回避するため、バックアップ CSS のインターフェイスをクリアしません。この引数を使用する場合には、**-a** 引数は使用しないでください。使用すると、マスター CSS とバックアップ CSS のインターフェイス設定に相違がある場合、設定が一致しなくなり、スクリプトが異常終了します。
- **-nv (No Verify)** : 設定同期化が成功したかどうかを確認しません。失敗したかどうかはスクリプトが通知します。



(注) このスクリプトは、デフォルトでは、設定同期化の確認を行います。

- **-s (Silent)** : スクリプトの進行状況を示すメッセージは表示せず、スクリプトの実行結果 (Config Sync Successful または Config Sync Failed) だけを表示します。



(注) **-s** オプションは、バックアップ CSS で実行する `commit_redundancy` スクリプトとは互換性がありません。**-s** オプションを使用してスクリプトをバックアップ CSS で実行すると、スクリプトは設定を変更せずに終了します。



(注) スクリプト引数は任意の順序で指定できます。

たとえば、次のように入力します。

```
CSS11503# script play commit_redundancy "10.7.100.93 -d -s"
```

次のようなメッセージが表示されます。

```
Verifying that IP redundancy is activated on Master switch.
```

```
Verifying that app session is up with backup switch.
```

```
Making sure app session is up.
```

```
Checking Master redundancy-config for redundancy-protocol set and if  
so storing it in variable MASTER_IP.
```

```
Verify that the IP Address specified is the Backup IP Address.
```

```
Making sure app session is up.
```

```
Saving Master running-config to startup-config and archiving  
startup-config.
```

```
Copying running-config to startup-config.
```

```
Archiving startup-config.
```

```
Copying startup-config to a temp file tmp.cfg.
```

```
Swapping Master and Backup ip addresses in tmp.cfg for app
```

```
Removing CIRCUIT and INTERFACE modes from tmp.cfg.
```

```
Checking if IP redundancy master is set.
```

```
Using rcmd to copy tmp.cfg to a file on Backup switch.
```

```
Retrieving circuit info for redundancy-protocol link.
```

```
Archiving copy to Backup startup-config.
```

```
Archiving Backup current startup-config.
```

```
Restoring startup-config (new copy) to startup-config.
```

```
Clearing running-config.
```

```
Script playing the copy script of the Master running-config.
```

```
Making sure app session is down.
```

```
Copy success being verified by comparing byte sizes of archived  
running-configs of the Master switch and the Backup switch.
```

```
Making sure app session is up.
```

```
Comparing the byte count now.
```

```
Config Sync Successful.
```

このスクリプトの例では、次の処理が実行されています。

- 設定部分同期化（デフォルト）
- スクリプトの進行に伴う実行中のタスクの表示 (-d)
- 進行状況メッセージの非表示 (-s)
- 設定同期化の結果の検証 (-v)

詳細については、『*Cisco Content Services Switch Administration Guide*』を参照してください。

## 設定同期化ロック ファイル

スクリプトを実行すると、スクリプト自身の ディレクトリ内にロック ファイル (`config_sync_lock`) が作成され、同じ CSS 上の他のセッションによるスクリプトの呼び出しが防止されます。ロック ファイルが存在している場合、同期化スクリプトの実行を試みると、次のメッセージが表示されます。

```
The script is in use by another session.
```

スクリプトが異常終了した場合、ロック ファイルは削除されません。次にスクリプトを実行したときに、上記のメッセージが表示されます。スクリプトが別のセッションで実行されていないことが確かである場合、引数 `-f` を使用してロック ファイルを削除できます。

`-f` 引数を使用してスクリプトを実行すると、次のようなメッセージが表示された後、スクリプトが終了します。

```
Config Sync lock file removed.
```

これでスクリプトを再実行できます。

## REMOTE\_IP 変数の設定

`commit_redundancy` 設定同期化スクリプトを実行するたびにリモート IP アドレスを指定するのは煩雑です。変数 `REMOTE_IP` に IP アドレスの値を設定して、ユーザ プロファイルに保存すれば、この手間を省くことができます。値を設定してユーザ プロファイルに保存すれば、この変数は CSS にログインするたびに有効になります。

`REMOTE_IP` 変数に、ローカル CSS に設定されている APP セッション IP アドレスを設定します。APP セッションアドレスは、リモート CSS の回線の IP アドレスです。変数に値を設定するには、次のように入力します。

```
# set REMOTE_IP "remote_ip_address" session
```

