



サービスの設定

この章では、コンテンツ サービスの設定方法について説明します。この章の内容は、特に指定のない限り、すべての CSS モデルに共通です。

- サービス設定のクイック スタート
- サービスの作成
- サービスへの IP アドレスの割り当て
- ポートの指定
- プロトコルの指定
- ドメイン名の指定
- HTTP リダイレクト文字列の指定
- HTTP 302 リダイレクト メッセージへの TCP FIN フラグまたは RST フラグの設定
- リダイレクト文字列やドメイン名への `http://` の追加
- 拡張ロード バランシング文字列の設定
- サービスの HTTP Cookie の設定
- 重みとグレースフル シャットダウンの設定
- サービス タイプの指定
- サービス アクセスの設定
- キャッシュのコンテンツ ルールのバイパス
- 透過キャッシュのためのネットワーク アドレス変換 (NAT) の設定
- キャッシュ ファームをバイパスするサービスの設定
- TCP 最大接続数の設定

- サービスのキープアライブの設定
- サービスのアクティブ化
- サービスの一時停止
- サービス設定の表示
- サービス統計情報カウンタのクリア

サービス、所有者、コンテンツ ルールの関係についての概要は、第 1 章「コンテンツ ロード バランシングの概要」を参照してください。

サービスでの HTTP 応答データ圧縮については、『*Cisco Content Services Switch SSL Configuration Guide*』を参照してください。



(注)

CSS は、アクティブ - バックアップ VIP 冗長と仮想インターフェイス冗長を実装した環境で、CSS 11500 シリーズのピア間の Adaptive Session Redundancy (ASR; 適応型セッションの冗長性) をサポートします。これにより、既存フローのステートフル フェールオーバーが可能になります。ASR 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。

サービス設定のクイック スタート

表 3-1 に、サービスの設定に必要な基本手順の概要を示します。それぞれの手順に、作業を実行するために必要な CLI コマンドも示します。CLI コマンドの各機能とすべてのオプションの詳細については、表 3-1 以降の項を参照してください。

表 3-1 サービス設定のクイック スタート

作業とコマンドの例

1. `config` と入力して設定モードに入ります。

```
# config
(config)#
```

2. サービスを作成します。サービスを作成すると、CLI は次に示すようなサービス モードに入ります。さらにサービスを追加するには、`service` コマンドを再度入力します。

```
(config)# service serv1
(config-service[serv1])#
(config-service[serv1])# service serv2
(config-service[serv2])#
```

3. IP アドレスを各サービスに割り当てます。この IP アドレスには、サーバの実際の IP アドレスを指定します。

```
(config-service[serv2])#
(config-service[serv2])# ip address 10.3.6.2
(config-service[serv2])# service serv1
(config-service[serv1])# ip address 10.3.6.1
```

4. 各サービスをアクティブにします。

```
(config-service[serv1])# active
(config-service[serv1])# service serv2
(config-service[serv2])# active
(config-service[serv2])# exit
```

5. (推奨) サービス情報を表示してサービスの設定を確認します。

```
(config-service[serv2])# show service summary
```

次の実行設定例は、表 3-1 のコマンドを入力した結果です。

```
!***** SERVICE *****
service serv1
  ip address 10.3.6.2
  active

service serv2
  ip address 10.3.6.1
  active
```

サービスの作成

サービスには、インターネットのコンテンツを格納および提供する宛先またはエンティティ（サーバ、FTP などのサーバ上のアプリケーション、またはストリーミング オーディオ）を指定できます。サービスには名前を付け、IP アドレス（オプションでプロトコルとポート番号）に関連付けます。

サービスを作成すると、サービスを識別したり、CSS にサービスを認識させることができます。コンテンツ ルールをサービスに適用すると、CSS は次の処理を実行できるようになります。

- サービスへのコンテンツ要求の送信
- サービスへのコンテンツ要求の拒否

サービス名は 1 ～ 31 文字で入力します。たとえば、サービス *serv1* を作成するには、次のように入力します。

```
(config)# service serv1
```

CSS は、新しく作成されたサービスのモードに変わります。

```
(config-service[serv1])#
```

既存のサービスを削除するには、次のように入力します。

```
(config)# no service serv1
Delete service <serv1>, [y/n]:y
```

サービスへの IP アドレスの割り当て

CSS でコンテンツ要求を適切なサービスに送信できるようにするには、IP アドレスまたは IP アドレスの範囲をサービスに割り当てる必要があります。サービスに IP アドレスを割り当てると、そのサービスが CSS で識別されます。CSS がコンテンツ要求を受信すると、VIP（および、ポートの場合もある）がサービスの IP アドレス（複数の場合もある）とポートに変換されます。

CSS では、サービスの VIP アドレスと IP アドレスは、A、B、C のいずれかのクラスに限定されます。マルチキャスト アドレス（クラス D および E）や、アドレス上限を超える範囲の IP アドレスは許可されません。

たとえば、IP アドレスを *serv1* に割り当てるには、次のように入力します。

```
(config-service[serv1])# ip address 172.16.1.1
```

サービスの IP アドレスをデフォルトの 0.0.0.0 に戻すには、次のように入力します。

```
(config-service[serv1])# no ip address
```



(注)

一部のサービスには、IP アドレスの割り当ては不要です。IP アドレスが不要なサービスは、次のとおりです。

- サービス タイプとして **ssl-accel** を指定して設定されたサービス
- サービス タイプとして **redirect** を指定して設定されたサービス
- **bypass-hosttag** コマンドで設定されたサービス

