



# SSL 設定のクイック スタート

この章では、CSS での SSL 証明書の管理方法、仮想 SSL サーバとバックエンド SSL サーバ用の SSL プロキシリストの作成方法、および SSL プロキシリストの SSL サービスへの追加方法の概要を示します。それぞれの手順に、作業を行うために必要な CLI コマンドも示します。ここに示すクイック スタート手順では、暗号化と認証用の公開キー アルゴリズムとして、広く普及している RSA を使用しています。

CSS に SSL 終了を設定するには、次のクイック スタート手順を実行してください。

1. [RSA 証明書およびキー生成のクイック スタート \(表 2-1\)](#)
2. [RSA 証明書およびキーのインポートのクイック スタート \(表 2-2\)](#)
3. [SSL 終了のプロキシリストのクイック スタート \(表 2-3\)](#)
4. [SSL 終了サービスとコンテンツ ルールのクイック スタート \(表 2-6\)](#)

設定にバックエンド SSL が含まれている場合は、次のクイック スタート手順も実行してください。

1. [バックエンド SSL のプロキシリストのクイック スタート \(表 2-4\)](#)
2. [バックエンド SSL サービスとコンテンツ ルールのクイック スタート \(表 2-7\)](#)

SSL 開始を設定するには、次のクイック スタート手順を実行してください。

1. [SSL 開始のプロキシリストのクイック スタート \(表 2-5\)](#)
2. [SSL 開始サービスのクイック スタート \(表 2-8\)](#)
3. [SSL 開始のコンテンツ ルールのクイック スタート \(表 2-9\)](#)

## RSA 証明書およびキー生成のクイック スタート

表 2-1 に、CSS で RSA キー ペアと証明書を生成し、関連付ける手順の概要を説明します。キーおよび証明書の生成は、CSS に既存のキーまたは証明書が存在しない場合に必要です。内部で SSL のテストを行う場合は、最初に RSA キーと仮証明書を作成することができます。生成された仮証明書の有効期限は 30 日間です。

表 2-1 RSA 証明書およびキーの生成のクイック スタート

---

### 作業とコマンドの例

---

1. グローバル設定モードに移行します。

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

2. 交換に使用する RSA キー ペアを生成します。

```
(config) # ssl genrsa CSSrsakey1 1024 "passwd123"
Please be patient this could take a few minutes
```

3. 生成した RSA キー ペアをファイルと関連付けます。

```
(config) # ssl associate rsakey myrsakey1 CSSrsakey1
```

---

表 2-1 RSA 証明書およびキーの生成のクイック スタート (続き)

## 作業とコマンドの例

4. RSA キーペアを生成した後に、RSA キーペア ファイルの Certificate Signing Request (CSR; 証明書署名要求) を作成します。次に例を示します。

```
(config) # ssl genscr myrsakey1
You are about to be asked to enter information
that will be incorporated into your certificate
request. What you are about to enter is what is
called a Distinguished Name or a DN.
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [US]US
State or Province (full name) [SomeState]MA
Locality Name (city) [SomeCity]Boxborough
Organization Name (company name) [Acme Inc]Cisco Systems, Inc.
Organizational Unit Name (section) [Web Administration]Web Admin
Common Name (your domain name) [www.acme.com]www.cisco.com
Email address [webadmin@acme.com]webadmin@cisco.com
```

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBWCCAQICAQAwZwxCzAJBGNVBAYTA1VTMQswCQYDVQQIEWJNQTETMBEGA1UE
BxMKQm94Ym9yb3VnaDEcMBoGA1UEChMTQ2l2Y28uU3lzdGVtcywgSW5jLjEjESMBAG
A1UECXMJV2ViIEFkbWluMRYwFAYDVQQDEw13d3cuY2l2Y28uY29tMSEwHwYJKoZI
hvcNAQkBFhJra3JvZWJlckBjaXNjby5jb20wX DANBgkqhkiG9w0BAQEFAANLADBI
AkEAqHXjtQUVXvmo6tAWPiMpe6oYhZbJUDgTxbW4VMCygzGZn2wUJTGlrifDB6N3
v+1tKFndE686BhKqfyOidml3wQIDAQABoAAwDQYJKoZIhvcNAQEEBQADQQA94yC3
4SUJJ4UQEnO2OqRGL0ZpAE1c4+IV9aTWK6NmiZsM9Gt0vPhIkLx5jjhVRL1b27Ak
H6D5omXa0SPJan5x
-----END CERTIFICATE REQUEST-----
```