ユーザ プロファイルに変数を保存するには、次のコマンドを入力します。

```
# copy profile user-profile
```

以前のリリースで `BACKUP_IP` 変数をすでに作成している場合には、スクリプトは代わりに新しい変数（存在する場合）を使用します。

## 設定同期化スクリプトの結果メッセージのロギング

設定同期化スクリプトを実行するたびに、スクリプト結果メッセージ（スクリプトの成功または失敗を示すメッセージ）が現在のロギング デバイスに自動で送信されるように指定できます。スクリプト結果メッセージをログ ファイルに記録するには、次のコマンドを入力して `info-6` または `debug-7` レベルの NETMAN でのロギングを有効にします。

```
(config)# logging subsystem netman level info-6
```



(注) ログ メッセージは、引数 `-s` (`silent`) の指定の有無に関係なく作成されます。この章の「[設定同期化スクリプトの実行](#)」を参照してください。

■ **redundancy force-master** コマンドの使用法

たとえば、バックアップ CSS に対して APP セッションが実行されていない場合は、次のようなログメッセージが生成されます。

```
config sync: app session is DOWN
```

ログの追跡を容易にするため、各ログメッセージには文字列「config sync」が含まれています。

## redundancy force-master コマンドの使用法

**redundancy force-master** コマンドを使用して、バックアップ CSS を一時的なマスターとして設定します。これは実行設定にコピーされない一時的な設定です。このコマンドは、冗長性設定でメンテナンスやアップグレードのためにマスター CSS をオフラインにする必要があるときに役立ちます。

バックアップ CSS のグローバル設定モードで **redundancy force-master** コマンドを実行すると、バックアップ CSS がマスターとして設定され、ユーザによるサーバおよびコンテンツへのアクセスも確実に継続されます。マスターになったバックアップ CSS は、次のいずれかの状況が生じるまでマスターとしてのステータスを維持します。

- 一度オフラインになり、バックアップとして再度オンラインになる。
- **redundancy force-master** コマンドまたは **ip redundancy master** コマンドを使用して、他の CSS を明示的にマスターに指定する。



(注)

---

**ip redundancy master** コマンドを使用して明示的にマスター CSS を指定している場合には、バックアップ CSS で **redundancy force-master** コマンドを実行することはできません。その場合には、バックアップ CSS で **redundancy force-master** コマンドを実行する前に **no ip redundancy master** コマンドを実行し、マスター CSS の割り当てを解除する必要があります。

---

## 複数の冗長アップリンク サービスの設定

CSS の冗長性設定では、複数の冗長アップリンク クリティカル サービス（最高 512 まで）を設定できます。 **type redundancy-up** コマンドを使用して、1 台以上のルータを **redundancy-up** タイプのクリティカル サービスとして指定します（通常、設定に含まれるルータは 10 台以下）。このクリティカル サービスタイプを指定すると、デフォルトのキープアライブ Internet Control Message Protocol (ICMP; インターネット制御メッセージプロトコル) を使用して、マスター CSS からルータ サービスに PING を実行できます。マスター CSS に障害が発生したり、すべてのルータ アップリンク クリティカル サービスがダウンしたことをマスター CSS が検出すると、バックアップ CSS がマスターになります。

ルータを **redundancy-up** タイプのクリティカル サービスとして設定していない冗長性設定では、バックアップ CSS は現在のマスター CSS がダウンした場合のみマスターになります。このような設定の場合、ルータ サービスがダウンしても切り替えは行われません。

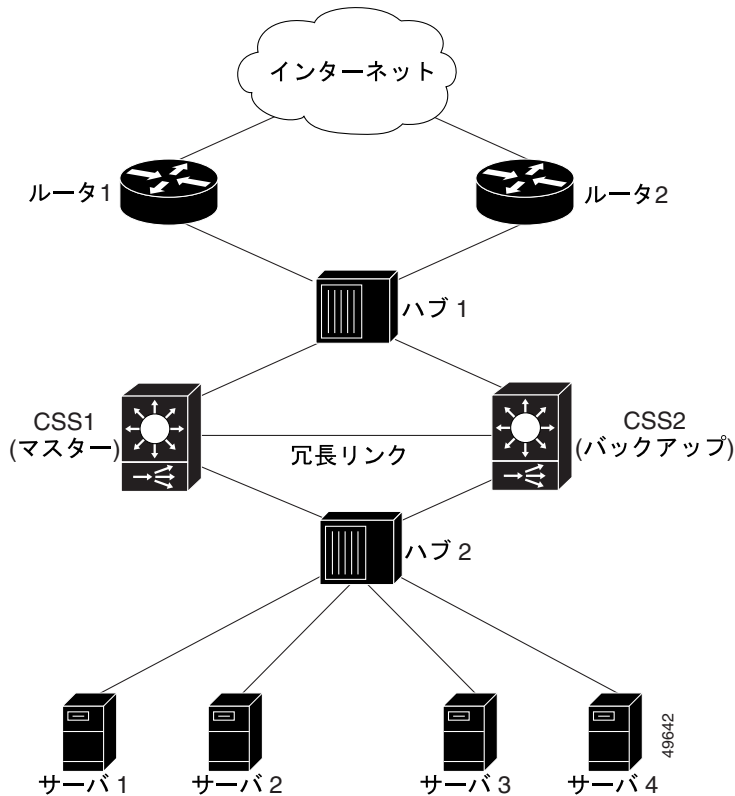


(注)

