



## SBC multi-VRF の実装

カスタマー エッジ (CE) デバイス (つまり、顧客宅内ルータ) で SBC の multi-VRF (VPN ルーティングおよび転送) サポート機能を使用すると、プロバイダー エッジ (PE) がパケットを相互に再配布する場合にループの防止に必要な、PE のチェックを抑制する機能を提供します。マルチ VRF では、1 台だけのルータを使用して、通常であれば複数のルータが実行するタスクを完了できます。マルチプロトコル ラベル スイッチング (MPLS) およびボーダー ゲートウェイ プロトコル (BGP) を備える必要がないネットワーク上で実行できます。



(注)

この章で使用するコマンドの詳細な説明については、『*Cisco IOS XR Session Border Controller Command Reference*』を参照してください。この章で言及する他のコマンドについては、コマンドリファレンス マスター インデックス (オンライン検索) を使用して、該当するマニュアルを参照してください。

### SBC multi-VRF の実装機能の履歴

リリース	変更内容
リリース 3.3.0	この機能は、Cisco XR 12000 シリーズ ルータで導入されました。
リリース 3.4.0	変更なし。
リリース 3.4.1	Cisco XR 12000 シリーズ ルータで、メディア バイパスが異なる隣接間で発生する可能性があります。 <b>media-bypass</b> コマンドが廃止され、 <b>media-bypass-forbid</b> コマンドが追加されました。
リリース 3.5.0	変更なし。

## 内容

- 「[Multi-VRF の実装の前提条件](#)」 (P.308)
- 「[マルチ VRF の実装の概要](#)」 (P.308)
- 「[Multi-VRF の実装方法](#)」 (P.309)
- 「[マルチ VRF 実装の設定例](#)」 (P.318)
- 「[その他の関連資料](#)」 (P.322)
- 「[関連コマンドのまとめ](#)」 (P.324)

## Multi-VRF の実装の前提条件

次に、SBC multi-VRF を実装するための前提条件を示します。

- 使用される SBC コマンドの適切なタスク ID を含むタスク グループに関連付けられているユーザグループに属している必要があります。ユーザ グループおよびタスク ID の詳細については、『Cisco IOS XR System Security Configuration Guide』の「Configuring AAA Services on Cisco IOS XR Software」モジュールを参照してください。
- SBC ソフトウェアのパッケージ インストール エンベロップ (PIE) をインストールしてアクティブにする必要があります。  
PIE のインストールに関する詳細については、『Cisco IOS XR Getting Started Guide』の「Upgrading and Managing Cisco IOS XR Software」モジュールを参照してください。
- multi-VRF を実装する前に、SBC を作成しておく必要があります。手順については、「SBC 設定の前提条件」モジュールを参照してください。

## マルチ VRF の実装の概要

カスタマー エッジ (CE) デバイス (つまり、顧客宅内ルータ) で SBC の multi-VRF (VPN ルーティングおよび転送) サポート機能を使用すると、プロバイダー エッジ (PE) がパケットを相互に再配布する場合にループの防止に必要な、PE のチェックを抑制する機能を提供します。マルチ VRF では、1 台だけのルータを使用して、通常であれば複数のルータが実行するタスクを完了できます。マルチプロトコル ラベル スイッチング (MPLS) およびボーダー ゲートウェイ プロトコル (BGP) を備える必要がないネットワーク上で実行できます。

VRF を PE 以外のルータで使用すると、チェックはオフになり、IP プレフィックスへの経路を使用して VRF ルーティング テーブルを正しく入力できます。また、バーチャルプライベート ネットワーク (VPN) 機能は低エンド システムでは完全にサポートされていないため、マルチ VRF が重要になります。マルチ VRF は、インプリケーション アドレス空間によって、1 台のルータ内でルーティング インスタンスを論理的に区切ります。