これらのサービスは、キープアライブ タイプを **none** に指定して設定する必要があります。

ip address コマンドで IP アドレスを指定するときに **range** オプションを使用すれば、そのアドレスから始まる IP アドレスの範囲を指定できます。1 ~ 65535 の数値を入力します。デフォルトは 1 です。たとえば、範囲を 10 に指定して 172.16.1.1 という IP アドレスを入力すると、IP アドレスの範囲は 172.16.1.1 ~ 172.16.1.10 になります。

■ サービスへの IP アドレスの割り当て

次に使用例を示します。

```
(config-service[serv1])# ip address 172.16.1.1 range 10
```

ip address range コマンドを使用する場合は、使用中のサブネット内の IP アドレスを使用します。CSS は、回線のサブネットにない IP アドレスは ARP 変換しません。たとえば、回線を 10.10.10.1/24 に設定し、VIP の範囲を 10.10.10.2 range 400 として設定した場合、10.10.10.254 より後の IP アドレスは ARP 変換されません。これと同じ例で、VIP の範囲を 200 に設定した場合は、この範囲のすべての IP アドレスが ARP 変換の対象になります。



(注)

CSS は、サービス範囲の最初のアドレスだけにキープアライブを送信します。キープアライブのスクリプトを設定する場合は、サービス範囲の先頭アドレスを、引数の 1 つとしてスクリプトに含める必要があります。

範囲内のいずれかのアドレス上のサービスに CSS から要求を転送するためには、CSS がサービス範囲内の先頭アドレスを正しく ARP 変換できることが前提になります。この処理はキープアライブに依存しません。

ポートの指定

サービスには、TCP または UDP の宛先ポート番号が関連します。サービスに TCP/UDP のポート番号またはポート番号の範囲を指定するには、**port** コマンドを使用します。ポート番号は 0 ～ 65535 の整数で入力します。デフォルトは 0 (any) です。

たとえば、次のように入力します。

```
(config-service[serv1])# port 80
```

キープアライブに使用するポートを指定するには、サービス モードの **keepalive port** コマンドを使用してください。

port コマンドを使用して指定したポート番号から開始するポート番号の範囲を指定するには、**range** オプションを使用します。範囲は 1 ～ 65535 の数値で入力します。デフォルトは 1 です。たとえば、範囲を 10 に設定してポート番号を 80 に指定すると、ポート番号の範囲は 80 ～ 89 になります。**port range** コマンドは、ローカル (デフォルト) のサービスでだけ使用できます。

たとえば、次のように入力します。

```
(config-service[serv1])# port 80 10
```

ポートをデフォルトの 0 (any) に設定するには、次のように入力します。

```
(config-service[serv1])# no port
```

プロトコルの指定

デフォルト設定では、CSS はサービスにあらゆる IP プロトコルを使用します。サービスが使用するサービス IP プロトコルを指定するには、**protocol** コマンドを使用します。このコマンドには次のオプションがあります。

- **protocol tcp** : サービスは TCP プロトコルスイートを使用する。
- **protocol udp** : サービスは UDP プロトコルスイートを使用する。

次に使用例を示します。

```
(config-service[serve1])# protocol tcp
```

プロトコルをデフォルトの **any** に設定するには、次のように入力します。

```
(config-service[serve1])# no protocol
```


ドメイン名の指定

CSS は、ここで設定するドメイン名を、要求されたコンテンツの新しい場所としてリダイレクトメッセージ内で使用します。CSS は要求された URL の前に、このドメイン名を追加します。このドメイン名が設定されていない場合、CSS は元の要求の `host-tag` フィールドにあるドメイン名を使用します。ホストタグが見つからない場合は、サービスの IP アドレスにリダイレクトします。HTTP リダイレクト サービスが「302 object moved」というメッセージを生成するときに、要求されたコンテンツの前に追加するドメイン名を指定するには、**domain** コマンドを使用します。



(注) サービスのリダイレクト ドメインは、**redirect** に設定されたサービスタイプだけに使用できます。リダイレクト サービスが適切な HTTP リダイレクト先を取得できるようにするには、**domain** コマンドを指定する必要があります。



(注) 同じサービスに、**domain** コマンドと **(config-service) redirect-string** コマンドを同時に設定することはできません。



(注) **redirect-string** コマンドと **(config-service) domain** コマンドは似ています。ただし、**redirect-string** コマンドでは、設定されている文字列が返されます。一方 **(config-service) domain** コマンドでは、最初に要求された URL の前にドメイン名が追加されます。

サービスのドメイン名は、スペースを含まない 64 文字以内のテキスト文字列を引用符で囲まずに入力します。



(注) ドメイン名の前には `http://` が自動的に追加されます。

■ ドメイン名の指定

たとえば、次のように入力します。

```
(config-service[serve1])# domain www.arrowpoint.com
```

次の入力も有効です。

```
(config-service[serve1])# domain 172.16.3.6
```

このサービスのリダイレクトドメインを消去するには、次のように入力します。

```
(config-service[serve1])# no domain www.arrowpoint.com
```

次の入力も有効です。

```
(config-service[serve1])# no domain 172.16.3.6
```

HTTP リダイレクト文字列の指定

CSS は、リダイレクト文字列全体を、要求されたコンテンツの新しい場所として使用します。リダイレクト文字列が設定されていない場合、CSS は **(config-service) domain** コマンドで設定したドメイン名を、最初に要求された URL の前に追加します。リダイレクト文字列かドメイン名のいずれも設定されていない場合、CSS は元の要求の **host-tag** フィールドの URI のドメイン名と要求の HTTP コンテンツを併せて使用します。ホスト タグが見つからない場合は、サービスの IP アドレスを使用して、リダイレクトを生成します。