冗長アップリンク クリティカル サービスは、コンテンツ ルールには追加できません。

図 3-2 に、標準的な冗長性設定を示します。CSS1 に障害が発生した場合、またはルータ 1 サービスおよびルータ 2 サービスの両方と通信できない場合は、CSS2 が自動的にマスター CSS になります。

図 3-2 複数の冗長アップリンク サービスの設定例



(注)

**ip redundancy master** コマンドを使用して明示的にマスター CSS を指定している場合、その CSS では **type redundancy-up** コマンドは実行できません。その場合には、**type redundancy-up** コマンドを実行する前に **no ip redundancy master** コマンドを実行し、マスター CSS の割り当てを解除する必要があります。



**type redundancy-up** コマンドを使用して、各ルータ サービスを冗長アップリンク クリティカル サービスとして設定します。たとえば、次のように入力します。

```
(config-service[router1])# type redundancy-up  
(config-service[router1])# ip address 192.168.1.1  
(config-service[router1])# active
```

**show redundancy** コマンドを使用して、クリティカル サービスを表示します。「[冗長性設定の表示](#)」を参照してください。

たとえば、次のように入力します。

```
CSS1(config)# show redundancy
```

## redundancy-phy コマンドの使用法

インターフェイス モードで **redundancy-phy** コマンドを使用して、物理リンク設定リストにインターフェイスを追加します。設定リスト内の物理リンクが 1 つでもダウンするとフェールオーバーが発生し、バックアップ CSS がマスター CSS になります。最大で 32 個のインターフェイスを設定できます。この設定情報は、CSS が実行設定に保存します。

たとえば、次のように入力します。

```
(config-if [2/1])# redundancy-phy
```



(注)

マスター CSS のインターフェイスで **redundancy-phy** コマンドを設定し、**phy** コマンドを使用してそのインターフェイスのポート設定を変更すると（たとえば、**auto-negotiate** を **100Mbps-FD** に変更）、マスター CSS からバックアップ CSS へフェールオーバーが行われます。フェールオーバーの発生を回避するには、インターフェイスで **no redundancy-phy** コマンドを入力した後、ポート設定を変更し、**redundancy-phy** コマンドを再度入力します。

**ip redundancy master** コマンドでマスター CSS を指定している場合には、**redundancy-phy** コマンドは使用できません。このような場合は、**redundancy-phy** コマンドを使用する前に、**no ip redundancy master** コマンドを入力する必要があります。

設定したインターフェイスを無効にし、物理リンク リストから削除するには、次のコマンドを入力します。

```
(config-if)# no redundancy-phy
```



(注)

インターフェイスで **redundancy-phy** コマンドを入力し、**admin-shutdown** コマンドを使用してインターフェイスを無効にすると、マスター CSS からバックアップ CSS へフェールオーバーが行われます。管理のためにインターフェイスを無効にしたときに CSS のフェールオーバーの発生を回避するには、**no redundancy-phy** コマンドを入力して **redundancy-phy** コマンドを削除してから、そのインターフェイスで **admin-shutdown** コマンドを入力します。

両方の CSS がレイヤ 2 スイッチに接続している状態で **redundancy-phy** コマンドを使用する場合、物理リンク障害の有無を監視すべき対象はクリティカル物理リンクだけです。CSS 間の冗長リンクは監視しません。このように監視を行うことで、一方の CSS のリブートやマスター/バックアップの状態切り替えの際に、誤って物理リンクのダウンやスラッシングを検出する事態を回避できます。

## ステートレス冗長フェールオーバーの設定

コンテンツ設定モードまたはグループ設定モードで **redundancy-l4-stateless** コマンドを使用して、ボックスツーボックス冗長性または VIP および 仮想インターフェイス冗長性が設定されている CSS で、ステートレス冗長フェールオーバーを有効にします。ステートレス冗長フェールオーバーでは、ロード バランシング CSS で障害が発生したときに、バックアップ CSS でミッドストリーム TCP フローをセットアップすることによって、重要な TCP/IP トラフィックを継続させることができます。デフォルトでは、この機能は無効に設定されています。