```
CSS11503(config)#
```

**ssl genscr** コマンドを実行すると、Privacy Enhanced Mail (PEM; プライバシー強化メール) フォーマットで符号化された PKCS10 で CSR が生成され、画面に出力されます。生成された CSR は CSS に保存されません。

5. 証明書要求を認証局 (CA) に転送します。主な認証局の多くが、画面に表示された証明書をそのままコピー アンド ペーストして送付できる Web アプリケーションを提供しています。

輸出規制されたブラウザでの 128 ビットの暗号化を許可するグローバル サイト証明書が必要な場合は、CA に SETUP/SGC 証明書またはチェーン証明書を要求してください。

証明書は、7 日以内に届きます。

表 2-1 RSA 証明書およびキーの生成のクイック スタート (続き)

---

**作業とコマンドの例**

---

6. (省略可) 署名付きの証明書が届くまでの間に、CSR を作成し独自の秘密キーでその CSR に署名して仮証明書を作成し、CSR ファイルをテストすることができます。これによって有効な証明書が生成されますが、この証明書はほとんどの Web ブラウザで、認識できない CA によって署名されたものとみなされます。仮証明書の生成については、「[自己署名証明書の生成](#)」を参照してください。
  7. 証明書が 7 日以内に届いたら、その証明書を安全な FTP サーバにファイルとして保存します。
    - サーバ証明書を受け取った場合は、ステップ 11. に進みます。
    - グローバル サイト証明書を受け取った場合は、チェーン証明書を作成する必要があります。ステップ 8. に進みます。
  8. <http://www.verisign.com/support/install/intermediate.htm> でグローバル サイト証明書の中間証明書を取得します。

この証明書は、安全な FTP サーバにファイルとして保存します。
  9. ファイルを作成し、グローバル サイト証明書と中間証明書をそのファイルにコピーします。最初にグローバル サイト証明書をコピーしてから中間証明書をコピーする必要があります。サーバ証明書と中間証明書の間には必ず改行を 1 行挿入してください。
  10. ファイルを保存します。
  11. 「[RSA 証明書およびキーのインポートのクイック スタート](#)」で説明するステップを実行して、CSS に証明書をインポートします。
-

## RSA 証明書およびキーのインポートのクイック スタート

表 2-2 に、RSA 証明書とキー ペアをリモート サーバから CSS にインポートして関連付けるために必要な手順の概要を説明します。

表 2-2 RSA 証明書およびキーのインポートのクイック スタート

---

### 作業とコマンドの例

---

1. 安全なファイル転送プロトコル (FTP) レコード ファイルを定義して、証明書と秘密キーを SFTP サーバから CSS へインポートします。

```
# ftp-record ssl_record 192.168.19.21 johndoe "abc123"
/home/johndoe
```

2. 安全な FTP を使用して、インポートした証明書と秘密キーを CSS に転送します。

```
# copy ssl sftp ssl_record import rsacert.pem PEM "passwd123"
Connecting
Completed successfully
```

```
# copy ssl sftp ssl_record import rsakey.pem PEM "passwd123"
Connecting
Completed successfully
```

3. 設定モードに入ります。

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

4. RSA 公開キーの交換と認証を使用するには、次の手順を実行します。

- a. インポートした RSA 証明書とファイルを関連付けます。

```
(config) # ssl associate cert myrsacert1 rsacert.pem
```

- b. インポートした RSA キー ペアをファイルと関連付けます。

```
(config) # ssl associate rsakey myrsakey1 rsakey.pem
```

5. 関連付けた証明書内の公開キーと、関連付けた秘密キーと共に保存されている公開キーを比較し、両者が同一であることを確認します。

```
(config) # ssl verify myrsacert1 myrsakey1
Certificate mycert1 matches key mykey1
```

---

次の実行設定例は、表 2-2 のコマンドの入力結果を示しています。