HTTP リダイレクト サービスで「302 object moved」のようなメッセージを生成させる場合、CSS が使用する HTTP リダイレクト文字列を指定するには、**redirect-string** コマンドを使用します。リダイレクト文字列は、タイプが **redirect** であるサービスに限り設定できます。



(注)

redirect-string コマンドと **domain** コマンドは似ています。CSS は、設定されている **redirect-string** コマンド文字列をそのまま返します。**redirect-string** コマンドを使用する代わりに、サービスで **domain** コマンドを設定すると、設定されたドメイン名が、要求された元の URL の前に追加されます。同一のサービスで **redirect-string** と **domain** コマンドの両方を設定することはできません。

このサービス モードのコマンドのシンタックスは次のとおりです。

redirect-string *string*

HTTP リダイレクト文字列は、252 文字以内の、テキスト文字列を引用符で囲んで、またはスペースを含まないテキスト文字列を引用符で囲まずに入力します。

次に入力例を示します。

```
(config-service[serve1])# redirect-string www.arrowpoint.com
```

サービスからリダイレクト文字列を削除するには、次のように入力します。

```
(config-service[serve1])# no redirect-string www.arrowpoint.com
```

HTTP 302 リダイレクトメッセージへの TCP FIN フラグまたは RST フラグの設定

CSS は HTTP 302 リダイレクトメッセージを送信するときに、デフォルトでは最初の接続で FIN フラグを送信し、固定接続の存続期間中の後続要求で RST フラグを送信します。ただし、クライアントに送信するパケットに Microsoft IE ブラウザへのリダイレクトメッセージが含まれている場合には、このデフォルトのフラグ設定はブラウザに適さないことがあります。リダイレクトメッセージとともに送信するフラグを指定するには、グローバル設定モードで **http-redirect-option** 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。

http-redirect-option [fin-rst|fin-fin|rst-rst]

各キーワードの意味を次に示します。

- **fin-rst** : デフォルトの動作。CSS は最初の接続で FIN フラグを送信し、固定接続の存続期間中の後続要求では RST フラグを送信します
- **fin-fin** : CSS は常に FIN フラグを送信します。
- **rst-rst** : CSS は常に RST フラグを送信します。

たとえば、302 リダイレクトメッセージとともに、常に FIN フラグが送信されるように設定するには、次のように入力します。

```
(config)# http-redirect-option fin-fin
```

デフォルトの動作にリセットするには、次のように入力します。

```
(config)# http-redirect-option fin-rst
```

HTTP リダイレクトメッセージのフラグ設定は、**show http-redirect-option** コマンドで確認できます。次のように入力します。

```
(config)# show http-redirect-option
HTTP Redirect Option: fin-rst
```

リダイレクト文字列やドメイン名への http:// の追加

デフォルトでは、CSS はリダイレクト文字列やドメイン名の先頭に `http://` を自動的に追加します。サービスに設定されたリダイレクト文字列またはドメイン名の前に `http://` が追加されないようにするには、次のように入力します。

```
(config-service[serv1])# no prepend-http
```

サービスに設定されたリダイレクト文字列またはドメイン名の先頭に `http://` を追加するには、`prepend-http` 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-service[serv1])# prepend-http
```

拡張ロード バランシング文字列の設定

スティッキ性のあるコンテンツ ルールには、拡張ロード バランシング方式を指定できます。コンテンツ ルールにスティッキ性が生じるのは、最初のセッションの接続と同じサービスに、同じユーザやクライアントからの追加セッションが送信され、通常のロード バランシングが無効になったときです。デフォルトでは、拡張バランシング方式は無効です。

サービスに拡張ロード バランシング文字列を指定するには、`string` コマンドを使用します。このコマンドは、拡張ロード バランシング メソッド `url`、`cookie`、または `cookieurl` とともに使用します。拡張ロード バランシング方式の詳細については、[第 11 章「コンテンツ ルールへのスティッキ パラメータの設定」](#)を参照してください。

1 ～ 15 文字以内の文字列を入力します。たとえば、次のように入力します。

```
(config-service[serv1])# string 172.16.3.6
```

サービスから文字列を削除するには、次のように入力します。

```
(config-service[serv1])# no string
```

サービスの HTTP Cookie の設定

スティッキ性のあるコンテンツ ルールには、拡張ロード バランシング方式を指定できます。コンテンツ ルールにスティッキ性が生じるのは、最初のセッションの接続と同じサービスに、同じユーザやクライアントからの追加セッションが送信され、通常のロード バランシングが無効になったときです。デフォルトでは、拡張バランシング方式は無効です。

拡張バランス方式の **cookies**、**url**、または **cookieurl** を使用して文字列を完全一致させる場合は、各サーバに一致させる一意の文字列を設定する必要があります。サービスに HTTP Cookie を指定するには、**string** コマンドを使用します。このサービス モードのコマンドのシンタックスは次のとおりです。

string *cookie_name*

cookie_name には、スペースを含まない 15 文字以内のテキスト文字列を引用符で囲まずに入力します。

たとえば、次のように入力します。

```
(config-service[serv1])# string userid3217
```

サービスから Cookie を削除するには、次のように入力します。