CSS のデフォルトの動作では、SYN で始まる TCP フレームを受信した場合のみ、ロード バランシングした TCP フローがセットアップされます。ステートレス冗長フェールオーバーが有効な場合にフェールオーバーが発生すると、バックアップ CSS が既存のすべての TCP セッションを対象に、ミッドストリーム フローを確立します。新しいフローについては、CSS の動作はデフォルトのままです。**redundancy-l4-stateless** コマンドの実行後、すべてのフローを対象に CSS の動作をデフォルトに戻すには、**no redundancy-l4-stateless** コマンドを実行します。



(注) この機能は、TCP/IP セッションだけに影響します。UDP はセッション指向のプロトコルではないため、動作は通常どおりです。

## スクリプト実行前の作業

ステートレス冗長フェールオーバーを初めて実装する場合には、作業を始める前に、この項全体に目を通してください。また、次にあげる概念を十分に理解していることが望まれます。

- 冗長性
- レイヤ 3 および 4 のコンテンツ ルール
- VIP (仮想 IP アドレス)
- ロード バランシング
- ソース グループ
- サービス
- キープアライブ
- コンバージェンス

## 環境に関する留意事項

ステートレス冗長フェールオーバーには、非常に特殊な冗長 CSS 設定（障害発生後の CSS 状態を判定可能にする）が必要です。この機能では、CSS を取り巻く高可用トポロジでの冗長ルートがサポートされます。ただし、このトポロジでは、TCP/IP ソケット接続を介して搬送されるパケットを、CSS の複数のイーサネット ポートに分配してバランスをとることは許されません。

したがって、ロード バランシングが設定された VIP のルート パスには、1 本の TCP/IP 接続の存続期間中に、1 本のパスしか優先されないように重み付けする必要があります。



(注) ステートレス冗長フェールオーバーでは、状態の維持が必要な Network Address Translation (NAT; ネットワーク アドレス変換) も、レイヤ 5 のコンテンツルールもサポートされません。

## 一般的な設定条件

ここでは、ボックスツーマップス冗長性と VIP および仮想インターフェイス冗長性の両方の設定に適用されるステートレス フェールオーバーの条件を説明します。

## 設定の制約

次に示す設定の制約は、特に指示のない限り、すべての CSS に該当します。

- ステートレス冗長フェールオーバーは、サービス リマッピング (**persistence reset remap** コマンド) と互換性がありません。ステートレス冗長フェールオーバーでは、CSS がクライアントの発信元ポートを NAT 変換しないことが条件になります。一方、バックエンド リマッピングは CSS によるポートマッピングを可能にしますが、それによってすべてのフローの発信元ポートが NAT 変換されます。サービス リマッピングの詳細については、『*Cisco Content Services Switch Content Load-Balancing Configuration Guide*』を参照してください。
- セッション レベルの冗長性とステートレス冗長フェールオーバーを同じ CSS に設定することはできません。

## ■ ステートレス冗長フェールオーバーの設定

- 送信先ポートを変更するサービスをコンテンツ ルールに設定しないでください。送信先ポートを変更するサービスを設定すると、CSS によって送信先ポートが NAT 変換（ポートマップ）されます。その結果、CSS がフェールオーバーしたときに、バックアップ CSS が元の送信先ポートを認識できなくなります。

## CSS パラメータの設定

次の各パラメータは、両方の冗長 CSS で正確に一致している必要があります。

- ステートレス冗長性フェールオーバー コマンド**：冗長 VIP に関連付けられたコンテンツ ルールとソース グループの両方に **redundancy-l4-stateless** コマンドを含めます。
- コンテンツ ルール**：両方の CSS に、次の各パラメータを含む同一のコンテンツ ルールを作成します。『*Cisco Content Services Switch Content Load-Balancing Configuration Guide*』を参照してください。
  - VIP**：各コンテンツ ルールに割り当てる仮想 IP アドレス。アドレスにワイルドカードは使用できません。また、コンテンツ ルールには VIP 範囲は使用できません。
  - ロード バランシング方式**：発信元 IP アドレスを意味する **srcip** を設定します。ステートレス冗長フェールオーバーでサポートされるロード バランシング方式は、発信元 IP アドレスだけです。
  - フェールオーバー方式**：サービスのフェールオーバー方式として、**linear**（デフォルト）または **next** を設定します。**bypass** はサポートされていません。
- サービス**：ロード バランシングが適用される各サーバ ファームで、両方の CSS のコンテンツ ルールが同一になるように、次のサービス関連のパラメータを設定します。『*Cisco Content Services Switch Content Load-Balancing Configuration Guide*』を参照してください。
  - サービス名**：マスター CSS とバックアップ CSS に同一のサービス名を使用します。サービス名では、大文字と小文字が区別されます。
  - IP アドレス**：マスター CSS とバックアップ CSS に同一の IP アドレスを使用します。