```
!***** GLOBAL *****  
ftp-record ssl-record 192.168.19.21 johndoe des-password  
1frapbyg4fldce4d /home/johndoe  
  
ssl associate cert myrsacert1 rsacert.pem  
ssl associate rsakey myrsakey1 rsakey.pem
```

## SSL プロキシ リストのクイック スタート

SSL モジュールが送受信する情報の流れは、SSL プロキシ リストによって決定されます。ここでは、次の機能で使用するプロキシ リストの作成方法について説明します。

- SSL 終了
- バックエンド SSL
- SSL 開始

## SSL 終了のプロキシ リストのクイック スタート

SSL モジュールがクライアントからの SSL 接続を適切に処理して終了させ、サーバへの HTTP 接続を開始するように、SSL プロキシ リスト内に仮想 SSL サーバを定義する必要があります。

表 2-3 に、SSL プロキシ リスト内に仮想 SSL サーバの定義（RSA 証明書とキーペア用の）を作成するために必要な大まかな手順を示します。クライアント認証の設定については、第 4 章「SSL 終了の設定」の「クライアント認証の設定」を参照してください。

表 2-3 SSL 終了のプロキシ リストのクイック スタート

---

**作業とコマンドの例**


---

1. SSL プロキシ リストを作成します。

```
(config)# ssl-proxy-list ssl_list1
Create ssl-list <ssl_list1>, [y/n]: y
```

SSL プロキシ リストを作成すると、CLI が SSL プロキシ リストの `ssl-proxy-list` 設定モードに入ります。

```
(config-ssl-proxy-list[ssl_list1])#
```

---

2. SSL プロキシ リストで仮想 SSL サーバを識別する番号を指定します。

```
(config-ssl-proxy-list[ssl_list1])# ssl-server 20
```

---

3. 仮想 IP (VIP) アドレスを指定します。SSL コンテンツ ルールに一致する VIP アドレスを入力します。

```
(config-ssl-proxy-list[ssl_list1])# ssl-server 20 vip address
192.168.3.6
```

---

4. (省略可) コンテンツ ルールに対応するように仮想 TCP ポート番号を変更する必要がある場合は、その番号を指定します。デフォルトでは、仮想 TCP ポート番号は 443 です。

```
(config-ssl-proxy-list[ssl_list1])# ssl-server 20 port 444
```

---

5. SSL プロキシ リストの仮想 SSL サーバの既存の、RSA 証明書アソシエーションと RSA キー ペア アソシエーションの名前を指定します。

```
(config-ssl-proxy-list[ssl_list1])# ssl-server 20 rsacert
myrsacert1
(config-ssl-proxy-list[ssl_list1])# ssl-server 20 rsakey
myrsakey1
```

---

6. 使用中の RSA 証明書とキーに適した暗号スイート、その暗号スイートに使用するバックエンドのコンテンツ ルールの IP アドレス、およびバックエンドのコンテンツ ルールの TCP ポートを割り当てます。

```
(config-ssl-proxy-list[ssl_list1])# ssl-server 20 cipher
rsa-export-with-rc4-40-md5 192.168.3.6 8080 weight 5
```

---

7. (省略可) リダイレクトする URL のドメイン名の URL リライト オプションを指定して、保護されていない HTTP 300 シリーズのリダイレクトを回避します。

```
(config-ssl-proxy-list[ssl_list1])# ssl-server 20 urlrewrite 22
www.mydomain.com
```

---

表 2-3 SSL 終了のプロキシ リストのクイック スタート (続き)

**作業とコマンドの例**

8. バックエンド SSL サーバへのフローに暗号化が必要な場合、表 2-4 の作業を続けます。必要ない場合、ステップ 9 に進んでください。
9. 完成した SSL プロキシ リストをアクティブ化します。  
(config-ssl-proxy-list[ssl\_list1])# **active**

次の実行設定例は、表 2-3 のコマンドの入力結果を示しています。

```
!***** SSL PROXY LIST *****

ssl-proxy-list ssl_list1
  ssl-server 20
  ssl-server 20 vip address 192.168.3.6
  ssl-server 20 port 444
  ssl-server 20 rsacert myrsacert1
  ssl-server 20 rsakey myrsakey1
  ssl-server 20 cipher rsa-export-with-rc4-40-md5 192.168.3.6 8080
weight 5
  ssl-server 20 urlrewrite 22 www.mydomain.com
active
```

**バックエンド SSL のプロキシ リストのクイック スタート**

CCS から SSL サーバに暗号化したデータを送る必要がある場合は、SSL モジュールでデータを暗号化し、サーバへの SSL 接続を開始できるように、SSL プロキシ リスト内でバックエンドサーバを定義します。バックエンド SSL は、SSL 終了とともに設定する必要があります。SSL 終了のクイック スタートの手順については、「[SSL 終了のプロキシ リストのクイック スタート](#)」を参照してください。



表 2-4 に、バックエンド SSL プロキシ リストの作成に必要な手順の概要を示します。

表 2-4 バックエンド SSL プロキシ リストのクイック スタート

---

#### 作業とコマンドの例

1. 既存の SSL 終了のプロキシ リスト内でバックエンド SSL サーバを識別する番号を指定します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1
```