次に、multi-VRF の機能の概要を説明します。

- 1 台の物理ルータを複数の仮想ルータに分割し、各ルータに独自のインターフェイスのセット、ルーティング テーブル、およびルーティング テーブルを含めることができます。SBC は、カスタマーごとに複数の (重複および独立した) ルーティング テーブル (アドレッシング) をサポートしています。仮想ルーティング コンテキストを使用して、1 台のルータ内のルーティング ドメインを区切ります。
- 複数のルータが必要で、利用可能なルータが 1 台だけの場合に、マルチ VRF を使用できます。
- 1 つの物理インターフェイスを複数の仮想ルータに所属させるには、サブインターフェイス (フレームリレー、ATM、VLAN) を使用します。
- BGP と MPLS は使用されません。
- VRF 間の接続は提供されません (VRF 間での内部エクスポートおよびインポートの場合には、BGP を使用する必要があります)。
- 同じ VPN サイトの 2 つのエンドポイント間にコールを配置すると、SBC はそのエンドポイント間で直接メディアをルーティングして、ネットワーク使用率を削減できます。
- SBC で multi-VRF を実現すると、両方のエンドポイントが同じ VPN にある場合には、メディアバイパスを有効にして、最適化を図ることができます。

リリース 3.4.1 の場合、デフォルトでは同じ VPN 上のすべての隣接でメディア バイパスが有効になっています。 **media-bypass-forbid** コマンド（このコマンドは CAC ポリシー専用の実装されています）を使用すると、メディア バイパスをオフにできます。

## Multi-VRF の実装方法

SBC multi-VRF の実装については、次の項を参照してください。

- [multi-VRF の設定](#)
- [H.323 隣接への VRF の関連付け](#)
- [SIP 隣接と VRF との関連付け](#)
- [DBE への VRF の設定（分散モデル専用）](#)

## multi-VRF の設定

このタスクでは、SBC が統合配置モデルの multi-VRF モードで実行されるルータを設定します。必要に応じて、インターフェイスと、SBC の SVI、隣接、および DBE メディア アドレスとの間の関係をメモします。

### 手順の概要

1. **configure**
2. **interface sbc number**
3. **vrf vrf-name**
4. **ipv4 address address prefix**
5. **service-location preferred-active node-id [preferred-standby node-id]**
6. **sbc service-name**
7. **service-location preferred-active node-id [preferred-standby node-id]**
8. **sbc**
9. **activate**
10. H.323 隣接と VRF を関連付けます（「[H.323 隣接への VRF の関連付け](#)」を参照）。
11. SIP 隣接と VRF を関連付けます（「[SIP 隣接と VRF との関連付け](#)」を参照）。
12. **dbe**
13. **media-address ipv4 ipv4\_address [vrf vrf\_name]**
14. **activate**
15. **end**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<b>configure</b>  例： RP/0/0/CPU0:router# configure RP/0/0/CPU0:router(config)#	グローバル コンフィギュレーション モードをイネーブルにします。
ステップ2	<b>interface sbc number</b>  例： RP/0/0/CPU0:router(config-if)# interface SBC1	SBC インターフェイス（必要に応じて作成）のモードを開始します。 <i>number</i> 引数は、1 ~ 2000 の値である必要があります。
ステップ3	<b>vrf vrf_name</b>  例： RP/0/0/CPU0:router(config-if)# vrf my_vrf1	特定のバーチャル プライベート ネットワーク（VPN）に H.323 隣接を関連付けます。
ステップ4	<b>ipv4 address address prefix</b>  例： RP/0/0/CPU0:router(config-if)# ipv4 address 88.88.101.10 255.255.255.0	インターフェイスに IPv4 アドレスを割り当てます。
ステップ5	<b>service-location preferred-active node-id</b> [ <b>preferred standby node-id</b> ]  例： RP/0/0/CPU0:router(config-if)# service-location preferred-active 0/4/CPU0 preferred-standby 0/5/CPU0	サービス カードが SBC 機能をプライマリとして、およびオプションでセカンダリ ロケーションとして実行できるようにします。
ステップ6	<b>sbc service-name</b>  例： RP/0/0/CPU0:router(config-if)# sbc lite_8	SBC サービスのモードを開始します。  • <i>service-name</i> 引数を使用して、サービスの名前を定義します。
ステップ7	<b>service-location preferred-active node-id</b> [ <b>preferred standby node-id</b> ]  例： RP/0/0/CPU0:router(config-sbc)# service-location preferred-active 0/4/CPU0 preferred-standby 0/5/CPU0	サービス カードが SBC 機能をプライマリとして、およびオプションでセカンダリ ロケーションとして実行できるようにします。
ステップ8	<b>sbe</b>  例： RP/0/0/CPU0:router(config-sbc)# sbe RP/0/0/CPU0:router(config-sbc-sbe)#	SBC サービス内で SBE エンティティのモードを開始します。