```
(config-service[serv1])# no string
```

CSS でのスティッキの設定については、[第 11 章「コンテンツ ルールへのスティッキ パラメータの設定」](#) を参照してください。

重みとグレースフル シャットダウンの設定

コンテンツ ルールで ACA または加重ラウンドロビン ロード バランシング方式を設定している場合に、この重みが使用されます。デフォルトでは、すべてのサービスの重みは1に設定されます。重みを大きくすると、特定のサービスにフローが偏ります。サーバの重みに基づく ACA ロード バランシング判定の背景情報については、第6章「サービスの負荷の設定」の「サーバの負荷と重みに基づいた ArrowPoint Content Awareness (ACA) の使用」を参照してください。

サービスの相対的な重みを指定するには、サービス モードで **weight** コマンドを使用します。サービスの重みを設定するには、0 ~ 10 の数値を入力します。デフォルトは1で、0はグレースフル シャットダウンです。

過負荷サービスのグレースフル シャットダウンを行う場合、または、メンテナンスのためにサービスをオフラインにする場合は、重み0を指定すると、新しい接続がそのサービスに送信されません。ただし、接続中のスティッキ セッションを除きます。時間が経過して、存在するスティッキセッションが完了すると、そのサービスの負荷は小さくなり始めます。重みを0から他の値(1 ~ 10)に変更すると、すべてのロード バランシング方式で、そのサービスがローテーションに再度組み込まれます。



(注)

CSS に絶対負荷計算方式を設定し、サービスの重みをゼロに設定している場合、CSS がこのサービスの負荷を、アドバタイズするコンテンツ ルールの負荷に含めることはありません。

この CSS は、ロケーションクッキーを含むコンテンツ要求を、スティッキセッションの一部として認識します。したがって、重みを0に設定したサービスをロケーション サービスとしてコンテンツ ルールに追加しても、そのサービスから発信されたロケーションクッキーを含むすべての要求は、引き続きそのサービスに送信されます。

たとえば、重みを2に設定するには、次のように入力します。

```
(config-service[serve1])# weight 2
```

■ 重みとグレースフル シャットダウンの設定

重みを 0 に設定して特定サービスのグレースフルシャットダウンを行うには、次のように入力します。

```
(config-service[serv1])# weight 0
```

重みをデフォルトの 1 に戻すには、次のように入力します。

```
(config-service[serv1])# no weight
```

weight コマンドを使用してサービスに重みを設定した場合で、設定済みの Dynamic Feedback Protocol (DFP) エージェントがそのサービスに存在する場合は、DFP エージェントに設定された重みが、サービスに設定された重みより優先されます (加重ラウンドロビンのロード バランシング方式のみ)。

コンテンツ ルールに 1 つ、またはそれ以上のサービスを追加すると、CSS はサービス モードで設定されているサービスの重みを、サーバ固有のアトリビュートとして各ルールに適用します。コンテンツ ルール固有のサーバ重みを指定するには (コンテンツ ルールは加重ロード バランシング方式を使用していると仮定)、**add service** コマンドを **weight** オプション付きで使用します。これらのコマンドを使用すると、サーバ固有の重みが無効になり、サービスを追加したコンテンツ ルールだけに適用されます。**add service** コマンドの詳細については、[第 10 章「コンテンツ ルールの設定」](#)の「[サービスの重みの指定](#)」を参照してください。

サービス タイプの指定

デフォルトのサービス タイプは **local** です。サービスのタイプを指定するには、**type** コマンドを使用します。このサービス モード コマンドのシンタックスおよびオプションは、次のとおりです。

- **type nci-direct-return** : サービスをダイレクト リターン用の NAT チャネル通知に指定する。



(注) NAT ピアリングの設定には、**type nci-direct-return** コマンドを使用します。NAT ピアリングの詳細については、[第13章「キャッシングの設定」](#)を参照してください。

- **type nci-info-only** : サービスを読み取り専用の NAT チャネル通知に指定する。
- **type proxy-cache** : サービスをプロキシ キャッシュとして定義する。これは、キャッシュ固有のオプションです。このオプションを使用すると、キャッシュ サーバから着信する要求へのコンテンツ ルールの適用がバイパスされます。この場合、コンテンツ ルールをバイパスすることで、キャッシュと CSS 間のループが回避されます。プロキシ キャッシュの詳細については、[第13章「キャッシングの設定」](#)を参照してください。
- **type redirect** : サービスをリモート サービスとして定義して、ローカル サービスが使用できないとき（ローカル サービスの負荷が設定済みの負荷しきい値を超えた場合など）に、CSS がリモート サービスにコンテンツ要求をリダイレクトできるようにする。コンテンツ ルールに負荷しきい値を設定するには、所有者コンテンツ モードで **load-threshold** コマンドを使用します（[第10章「コンテンツ ルールの設定」](#)の「[負荷しきい値の指定](#)」を参照）。**type redirect** として定義されているリモート サービスが複数ある場合は、CSS はラウンドロビン ロード バランシング方式を使用して、それらのリモート サービス間で要求の負荷を分散します。

コンテンツ ルールにリダイレクト タイプのサービスを追加する場合は、コンテンツと一致する URL を設定する必要があります。たとえば、`/*` または `/vacations.html` のように設定します。



(注) 従来のバージョンでは、リダイレクトタイプのバックエンドサービスが存在する場合、DNS 要求で VIP アドレスを要求してもサービスの IP アドレスが返されました。バージョン 7.30.1 以降では、VIP アドレスが返されるように変更されました。

- **type redundancy-up** : 冗長用アップリンクのルータ サービスを指定する。
- **type rep-cache-redis** : サービスを、リダイレクトを伴うレプリケーション キャッシュとして指定する。
- **type rep-store** : サービスをレプリケーションストアとして指定する。
- **type rep-cache-redis** : サービスを、リダイレクトを伴うレプリケーションストアとして指定する。このサービスタイプからの要求には、コンテンツルールが適用されません。

- **type ssl-accel** : SSL アクセラレーション モジュール用の SSL 終了サービスに指定する。このタイプでは、次の処理ができます。

- サービスを SSL 終了サービスとして設定する。
- SSL プロキシリストを、**(config-service) add ssl-proxy-list** コマンドで SSL 終了サービスとして追加する。

SLL 終了の設定については、『*Cisco Content Services Switch SSL Configuration Guide*』を参照してください。

- **type ssl-accel-backend** : バックエンド SSL サーバ用のサービスに指定する。このタイプでは、次の処理ができます。

- サービスをバックエンド SSL サービスとして設定する。
- SSL プロキシリストを、**(config-service) add ssl-proxy-list** コマンドでバックエンド SSL サービスとして追加する。

バックエンド SLL サーバの設定については、『*Cisco Content Services Switch SSL Configuration Guide*』を参照してください。

- **type ssl-init** : SSL 開始サーバ用のサービスに指定する。このタイプでは、次の処理ができます。

- サービスを SSL 開始サービスとして設定する。
- SSL プロキシリストを、**(config-service) add ssl-proxy-list** コマンドで SSL 開始サービスとして追加する。

SLL 開始サーバの設定については、『*Cisco Content Services Switch SSL Configuration Guide*』を参照してください。

- **type transparent-cache** : サービスを透過キャッシュとして指定する。これは、キャッシュ固有のオプションです。このサービス タイプからの要求には、コンテンツ ルールが適用されません。この場合、コンテンツ ルールをバイパスすることで、キャッシュと CSS 間のループが回避されます。透過キャッシュについては、第13章「キャッシングの設定」を参照してください。

たとえば、CSS で serv1 のコンテンツ要求をリダイレクトできるようにするには、次のように serv1 のコンテンツ ルールで **redirect** を指定します。