(注) 設定されたサービスは、CSS の送信先ポートには影響しません。ステートレス環境では、サーバからパケットが返されたときに、元の送信先ポートを特定する手段はありません。

- サービス番号と順序: CSS では、設定で入力したサービスの順序に関係なく、サービスが内部でアルファベット順に順序付けられます。
- キープアライブ: グローバルな **keepalive** コマンドを使用してキープアライブを作成します。その後、**keepalive type named** コマンドを使用してサービスをキープアライブに関連付けます。両方の CSS は同じサーバとの間で、キープアライブ メッセージを送受信できる必要があります。これにより、フェールオーバー イベントでバランシング方式の再配布が発生しなくなります。
- 重み: ロード バランシングが設定された VIP のルート パスには、1 本の TCP/IP 接続の存続期間中に 1 本のパスが優先されるように重み付けする必要があります。
- ソース グループ: 各 CSS のコンテンツ ルールの VIP と同じ VIP を持つソース グループを作成して、サーバから返されるパケットの発信元アドレスを NAT 変換するようにします。フェールオーバーが発生すると、サーバから CSS に着信する TCP/IP 転送の接続は、このソース グループによってセットアップされます。ファームのすべてのサーバは、設定されたソース グループのメンバーである必要があります。『Cisco Content Services Switch Content Load-Balancing Configuration Guide』を参照してください。



- (注) マスター CSS で障害が発生した場合、バックアップ CSS では、マスター CSS のソース グループにより NAT が設定されているポートを判断できないため、サーバから発信するトラフィックのソース グループは設定しないでください。データ接続の開始はサーバが行うので、この制約はアクティブな FTP にも適用されます。

## CSS 設定の同期化

CSS 設定を手動で同期化するには、設定がまったく同じであることを確認します。IP 冗長性設定の場合、**commit\_redundancy** 設定同期スクリプトを実行できません。スクリプトによりマスター CSS とバックアップ CSS の設定が自動的に同期化されます。この章の「[冗長性設定の同期化](#)」を参照してください。

## ボックスツーボックス冗長性の設定

ボックスツーボックス冗長性の詳細については、この章で前述した項を参照してください。

マスター CSS で障害が発生した場合は、バックアップ CSS がマスター CSS になります。次のような動作をします。

1. VLAN がすべてアクティブになる。
2. トポロジ プロトコル（スパニングツリーなど）が初期化およびコンバージされる。
3. 設定されたすべてのインターフェイス（回線）と VIP アドレスが、Gratuitous ARP により取得される。
4. マスター CSS では、キープアライブによりサーバが取得される。



(注) 設定規模が大きい場合や、サーバの応答が遅い場合には、ステップ 4 に数秒間かかることがあります。

CSS がルートブリッジを特定し、アドレス解決プロトコル（ARP）とキープアライブトラフィックの転送を開始すると、その CSS を取り巻く複雑なトポロジがコンバージされます。CSS がサーバを取得する前にサーバからの TCP/IP 再送信が CSS に着信した場合、CSS では、設定されたソースグループパス経由で接続が適切にセットアップされます。すべてのサーバが取得される前にクライアントからの再送信が CSS に着信し、クライアントの発信元の IP アドレスがオンラインになっていないサーバを示している場合、CSS では、コンテンツルールで設定されたフェールオーバー方式（next または linear）に基づいて接続がセットアップされます。



## レイヤ 2 およびレイヤ 3 設定とコンバージェンス

IP 冗長性によりバックアップ CSS の VLAN を経由するトラフィック転送が無効になるため、バックアップ CSS でサーバとアップリンク ルータ間のブリッジパスまたはルートパスを設定します。どちらの場合も、バックアップ CSS をロード バランシング サーバのデフォルト ゲートウェイにする必要があります。『*Cisco Content Services Switch Routing and Bridging Configuration Guide*』を参照してください。

複数の CSS を、複数のサーバおよび 1 つのバランス VIP と共に、アップリンク ルータ（ブリッジモード）と同じ VLAN 上で設定する場合は、**no redirect** コマンドを使用して、CSS からサーバに ICMP リダイレクトを送信しないように設定します。『*Cisco Content Services Switch Content Load-Balancing Configuration Guide*』を参照してください。

## 設定例

次に示す設定例（[図 3-3](#) 参照）では、次の事項を前提としています。

- CSS は、IP 冗長性設定で 2 つの外部 VLAN 間のルータとして動作している。
- レイヤ 2 およびレイヤ 3 デバイス上に 外部 VLANs が存在する。
- レイヤ 2 およびレイヤ 3 デバイスは障害ポイントではない。

```
ip route 0.0.0.0.0.0.0.0.192.168.20.100
ip redundancy
```

```
interface e2
  bridge vlan 1
  description "uplink VLAN"
```

```
interface e5
  bridge vlan 1
  description "uplink VLAN"
```

```
interface e9
  bridge vlan 3
  description "server VLAN"
```

```
interface e12
  bridge vlan 3
  description "server VLAN"
```

```
interface e1
```