2. IP アドレスを指定します。バックエンド SSL サーバのサービスのアドレスに対応する IP アドレスを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 ip address  
192.168.4.4
```

3. (省略可) デフォルトでは、バックエンドサーバの仮想 TCP ポート番号は 80 です。変更が必要な場合は、仮想 TCP 番号を割り当てます。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 port 8080
```

4. バックエンドサーバの IP アドレスを指定します。サーバの有効な IP アドレスを入力してください。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 server-ip  
192.168.4.4
```

5. (省略可) デフォルトでは、バックエンドサーバのポート番号は 443 です。変更が必要な場合は、サーバポート番号を割り当てます。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 server-port  
113
```



- (注) 同じアドレスで **backend-server number ip address** および **server-ip** コマンドを設定する場合は、異なるポート番号で **backend-server number port** および **server-port** コマンドを設定してください。
-

表 2-4 バックエンド SSL プロキシ リストのクイック スタート (続き)

**作業とコマンドの例**

6. (省略可) デフォルトでは、バックエンド サーバは利用可能なすべての CSS 暗号スイートをサポートします。必要に応じて、バックエンド SSL サーバが使用する、RSA 証明書とキーなどの特定の暗号スイートを割り当てます。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 cipher
rsa-export-with-rc4-40-md5
```

7. 完成した SSL プロキシ リストをアクティブ化します。

```
(config-ssl-proxy-list[ssl_list1])# active
```

次の実行設定例は、表 2-4 のコマンドを入力した結果 (ボールド体の部分) と、表 2-3 の仮想 SSL サーバに関連するコマンドを入力した結果を示しています。

```
!***** SSL PROXY LIST *****
ssl-proxy-list ssl_list1
  ssl-server 20
  ssl-server 20 vip address 192.168.3.6
  ssl-server 20 port 444
  ssl-server 20 rsacert myrsacert1
  ssl-server 20 rsakey myrsakey1
  ssl-server 20 cipher rsa-export-with-rc4-40-md5 192.168.3.6 8080
weight 5
  ssl-server 20 urlrewrite 22 www.mydomain.com
  active

backend-server 1
  backend-server 1 ip address 192.168.4.4
  backend-server 1 port 8080
  backend-server 1 server-ip 192.168.4.4
  backend-server 1 server-port 113
  backend-server 1 cipher rsa-export-with-rc4-40-md5
  active
```

## SSL 開始のプロキシ リストのクイック スタート

CCS がクライアントからクリア テキストを受信して、暗号化したデータを SSL サーバに送信するように設定する必要がある場合は、SSL モジュールがデータを暗号化してサーバへの SSL 接続を開始できるように、SSL プロキシ リスト内で SSL 開始 バックエンド サーバを定義します。

表 2-5 で、SSL 開始のプロキシ リストの作成に必要な手順の概要を説明します。

**表 2-5 SSL 開始のプロキシ リストのクイック スタート**

---

### 作業とコマンドの例

---

1. SSL プロキシ リストを作成します。

```
(config)# ssl-proxy-list ssl_list1
Create ssl-list <ssl_list1>, [y/n]: y
```

SSL プロキシ リストを作成すると、CLI が SSL プロキシ リストの `ssl-proxy-list` 設定モードに入ります。

```
(config-ssl-proxy-list[ssl_list1])#
```

---

2. 既存の SSL 終了のプロキシ リストでバックエンド SSL サーバを識別する番号を指定します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1
```

---

3. このバックエンド サーバを SSL 開始サーバとして定義します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 type
initiation
```

---

4. IP アドレスを指定します。バックエンド SSL サーバのサービスの IP アドレスに対応する IP アドレスを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 ip address
192.168.2.3
```

---

5. (省略可) デフォルトでは、バックエンド サーバの仮想 TCP ポート番号は 80 です。変更が必要な場合は、仮想 TCP 番号を割り当てます。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 port 8080
```

---

6. バックエンド サーバに有効な IP アドレスを指定します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 server-ip
192.168.2.3
```

---

表 2-5 SSL 開始のプロキシ リストのクイック スタート (続き)