```
(config-service[serv1])# type redirect
```

サービス タイプをデフォルト設定の **local** に戻すには、次のように入力します。

```
(config-service[serv1])# no type
```

CSS の各サーバタイプへのアクセス方法

レイヤ3または4のコンテンツ ルールを設定すると、ルールはローカル サービスにヒットします。ただし、次の例外があります。

- ローカル サービスが動作していないか未設定の場合、ルールはプライマリ ソーリー サーバにヒットする。
- プライマリ ソーリー サーバで障害が発生した場合、ルールはセカンダリ ソーリー サーバにヒットする。

レイヤ3またはレイヤ4のルールでは、HTTP プロトコルを使用するため、リダイレクト サービスとリダイレクト コンテンツの文字列は使用できません。

レイヤ5のコンテンツ ルールを設定すると、CSS はコンテンツ要求をローカル サービスに送出します。ただし、次の例外があります。

- ローカル サービスが動作していないか未設定の場合、ルールはリダイレクト サービスへの場所が指定されている HTTP リダイレクトをクライアントへ送信する。
- ローカル サービスおよびリダイレクト サービスが動作していない場合、または未設定の場合、ルールは HTTP 要求をプライマリ ソーリー サーバに転送する。
- セカンダリ ソーリー サーバ以外のすべてのサービスが停止している場合、ルールは HTTP 要求をセカンダリ ソーリー サーバに転送する。

コンテンツ ルールへのサービスの追加またはプライマリおよびセカンダリ ソーリー サーバの追加については、第10章「[コンテンツ ルールの設定](#)」を参照してください。

サービス アクセスの設定

パブリッシング サービスを提供するサービスは、パブリッシング、サブスクライビング、およびオンデマンド レプリケーションのアクティビティの実行中にコンテンツを移動できるように、FTP アクセス メカニズムと関連付ける必要があります。FTP アクセス メカニズムとサービスを関連付けるには、**access** コマンドを使用します。このコマンドは、パブリッシング サービスを提供するサービスごとに使用する必要があります。このコマンドは、サブスクライバ サービスではオプションになります。サブスクライバ サービスは、パブリッシャからアクセス メカニズムを継承します。

このコマンドを使用して、FTP アクセス メカニズムをサービスに関連付ける場合は、既存の FTP レコードのベース ディレクトリがツリーのルートになります。WWW デーモンと FTP デーモン間のマッピングの一貫性を維持するためには、FTP アクセスのベース ディレクトリを、クライアントが参照する WWW デーモンのルート ディレクトリと対応させます。FTP レコードの作成については、『*Cisco Content Services Switch Administration Guide*』の **(config) ftp-record** コマンドの説明を参照してください。

既存の FTP レコードの名前は、スペースを含まないテキスト文字列を引用符で囲まずに入力します。

たとえば、次のように入力します。

```
(config-service[serv1])# access ftp arrowrecord
```

サービスのアクセス メカニズムを削除するには、次のように入力します。

```
(config-service[serv1])# no access ftp
```

キャッシュのコンテンツ ルールのバイパス

デフォルトでは、プロキシ キャッシュまたは透過キャッシュ タイプのサービスからの要求にコンテンツ ルールは適用されません。要求の処理時に、プロキシ キャッシュまたは透過キャッシュ タイプのサービスから発信された要求に対して、コンテンツ ルールが適用されないようにするには、**cache-bypass** コマンドを使用します。



(注)

プロキシ キャッシュおよび透過キャッシュについては、[第13章「キャッシングの設定」](#)を参照してください。

次のように入力します。

```
(config-service[serve1])# cache-bypass
```

プロキシ キャッシュまたは透過キャッシュ タイプのサービスからの要求にコンテンツ ルールを適用するには、次のように入力します。