## ■ ステートレス冗長フェールオーバーの設定

```
bridge vlan 2
description "Redundancy Protocol Heartbeat"
circuit VLAN1
  redundancy
  ip address 192.168.20.1 255.255.255.0

circuit VLAN3
  redundancy
  ip address 192.168.10.1 255.0.0.0

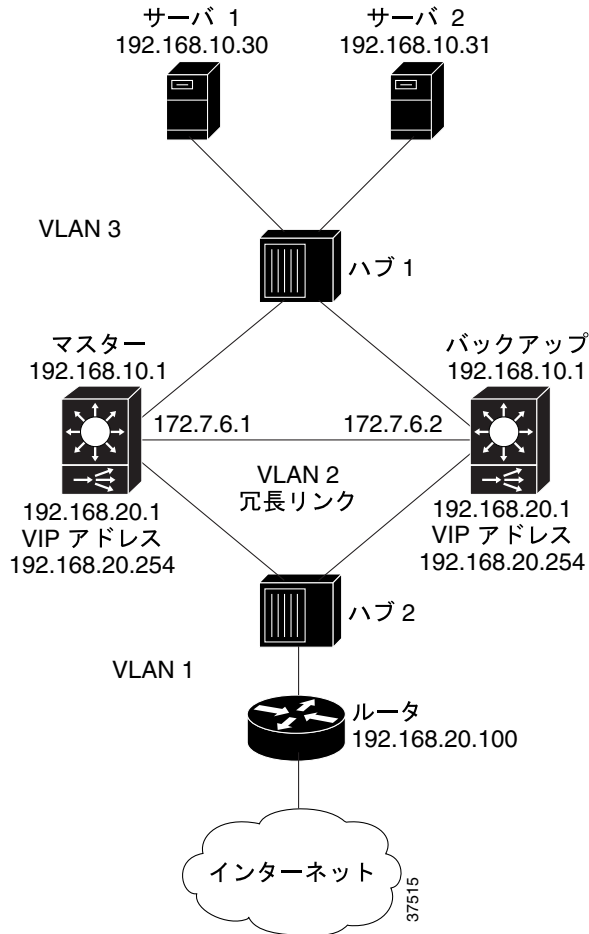
circuit VLAN2
  redundancy-protocol
  ip address 172.7.6.1 255.255.255.253

service s1
  ip address 192.168.10.30
  active
service s2
  ip address 192.168.10.31
  active

owner Redundant-Pool
  content web
  vip address 192.168.20.254
  protocol tcp
  port 80
  redundancy-l4-stateless
  add s1
  add s2
  balance srcip
  active

group Redundant-Pool
  vip address 192.168.20.254
  redundancy-l4-stateless
  add service s1
  add service s2
  active
```

図 3-3 ステートレス冗長フェールオーバーのためのボックスツボック冗長性の例



## VIP および仮想インターフェイス冗長性の設定

VIP および仮想インターフェイス冗長設定の詳細については、第 1 章「VIP および仮想インターフェイスの冗長性の設定」を参照してください。

### レイヤ 2 およびレイヤ 3 設定とコンバージェンス

仮想ルータ冗長プロトコル (VRRP) を運用している CSS が VLAN をシャットダウンすることはありません。したがって、VRRP 設定には同じ VLAN (ブリッジモード) 内のコンテンツ ルールの VIP、アップリンク、およびサーバのアドレスが含まれない可能性があります。その代わりに、冗長ペアの各 CSS がアップリンク VLAN とサーバ VLAN 間のルータとして機能するように CSS を設定する必要があります。CSS では、サーバのデフォルトのゲートウェイに対する仮想ルータアドレスが使用されます。

両方の CSS がアクティブで、共にトポロジプロトコルに関与しているため、障害発生時にはコンバージェンス時間が短くなることがあります。さらに、どちらの CSS も常にサーバとキープアライブトラフィックを取得するため、サーバの可用性に関して双方の CSS が所有する情報には、矛盾する点はありません。

VRRP では、VLAN からの冗長発信ルートパスが提供されますが、複数の VLAN を同期化することはできません。この制限があるため、一方の CSS がサーバ VLAN への接続のマスターになっている間は、もう一方の CSS はアップリンク VLAN のマスターにならないようにする必要があります。この分割状態を回避するために、CSS は VRRP の拡張実装の一部として、重要な外部 IP アドレスを監視できます。