---

**作業とコマンドの例**


---

7. (省略可) デフォルトでは、バックエンド サーバのポート番号は 443 です。変更が必要な場合は、サーバ ポート番号を割り当てます。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 server-port
40443
```



- (注) 同じアドレスで **backend-server number ip address** および **server-ip** コマンドを設定する場合は、異なるポート番号で **backend-server number port** および **server-port** コマンドを設定してください。
- 

8. (省略可) デフォルトでは、バックエンド サーバは利用可能なすべての CSS 暗号スイートをサポートします。必要に応じて、バックエンド サーバが使用する暗号スイートを割り当てます。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 cipher
rsa-with-rc4-128-md5 weight 10
```

9. 必要であれば、クライアントの証明書とキーを、それらを要求する SSL サーバのプロキシ リストに設定します。証明書とキーは、事前にインポートし、CSS 上のファイル名に関連付けておく必要があります。たとえば、既存の RSA クライアントの証明書とキーを設定するには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 rsacert
myrsacert
(config-ssl-proxy-list[ssl_list1])# backend-server 1 rsakey
myrsakey
```

10. (省略可) SSL モジュール (クライアント) によるサーバ認証のために、プロキシ リストに CA 証明書を設定します。CA 証明書は、事前にインポートし、CSS 上のファイル名に関連付けておく必要があります。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 cacert
mycert1
```

11. 完成した SSL プロキシ リストをアクティブ化します。

```
(config-ssl-proxy-list[ssl_list1])# active
```

---

次の実行設定例は、表 2-5 のコマンドの入力結果を表しています。

```
!***** SSL PROXY LIST *****  
  
ssl-proxy-list ssl-list1  
  backend-server 1  
  backend-server 1 initiation  
  backend-server 1 ip address 192.168.2.3  
  backend-server 1 port 8080  
  backend-server 1 server-ip 192.168.2.3  
  backend-server 1 server-port 40443  
  backend-server 1 cipher rsa-with-rc4-128-md5 weight 10  
  backend-server 1 rsacert myrsacert  
  backend-server 1 rsakey myrsakey  
  backend-server 1 cacert mycert1  
  active
```

## SSL サービスとコンテンツ ルールのクイック スタート

CSS に SSL プロキシリストを使用させるには、SSL サービスにプロキシを追加し、SSL コンテンツ ルールにこのサービスを追加する必要があります。ここでは、次の内容について説明します。

- SSL サービスの作成
- SSL コンテンツ ルールの作成
- SSL サービスのコンテンツ ルールへの追加

## SSL 終了サービスとコンテンツ ルールのクイック スタート

表 2-6 に、SSL サービスへの SSL プロキシ リストの追加と SSL コンテンツ ルールの作成など、SSL 終了用 SSL サービスの作成に必要な手順の概要を示します。

**表 2-6 SSL サーバ サービスとコンテンツ ルールのクイック スタート**

---

### 作業とコマンドの例

---

1. SSL サービスを作成します。

```
(config)# service ssl_serv1
Create service <ssl_serv1>, [y/n]: y
```

---

2. サービス タイプとして **ssl-accel** を指定します。

```
(config-service[ssl_serv1])# type ssl-accel
```

---

3. SSL モジュールを装着している CSS シャーシのスロットを指定します。

```
(config-service[ssl_serv1])# slot 3
```

---

4. CSS によるこのサービスへのキープアライブ メッセージの送信を無効にします。

```
(config-service[ssl_serv1])# keepalive type none
```

---

5. SSL プロキシ リストをこの SSL サービスに追加します。

```
(config-service[ssl_serv1])# add ssl-proxy-list ssl_list1
```

---

6. (省略可) 圧縮を有効にしてサーバ応答データを圧縮してからクライアントに戻します。

```
(config-service[ssl_serv1])# compress enable
```

---

HTTP 圧縮の詳細については、第 9 章「HTTP 圧縮の設定」を参照してください。

---

7. SSL サービスをアクティブ化します。

```
(config-service[ssl_serv1])# active
```

---

8. SSL コンテンツ ルールを作成します。

```
(config)# owner ssl_owner
Create owner <ssl_owner>, [y/n]: y
(config-owner[ssl_owner])# content ssl_rule1
Create content <ssl_rule1>, [y/n]: y
```

---

表 2-6 SSL サーバサービスとコンテンツ ルールのクイック スタート (続き)