```
(config-service[serve1])# no cache-bypass
```

透過キャッシュのためのネットワーク アドレス変換 (NAT) の設定

デフォルトでは、透過キャッシュ タイプのサービスに対する宛先 Network Address Translation (NAT; ネットワーク アドレス変換) は無効になっています。透過キャッシュ タイプのサービスに対して宛先 NAT を有効にするには、**transparent-hosttag** コマンドを使用します。

現時点では、**transparent-hosttag** コマンドは、Client Side Accelerator (CSA; クライアント側アクセラレータ) 環境で CSS が動作している場合だけに使用します。CSA 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。



(注)

透過キャッシュについては、[第13章「キャッシングの設定」](#)を参照してください。

次にコマンドの使用例を示します。

```
(config-service[serv1])# transparent-hosttag
```

透過キャッシュ タイプのサービスを対象に宛先 NAT を無効にするには、次のように入力します。

```
(config-service[serv1])# no transparent-hosttag
```

キャッシュ ファームをバイパスするサービスの設定

デフォルトでは、キャッシュに格納できないコンテンツは、キャッシュをバイパスします。CSS の CSA で、キャッシュ ファームをバイパスして本来の宛先サーバとの接続を確立し、キャッシュできないコンテンツを取得するには、**bypass-hosttag** コマンドを使用します。CSA で本来の宛先の IP アドレスを検索するには、**host tag** フィールドに示されているドメイン名を使用します。

現時点では、**bypass-hosttag** コマンドは、CSA 環境で CSS が動作している場合だけに使用します。CSA 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。

次にコマンドの使用例を示します。

```
(config-service[ serv1])# bypass-hosttag
```

キャッシュできないコンテンツに対してキャッシュのバイパスを無効にするには、次のように入力します。

```
(config-service[ serv1])# no bypass-hosttag
```

TCP 最大接続数の設定

デフォルトでは、1つのサービスとの間で確立できる TCP 接続数は制限されていません。サービスの TCP 最大接続数を定義するには、**max connections** コマンドを使用します。最大接続数を 6 ~ 65534 の範囲内で入力します。デフォルトは 65534 です。これは接続数に制限が無いことを意味します。



(注)

CSS に絶対負荷計算方式を設定している場合、その CSS は接続数が設定済みの最大値を超過したサービスの負荷を、アドバタイズするコンテンツ ルールの負荷にはいっさい含めません。

たとえば、次のように設定します。

```
(config-service[serve1])# max connections 7
```

TCP の最大接続数をデフォルト値に設定するには、次のように入力します。

```
(config-service[serve1])# no max connections
```



(注)

UDP コンテンツ ルールには、サービスの最大接続数は使用しないでください。UDP はコネクションレス型プロトコルであるため、サービス接続カウンタはゼロのまま増加しません。

サービスのキープアライブの設定

デフォルトのサービス キープアライブは ICMP で、キープアライブ間隔と再試行間隔はどちらも 5 秒、最大失敗回数は 3 回です。サービスのキープアライブの設定については、[第4章「サービス キープアライブ、グローバル キープアライブ、スクリプト キープアライブの設定」](#)を参照してください。

サービスのアクティブ化

サービスを設定したら、CSS がサービスにアクセスしてコンテンツ要求を取得できるようにサービスをアクティブにする必要があります。サービスをアクティブにすると、そのサービスは、コンテンツ要求のロード バランシングのためのリソース プールに格納され、キープアライブ機能が起動します。



(注)

サービスがアクティブになると、アクティブなサービスは **ip address**、**port**、**protocol**、**type**、**transparent-hosttag**、および **bypass-hosttag** コマンドでは変更できなくなります。アクティブなサービスを変更する必要がある場合は、まずそのサービスを一時停止してください。

サービス *serv1* をアクティブにするには、次のコマンドを入力します。

```
(config-service[serv1])# active
```



(注)

CSS は、シャーシ内の SSL アクセラレーション モジュールごとに 1 つのアクティブな SSL サービス (1 つのスロットにつき 1 つの SSL サービス) をサポートします。1 つのスロットに対して複数の SSL サービスを設定できますが、アクティブにできるのは一度に 1 つの SSL サービスだけです。このサービスをアクティブにするには、SSL プロキシ リストを **ssl-accel** タイプのサービスに追加してから、その SSL プロキシ リストをアクティブにする必要があります。

コンテンツ ルールの追加の詳細については、[第 10 章「コンテンツ ルールの設定」](#)を参照してください。ソース グループにサービスを追加する方法については、[第 5 章「サービスのソース グループの設定」](#)を参照してください。

サービスの一時停止

サービスを一時停止すると、そのサービスはコンテンツ要求のロード バランシングのためのプールから削除されます。サービスを一時停止しても既存のコンテンツのフローには影響はありませんが、新たに接続を確立してサービスにアクセスしてもそのコンテンツを取得することはできません。サービスのメンテナンスを実行するために、サービスを一時停止することができます。サービス *serv1* を一時停止するには、次のコマンドを入力します。

```
(config-service[serv1])# suspend
```



(注)

サービスを一時停止すると、フェールオーバー設定を使用して残りのサービスの間で負荷が再調整されます。

サービス設定の表示

サービスをアクティブにする前に、サービス設定を表示して、すべてのパラメータが適切であるか確認することができます。**show service** コマンドを実行すると、コマンドを実行する場所に応じて、CSS に現在設定されている特定のサービスまたはすべてのサービスに関する情報を表示することができます。

次の **show service** コマンドはどのモードからも実行できます。

- **show service** : 各サービスの設定を表示する。
- **show service service_name** : 特定のサービスに関するサービス情報を表示する。
- **show service summary** : 各サービスのサマリーを表示する。

特定のサービス モードから、**show service** コマンドを実行すると、そのサービスだけの設定情報が表示されます。たとえば、次のように入力します。