通常、1 台の CSS に最高の最優先を設定し、VRID ペア (アップリンク VLAN 側とサーバ VLAN 側) に **preempt** オプションを設定します。これにより、その CSS が利用可能であれば両方の VRID がコンバージェンスされ、分割状態が回避されます。

より複雑な障害状況に対処するには、スクリプト キープアライブを使用します。スクリプト キープアライブの詳細については、『*Cisco Content Services Switch Administration Guide*』を参照してください。

次の例では、アップリンク ルータまたはすべての Web サーバとの通信が失われると、マスター CSS は両方の仮想インターフェイスの制御を放棄します。

## 設定例

次に示す設定例（[図 3-4](#) を参照）では、次の事項を前提としています。

- CSS は VIP および仮想インターフェイス冗長性設定で、2 つの外部 VLAN 間のルータとして動作する。
- レイヤ 2 およびレイヤ 3 デバイス上に 外部 VLANS が存在する。
- レイヤ 2 およびレイヤ 3 デバイスは障害ポイントではない。

```
ip route 0.0.0.0 0.0.0.0 192.168.20.100

interface e2
  bridge vlan 1
  description "uplink VLAN"

interface e5
  bridge vlan 1
  description "uplink VLAN"

interface e9
  bridge vlan 3
  description "server VLAN"

interface e12
  bridge vlan 3
  description "server VLAN"

circuit VLAN1
  ip address 192.168.20.1 255.255.255.0
  ip virtual-router 1 priority 110 preempt
  ip redundant-vip 1 192.168.20.254
  ip redundant-interface 1 192.168.20.2
  ip critical-service 1 uplink
  ip critical-service 1 s1
  ip critical-service 1 s2

circuit VLAN3
  ip address 192.168.10.1.0.0.0
  ip virtual-router 1 priority 110 preempt
  ip redundant-vip 1 192.168.10.254
  ip redundant-interface 1 192.168.10.2
  ip critical-service 1 uplink
  ip critical-service 1 s1
  ip critical-service 1 s2
```

## ■ ステートレス冗長フェールオーバーの設定

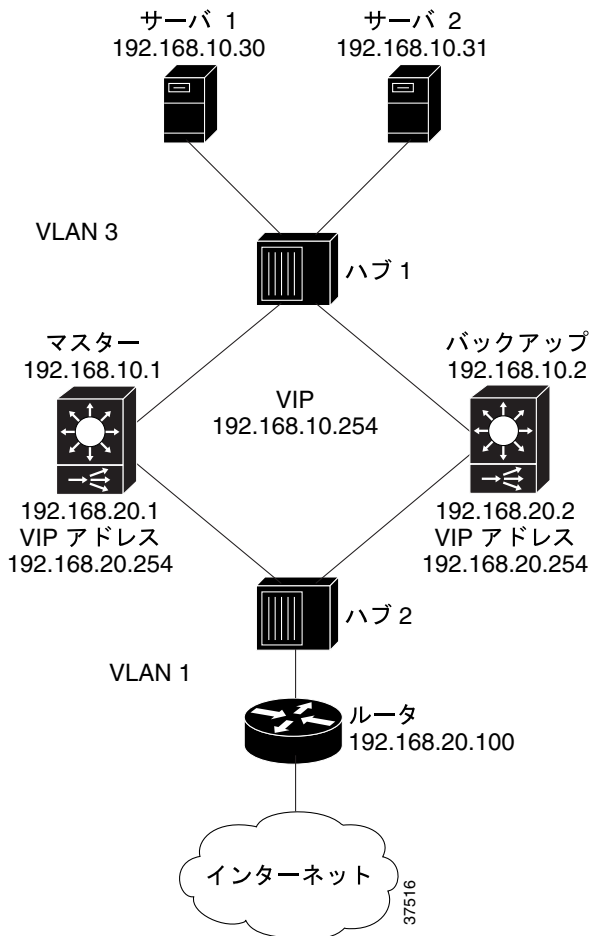
```
service uplink
  ip address 192.168.20.100
  type redundancy-up
  active

service s1
  ip address 192.168.10.30
  active
service s2
  ip address 192.168.10.31
  active

owner Redundant-Pool
  content web
  vip address 192.168.20.254
  protocol tcp
  port 80
  redundancy-l4-stateless
  add s1
  add s2
  balance srcip
  active

group Redundant-Pool
  vip address 192.168.20.254
  redundancy-l4-stateless
  add service s1
  add service s2
  active
```