---

**作業とコマンドの例**

---

9. このコンテンツ ルールの VIP アドレスまたはドメイン名を設定します。VIP アドレスは、SSL プロキシリストで指定したアドレスと同じものを指定してください。

```
(config-owner-content [ssl-rule1]# vip address 192.168.3.6
```

---

10. このコンテンツ ルールの TCP ポート番号を指定します。ポート番号は、SSL プロキシリストで指定したポートと同じものを指定してください。

```
(config-owner-content [ssl-rule1]# port 444
```

---

11. 2 つ以上の SSL モジュールを使用している場合に、レイヤ 5 コンテンツ ルールに対して SSL バージョン 3 のセッション ID に基づくスティッキ性を使用するには、コンテンツ ルールで次のようにパラメータを指定して、SSL セッション ID を再使用できるようにします。

- **application ssl** コマンドを入力して、SSL アプリケーション タイプを指定します。

```
(config-owner-content [ssl-rule1]# application ssl
```

---

- **advanced-balance ssl** コマンドを入力して、SSL に基づくスティッキ性を有効にします。

```
(config-owner-content [ssl-rule1]# advanced-balance ssl
```

---

12. SSL サービスをコンテンツ ルールに追加します。

```
(config-owner-content [ssl_rule1]# add service ssl_serv1
```

---

13. コンテンツ ルールをアクティブ化します。

```
(config-owner-content [ssl_rule1]# active
```

---

14. 設定の変更内容を実行設定に保存します。

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

---

15. 構成にバックエンド SSL が含まれる場合は表 2-7 の作業を、構成に SSL 開始が含まれる場合は、表 2-8 の作業を続けます。
-

次の実行設定例は、表 2-6 のコマンドを入力した結果を示しています。

```
!***** SERVICE *****
service ssl-serv1
  type ssl-accel
  slot 3
  keepalive type none
  add ssl-proxy-list ssl_list1
  compress enable
  active

!***** OWNER *****
owner ssl_owner

content ssl_rule1
  protocol tcp
  vip address 192.168.3.6
  port 444
  application ssl
  advanced-balance ssl
  add service ssl-serv1
  active
```



## バックエンド SSL サービスとコンテンツ ルールのクイック スタート

表 2-7 に SSL プロキシ リストへのサービスの追加や SSL コンテンツ ルールの作成など、バックエンド SSL サーバ用 SSL サービスの作成に必要な手順の概要を示します。SSL プロキシ リストでバックエンド SSL サーバを定義した場合は、この手順を実行します。

表 2-7 バックエンド SSL サービスとコンテンツ ルールのクイック スタート

---

### 作業とコマンドの例

---

1. SSL サービスを作成します。

```
(config)# service ssl_serv2
Create service <ssl_serv2>, [y/n]: y
```

---

2. サービス タイプとして **ssl-accel-backend** を指定します。

```
(config-service[ssl_serv2])# type ssl-accel-backend
```

---

3. このサービスの IP アドレスを設定します。このアドレスは、**backend-server number ip address** コマンドで SSL プロキシ リスト内に設定した IP アドレスと一致させる必要があります。

```
(config-service[ssl_serv2])# ip address 192.168.4.4
```

---

4. (省略可) バックエンド サーバの仮想ポート番号を設定します。このポート番号は、バックエンド サーバに設定した仮想 TCP ポート番号と一致させる必要があります。デフォルトでは、ポート番号は 80、この例では、8080 です。

```
(config-service[ssl_serv2])# port 8080
```

---

表 2-7 バックエンド SSL サービスとコンテンツ ルールのクイック スタート (続き)

**作業とコマンドの例**

5. (省略可) デフォルトでは、このサービスのキープアライブ タイプは ICMP です。バックエンド サービスのキープアライブ タイプは、なし、ネームド、スクリプト化、TCP、SSL、または固定 / 非固定暗号化 HTTP のキープアライブにできます。SSL、TCP、または暗号化のいずれかのタイプを指定する場合で、ポートがこれらのタイプのデフォルト ポートでない場合は、キープアライブで使用するポートを設定する必要があります。

たとえば、キープアライブ タイプを固定暗号化 HTTP に設定するには、次のようにコマンドを実行します。

```
(config-service[ssl_serv2])# keepalive type http encrypt
```

次にバックエンド SSL サーバのポートを設定します。次に例を示します。

```
(config-service[ssl_serv2])# keepalive port 443
```