```
(config-service[serv1])# show service
```

このコマンドを他のモードで実行すると、すべてのサービスの設定情報が表示されます。

特定のサービスの情報だけを表示する場合は、**show service** コマンドとサービス名を入力します。たとえば、次のように入力します。

```
# show service serv86
```

現在設定されているすべてのサービスのサマリーを表示するには、**show service summary** コマンドを使用します。



(注)

show service コマンドで表示される接続カウンタは、UDP フローについてはゼロのまま増加しません。UDP は、コネクションレス型プロトコルです。

表 3-2 に、`show service` コマンドの各出力フィールドと、その説明を示します。

表 3-2 `show service` コマンドの出力フィールド

フィールド	説明
Name	サービスの名前
Index	CSS によって割り当てられた一意の数値インデックス
Type	<p>サービスのタイプ。サービスのタイプを定義しない場合、デフォルトのサービス タイプは <code>local</code> になります。タイプは次のとおりです。</p> <ul style="list-style-type: none"> • nci-direct-return : NAT ピアリング用の NAT channel indication (NCI; NAT チャンネル通知) サービス • nci-info-only : サービスは情報用の NAT チャンネル通知 • proxy-cache : サービスはプロキシ キャッシュ。このタイプでは、キャッシュからの要求のコンテンツ ルールをバイパスします。 • redirect : サービスには直接アクセスできず、リダイレクトする必要がある。 • redundancy-up : サービスは冗長アップリンク • rep-cache-redir : サービスは、リダイレクトによるレプリケーション キャッシュ • rep-store : サービスは、ホスト コンテンツ用のレプリケーションストア サーバ • rep-store-redir : サービスは、コンテンツ要求がリダイレクトされるレプリケーションストア • ssl-accel : サービスは、SSL アクセラレーション モジュール用の SSL アクセラレーション サービス • ssl-accel-backend : サービスは SSL バックエンド サーバ • ssl-init : サービスは SSL 開始サービス • transparent-cache : サービスは透過キャッシュ。コンテンツ ルールはキャッシュからの要求に適用されません。

表 3-2 show service コマンドの出力フィールド（続き）

フィールド	説明
State	サービスの状態。このフィールドには、サービスが Alive、Dying、Down、または Suspended のいずれかの状態であることが示されます。Dying 状態は、サービスで障害が発生していることを、サービス モード コマンドで設定されているパラメータ keepalive retryperiod 、 keepalive frequency 、および keepalive maxfailure に従って示します。サービスが Down 状態に入ると、CSS は新しい接続をサービスには転送しなくなります（サービスは、コンテンツ ルールのロード バランシングのローテーションから削除されます）。ただし、CSS はサービスへの既存接続をすべて維持します（サービスへの接続は「切断」されません）。
Rule	サービスのアドレス、プロトコル、およびポート情報
Redirect Domain	HTTP リダイレクト サービスがそのサービスの「OBJECT MOVED」というメッセージを生成する際に使用するドメイン名。
Session Redundancy	サービスに対して ASR が有効か無効かを示す。ASR の詳細については、『 <i>Cisco Content Services Switch Global Server Load-Balancing Configuration Guide</i> 』を参照してください。
SSL-Accel Slot	SSL モジュールが置かれている CSS のスロット。SSL サービスでは、SSL プロキシリストを特定の SSL モジュールに関連づけるために、SSL モジュールのスロット番号が必要です。SSL の詳細については、『 <i>Cisco Content Services Switch SSL Configuration Guide</i> 』を参照してください。
Session Cache Size	サービスの SSL セッション ID のキャッシュ サイズ。このキャッシュ サイズは、SSL モジュール上のそのセッション用のキャッシュに格納できる SSL セッション ID の最大数です。
Redundancy Global Index	redundant-index コマンドをサービス設定モードで使用してサービスに割り当てた ASR の一意なグローバル インデックス値
Redirect String	HTTP リダイレクト サービスがそのサービスの「OBJECT MOVED」というメッセージを生成する際に使用する HTTP リダイレクト文字列

表 3-2 show service コマンドの出力フィールド (続き)

フィールド	説明
Keepalive	<p>キープアライブのタイプ、間隔、最大失敗回数、および再試行間隔。キープアライブのタイプは次のとおりです。</p> <ul style="list-style-type: none"> • ftp : FTP サーバにログインして、FTP サーバにアクセスするための、FTP レコード ファイルで定義されたキープアライブ メソッド • http : HTTP インデックス ページ要求。デフォルトでは、HTTP キープアライブは固定接続を使用します。HTTP Head キープアライブでは、応答コードも表示されます。 • icmp : ICMP エコー メッセージ (デフォルト) • named : キープアライブ設定モードで定義されたグローバル キープアライブ • none : キープアライブ メッセージはサービスに送信されない • script : サービスが使用するスクリプト キープアライブ。スクリプトは、キープアライブが実行されるたびに実行されます。 • ssl : このサービスの SSL HELLO キープアライブ。このキープアライブは、SSL をサポートするすべてのバックエンドサービスで使用します。CSS で SSL モジュールを使用する場合は、キープアライブ タイプとして none を使用します。 • tcp : TCP 接続のハンドシェイク要求 <p>キープアライブ間隔の値は、再試行メッセージをサービスに送信する時間間隔 (秒) です。デフォルトは 5 です。範囲は 2 ~ 255 です。</p> <p>キープアライブ 最大失敗回数は、サービスが「down」とみなされる前にキープアライブ メッセージの応答に失敗できる回数です。デフォルトは 3 です。範囲は 1 ~ 10 です。</p> <p>キープアライブ再試行間隔の値は、再試行メッセージをサービスに送信する時間間隔 (秒) です。デフォルトは 5 です。範囲は 2 ~ 255 です。</p>