図 3-4 ステートレス冗長フェールオーバーのための VIP および仮想インターフェイス冗長性設定の例



## 代替設定

ステートレス冗長フェールオーバーでは、別の設定とトポロジを採用することも可能です。他の高可用性環境でこの機能を使用する場合は、この章の他の項および第1章「VIP および仮想インターフェイスの冗長性の設定」で CSS 冗長性設定の詳細と例を参照してください。その他の情報については、RFC-2338「*Virtual Router Redundancy Protocol*」を参照してください。

## 設定管理

保守のためにサーバをオフラインにする必要がある場合は、冗長 CSS でも対応するサーバをオフラインにする必要があります。サーバファームの状態を同期しないと、保守期間中にフェールオーバーが発生したときに、接続を正しくマッピングできません。IP 冗長性設定でサービス状態を自動的に同期化させるには、設定同期スクリプト (`commit_redundancy`) を実行します。VIP/ インターフェイス冗長性設定については、手動でサービス状態を同期化させます。

## その他の留意事項

ステートレス冗長フェールオーバーでは、次の条件が適用されます。

- フェールオーバー後は、データチャネルの NAT 状態が失われるため、パッシングモード FTP は動作を継続できません。ただし、ポートモード FTP は継続して機能します。
- ソースグループのポートマップ機能は無効になっているため、冗長化でロードバランスされたファーム内のサーバから起動された接続には、発信元ポート変換は行われません。そのため、DNS などの機能に影響する場合があります。
- サービスレコードを設定して、ロードバランスされたトラフィックの送信先ポートを変更することはできません。
- クライアントとサーバ間の TCP/IP 接続には常にデータが流れている可能性があります。したがって、トポロジのコンバージ中や、一部のサービスがキープアライブトラフィックによって取得される前に着信したパケットは、正しく転送されるとは限りません。たとえば、サービスがキープアライブトラフィックへの応答を停止したにもかかわらず、対応する TCP 接続はその後も継続されることがありますが、バックアップ CSS はそのような接続の状態を認識できないため、接続を再開するときに推測を誤る可能性があります。



- 非常にクリティカルな環境では、接続損失割合とコンバージェンス時間の目安を設定します。その上で、さまざまなトポロジとトポロジプロトコルの組み合わせをテストし、目安とした接続損失割合とコンバージェンス時間を達成できることを確認してください。このテストでは、高可用性ネットワークで克服する必要がある発生可能なあらゆる障害状態を勘案する必要があります。トラフィックがクリティカルな場合、恒久的なテストベッドを作成して、新しい設定を配備する前にネットワーク コンポーネント システムを検証することもできます。

## 冗長性設定の表示

CSS 間の冗長性設定を表示するには、**show redundancy** コマンドを使用します。

たとえば、次のように入力します。

```
(config)# show redundancy
```

冗長性が設定されていない場合は、次のステータスが表示されます。

```
(config)# show redundancy
Redundancy: Disabled Redundancy Protocol: Not Running
```

**show redundancy** コマンドによる出力は、コマンドをマスター CSS とバックアップ CSS のどちらで実行したかによって異なります。

表 3-3 に、**show redundancy** コマンドで表示されるフィールドについて説明します。

**表 3-3 show redundancy コマンドのフィールド**

フィールド	説明
Redundancy	CSS で冗長性が有効かどうかを示します。
Redundancy Protocol	CSS で冗長プロトコルが実行されているかどうかを示します。
Redundancy State	CSS の現在の冗長状態 (Master または Backup)
MasterMode	CSS がマスターとして設定されているかどうかを示します。Yes は、 <b>ip redundancy master</b> コマンドにより CSS がマスターとして指定されていることを示します。No は、 <b>ip redundancy</b> コマンドにより CSS がマスターとして指定されていないことを示します。
Number of times redundancy state changed to Master/Backup	マスターとバックアップの間で CSS の状態が変更された回数
Redundancy interface	冗長インターフェイスのアドレス

表 3-3 show redundancy コマンドのフィールド (続き)

フィールド	説明
Current State Duration	CSS が現在の冗長状態 (マスターまたはバックアップ) になってからの経過期間
Last Fail Reason	前回 CSS の冗長性に障害が発生したときの理由
VRID	VRID (仮想ルータ ID)
Priority	仮想ルータのピアに対する優先度。デフォルトの優先度は 100 です。1 ~ 255 の整数を入力します。
Physical Link Failure Monitor	
Interface/State	<b>redundancy-phy</b> コマンドで設定されたインターフェイスとその状態のリスト <b>show</b> コマンドの出力は、インターフェイス ポートの番号順にソートされます。
Uplink Enabled	有効に設定されたサービス アップリンクの数
Number Alive	有効な (Up 状態) サービス アップリンクの数
Service Name/State	アップリンク サービスとその状態のリスト <b>show</b> コマンドの出力は、サービス インデックスの番号順にソートされます。

■ 冗長性設定の表示