(注) 暗号化 HTTP でキープアライブを設定する場合、設定済みのバックエンド サーバの IP アドレスと実サーバの IP アドレスは同じである必要があります。

6. SSL プロキシリストをこの SSL サービスに追加します。

```
(config-service[ssl_serv2])# add ssl-proxy-list ssl_list1
```

7. SSL サービスをアクティブ化します。

```
(config-service[ssl_serv2])# active
```

8. このバックエンド サーバを SSL コンテンツ ルールに追加します。

```
(config)# owner ssl_owner
(config-owner[ssl_owner])# content ssl_backend_rule1
Create content <ssl_backend_rule1>, [y/n]: y
```

9. このコンテンツ ルールの仮想 IP (VIP) アドレスまたはドメイン名を設定します。VIP アドレスが仮想 SSL サーバのコンテンツ ルールで指定したアドレスと同じになるようにしてください。

```
(config-owner-content[ssl_backend_rule1])# vip address 192.168.3.6
```

表 2-7 バックエンド SSL サービスとコンテンツ ルールのクイック スタート (続き)

---

**作業とコマンドの例**

---

10. このコンテンツ ルールの TCP ポート番号を指定します。SSL プロキシ リストのバックエンド SSL のエントリに指定した仮想 TCP ポートと同じポート番号を指定してください。

```
(config-owner-content [ssl_backend_rule1])# port 8080
```

---

11. **advanced-balance arrowpoint-cookie** コマンドを入力して、arrowpoint キーに基づくコンテンツ ルールのスティッキ性を有効にします。

```
(config-owner-content [ssl_backend_rule1])# advanced-balance  
arrowpoint-cookie
```

---

12. クッキーに基づくスティッキ性を使用するように、**url** コマンドを、**/\*** を指定して実行します。

```
(config-owner-content [ssl_backend_rule1])# url "/*"
```

---

13. SSL サービスをコンテンツ ルールに追加します。

```
(config-owner-content [ssl_backend_rule1])# add service ssl_serv2
```

---

14. コンテンツ ルールをアクティブ化します。

```
(config-owner-content [ssl_backend_rule1])# active
```

---

15. 設定の変更内容を実行設定に保存します。

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

---

次の実行設定例は、表 2-7 のコマンドを入力した結果（ボールド体の部分）と、表 2-6 の仮想 SSL サーバに関連するコマンドを入力した結果を示しています。

```
!***** SERVICE *****
service ssl_serv1
  type ssl-accel
  slot 3
  keepalive type none
  add ssl-proxy-list ssl_list1
  active

service ssl_serv2
  type ssl-accel-backend
  ip address 192.168.4.4
  port 8080
  keepalive http encrypt
  keepalive port 443
  add ssl-proxy-list ssl_list1
  active

!***** OWNER *****
owner ssl_owner

content ssl_backend_rule1
  vip address 192.168.3.6
  advanced-balance arrowpoint-cookie
  protocol tcp
  port 8080
  url "/*"
  add service ssl_serv2
  active

content ssl_rule1
  protocol tcp
  vip address 192.168.3.6
  port 444
  application ssl
  advanced-balance ssl
  add service ssl_serv1
  active
```

## SSL 開始サービスのクイック スタート

表 2-8 に、SSL 開始サーバ用 SSL サービスの作成に必要な手順の概要を示します。SSL プロキシリストで SSL 開始サーバを定義した場合は、この手順を実行します。

表 2-8 SSL 開始サービスのクイック スタート

---

### 作業とコマンドの例

---

1. SSL サービスを作成します。

```
(config)# service ssl_serv1  
Create service <ssl_serv1>, [y/n]: y
```

---

2. サービスタイプとして **ssl-init** を指定します。

```
(config-service[ssl_serv1])# type ssl-init
```

---

3. このサービスの IP アドレスを設定します。このアドレスは、**backend-server number ip address** コマンドで SSL 開始プロキシリスト内に設定した IP アドレスと一致させる必要があります。「[SSL 開始のプロキシリストのクイック スタート](#)」を参照してください。

```
(config-service[ssl_serv1])# ip address 192.168.2.3
```

---

4. サービスポートを設定します。このサービスポートは、SSL 開始バックエンドサーバポートと一致させる必要があります。

```
(config-service[ssl_serv1])# port 8080
```

---

表 2-8 SSL 開始サービスのクイック スタート (続き)