表 3-2 show service コマンドの出力フィールド (続き)

フィールド	説明
Keepalive Encrypted	HTTP キープアライブの SSL 暗号化状態。値は Enabled (有効) または Disabled (無効) です。
Last Clearing of Stats Counters	State Transitions、Total Connections、Total Reused Conns カウンタのいずれかをクリアした (0 にリセットした) 最後の日時。最初に表示される日時のタイムスタンプは、サービスがアクティブになった時点を示します。サービスが停止している場合は「01/01/00 00:00:00」になります。
Mtu	サービスで送受信できるデータグラムの最大サイズ
State Transitions	サービスの状態遷移の合計数。このフィールドが 0 の場合、グローバル設定モードの zero service state-transitions コマンドまたはコンテンツ モードの zero state-transitions コマンドで、カウンタがリセットされた可能性があります。また、サービスが停止している場合または、サービスが起動中でトラフィックがない場合にも、カウンタは 0 になります。
Total Local Connections	ASR 設定の CSS マスター側の TCP 接続の総数
Current Local Connections	ASR 設定の CSS で現在アクティブな TCP 接続の数
Total Backup Connections	ASR 設定のマスター CSS でバックアップされる TCP 接続の総数
Current Backup Connections	ASR 設定で CSS が現在バックアップしている TCP 接続の数
Total Connections	サービスにマッピングされた接続の合計数。適応型セッションの冗長設定では、接続の総数は、ローカル接続の総数とバックアップ接続の総数の合計です。このフィールドが 0 の場合、グローバル設定モードの zero service total-connections コマンドまたはコンテンツ モードの zero total-connections コマンドで、カウンタがリセットされた可能性があります。また、サービスが停止している場合または、サービスが起動中でトラフィックがない場合にも、カウンタは 0 になります。
Max Connections	サービスに設定できる TCP の最大接続数

表 3-2 show service コマンドの出力フィールド (続き)

フィールド	説明
Total Reused Conns	固定接続中に複数のコンテンツ要求で繰り返し使用された合計接続数。このフィールドが 0 の場合、グローバル設定モードの zero service total-reused-connections コマンドまたはコンテンツモードの zero total-reused connections コマンドでカウンタがリセットされた可能性があります。また、サービスが停止している場合または、サービスが起動中でトラフィックがない場合にも、カウンタは 0 になります。
Weight	負荷の割り当てを決定するために負荷メトリックとともに使用されるサービスの重み。重みは ArrowPoint Content Awareness (ACA) および加重ラウンドロビン ロード バランシング方式で使用されます。範囲は 1 ~ 10 です。デフォルトは 1 です。
Load/Average Load	サービスの現在の負荷と平均負荷
Weight Reporting	サービスの重みの報告の方式。このサービスに Dynamic Feedback Protocol (DFP) と Server/Application State Protocol (SASP) のどちらを使用するか、または重みの報告を行わないかを示します。値は DFP、SASP、None のいずれかです。DFP の詳細については、第 8 章「 Dynamic Feedback Protocol のサーバロードバランシングのための設定 」を参照してください。SASP の詳細については、第 7 章「 サーバのロードバランシングのための SASP の設定 」を参照してください。
Reported-Weight	最後に更新された SASP の重み

サービス統計情報カウンタのクリア

zero service コマンドを使用すると、既存の全 CSS サービスを対象に、特定のサービスの統計カウンタをクリアして、ゼロに設定できます。**show service** コマンドの出力で、リセットされたカウンタの統計情報はゼロになります。**zero service** コマンドは、すべてのモードで使用できます。

次の **zero service** コマンドは、どのモードからも実行できます。

- **zero service total-connections** : すべてのサービスに対して Total Connections カウンタをゼロに設定する。
- **zero service total-reused-connections** : すべてのサービスに対して Total Reused Conns カウンタをゼロに設定する。
- **zero service state-transitions** : すべてのサービスに対して State Transitions カウンタをゼロに設定する。

たとえば、すべてのサービスに対して Total Connections カウンタをクリアするには次のように入力します。

```
(config)# zero service total-connections
```



(注)

特定のコンテンツ ルールのすべてのサービスの統計情報カウンタをクリアするには、コンテンツ モードで **zero** コマンドを使用します。またこのコマンドを使用してコンテンツ ルールに関連した特定サービスのコンテンツをクリアします。コンテンツ ルールに関連するサービス統計情報のクリアの詳細については、[第10章「コンテンツ ルールの設定」](#)を参照してください。

以降の内容について

キープアライブ、ソース グループ、負荷、SASP、および DFP の設定については、以降の各章を参照してください。

- [第4章「サービス キープアライブ、グローバル キープアライブ、スクリプト キープアライブの設定」](#)
- [第5章「サービスのソース グループの設定」](#)
- [第6章「サービスの負荷の設定」](#)
- [第7章「サーバのロード バランシングのための SASP の設定」](#)
- [第8章「Dynamic Feedback Protocol のサーバ ロード バランシングのための設定」](#)

所有者の作成と設定については、[第9章「所有者の設定」](#)を参照してください。