**作業とコマンドの例**

5. デフォルトでは、このサービスのキープアライブ タイプは ICMP です。SSL 開始では、キープアライブ タイプは ICMP、なし、SSL、TCP、ネームド、スクリプト化、または固定 / 非固定の暗号化 HTTP キープアライブに設定できます。SSL、TCP、または暗号化タイプのいずれかを指定する場合で、ポートがこれらのタイプのデフォルト ポートでない場合は、キープアライブで使用するポートを設定する必要があります。キープアライブ ポートは、SSL 開始バックエンドサーバ ポートと一致させる必要があります。

たとえば、キープアライブ タイプを固定暗号化 HTTP に設定するには、次のようにコマンドを実行します。

```
(config-service[ssl_serv1])# keepalive type http encrypt
(config-service[ssl_serv1])# keepalive port 40443
```



**(注)** 暗号化 HTTP のキープアライブを設定する場合、設定された開始サーバの IP アドレスと実サーバの IP アドレスは同じである必要があります。

6. SSL 開始に指定した SSL モジュールが装着されている CSS シャーシのスロットを指定します。

```
(config-service[ssl_serv1])# slot 5
```

7. SSL プロキシリストをこの SSL サービスに追加します。

```
(config-service[ssl_serv1])# add ssl-proxy-list ssl_list1
```

8. (省略可) 圧縮を有効にしてサーバ応答データを圧縮してからクライアントに戻します。

```
(config-service[ssl_serv1])# compress enable
```

HTTP 圧縮の詳細については、第 9 章「HTTP 圧縮の設定」を参照してください。

9. SSL サービスをアクティブ化します。

```
(config-service[ssl_serv1])# active
```

次の実行設定例は、表 2-8 のコマンドの入力結果を表しています。

```
!***** SERVICE *****
service ssl-serv2
  type ssl-init
  ip address 192.168.2.3
  port 8080
  slot 5
  keepalive type http encrypt
  keepalive port 40443
  add ssl-proxy-list ssl_list1
  compress enable
  active
```

## SSL 開始のコンテンツ ルールのクイック スタート

表 2-9 に、SSL 開始サーバ用 SSL コンテンツ ルールの作成に必要な手順の概要を示します。SSL プロキシ リストで SSL 開始サーバを定義した場合は、この手順を実施します。

**表 2-9 SSL 開始のコンテンツ ルールのクイック スタート**

1.	必要に応じて所有者を作成します。
	<pre>(config)# owner ssl_owner Create owner &lt;ssl_owner&gt;, [y/n]: y</pre>
2.	SSL 開始バックエンドサーバを SSL コンテンツ ルールに追加します。
	<pre>(config)# owner ssl_owner (config-owner[ssl_owner])# content ssl_init_rule1 Create content &lt;ssl_init_rule1&gt;, [y/n]: y</pre>
3.	このコンテンツ ルールの仮想 IP (VIP) アドレスまたはドメイン名を設定します。
	<pre>(config-owner-content [ssl_backend_rule1])# vip address 192.168.2.3</pre>
4.	このコンテンツ ルールの TCP ポート番号を指定します。
	<pre>(config-owner-content [ssl_backend_rule1])# port 80</pre>
5.	(省略可) クッキーに基づくスティッキ性を使用するように、 <code>url</code> コマンドを、 <code>/*</code> を指定して実行します。
	<pre>(config-owner-content [ssl_backend_rule1])# url "/*"</pre>
6.	(省略可) <code>advanced-balance arrowpoint-cookie</code> コマンドを入力して、 <code>arrowpoint</code> クッキーに基づくコンテンツ ルールのスティッキ性を有効にします。
	<pre>(config-owner-content [ssl_backend_rule1])# advanced-balance arrowpoint-cookie</pre>
7.	SSL サービスをコンテンツ ルールに追加します。
	<pre>(config-owner-content [ssl_backend_rule1])# add service ssl_serv2</pre>
8.	コンテンツ ルールをアクティブ化します。
	<pre>(config-owner-content [ssl_backend_rule1])# active</pre>
9.	設定の変更内容を実行設定に保存します。
	<pre># copy running-config startup-config</pre>



次の実行設定例は、表 2-9 のコマンドを入力した結果を示しています。

```
!***** OWNER *****
owner ssl_owner

content ssl_init_rule1
  vip address 192.168.2.3
  port 80
  url "/*"
  advanced-balance arrowpoint-cookie
  add service ssl_serv1
  active
```