	コマンドまたはアクション	目的
ステップ9	<code>activate</code>  例： RP/0/0/CPU0:router(config-sbc-sbe-acc)# activate	SBC サービスを開始します。
ステップ10	H.323 隣接 <code>h323my_vrf1</code> を <code>vpn1</code> に関連付けます。  例： 「 <a href="#">H.323 隣接への VRF の関連付け：例</a> 」の項を参照してください。	H.323 隣接を VPN に関連付けます。「 <a href="#">H.323 隣接への VRF の関連付け</a> 」の項を参照してください。
ステップ11	SIP 隣接 <code>sipmy_vrf1</code> を <code>vpn1</code> に関連付けます。  例： 「 <a href="#">SIP 隣接と VRF との関連付け：例</a> 」の項を参照してください。	SIP 隣接を VPN に関連付けます。「 <a href="#">SIP 隣接と VRF との関連付け</a> 」の項を参照してください。
ステップ12	<code>dbe</code>  例： RP/0/0/CPU0:router(config-sbc-sbe)# <code>dbe</code> RP/0/0/CPU0:router(config-sbc-dbe)#	SBC サービス内で DBE エンティティのモードを開始します。
ステップ13	<code>media-address ipv4 ipv4_address [vrf vrf_name]</code>  例： RP/0/0/CPU0:router(config-sbc-dbe)# media-address	DBE メディア アドレス プール内に IPv4 アドレスを作成します。
ステップ14	<code>activate</code>  例： RP/0/0/CPU0:router(config-sbc-sbe-acc)# activate	SBC サービスを開始します。
ステップ15	<code>end</code>  例： RP/0/0/CPU0:router(config-sbc-sbe-acc)# activate	現在のコンフィギュレーション セッションを終了します。

## H.323 隣接への VRF の関連付け

このタスクでは、H.323 隣接を VPN に関連付けます。

### 手順の概要

1. `adjacency h323 adjacency-name`
2. `vrf vrf_name`
3. `signaling-address ipv4 local_signaling_IP_address`

4. `signaling-port port_num`
5. `remote-address ipv4 remote_IP_address/prefix`
6. `signaling-peer [gk] peer_address`
7. `signaling-peer-port port_num`
8. `dtmf-relay rtp-nte`
9. `account account_name`
10. `media-bypass` (Optional command)
11. `media-bypass-forbid`
12. `attach`

## 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<b>adjacency h323 adjacency-name</b>  <b>例:</b> RP/0/0/CPU0:router(config-sbc-sbe)# adjacency h323 h323my_vrfl RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)#	SBE H.323 隣接のモードを開始します。  <ul style="list-style-type: none"> <li><code>adjacency-name</code> 引数を使用して、サービスの名前を定義します。</li> </ul>
ステップ2	<b>vrf vrf_name</b>  <b>例:</b> RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# vrf my_vrfl	特定のバーチャルプライベートネットワーク (VPN) に H.323 隣接を関連付けます。
ステップ3	<b>signaling-address ipv4 local_signaling_IP_address</b>  <b>例:</b> RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# signaling-address ipv4 88.88.101.11	H.323 隣接のローカル IPv4 シグナリング アドレスを指定します。
ステップ4	<b>signaling-port port_num</b>  <b>例:</b> RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# signaling-port 1720	H.323 隣接のローカル シグナリング ポートを指定します。
ステップ5	<b>remote-address ipv4 ipv4_IP_address/prefix</b>  <b>例:</b> RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# remote-address ipv4 10.10.101.4/32	隣接を通じてコンタクトされる一連のリモート シグナリング ピアを、特定の IP アドレス プレフィックスを持つものに限定します。
ステップ6	<b>signaling-peer [gk] peer_address</b>  <b>例:</b> RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# signaling-peer gk 10.10.101.4	H.323 隣接が使用するリモート シグナリング ピアを指定します。

	コマンドまたはアクション	目的
ステップ7	<code>signaling-peer-port port_num</code>  例： RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# signaling-peer-port 1720	H.323 隣接が使用するリモートシグナリングピアポートを指定します。
ステップ8	<code>dtmf-relay [rtp-nte]</code>  例： RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# dtmf-relay	H.323 隣接のDTMFリレーを設定します。  • <code>rtp-nte</code> を指定すると、RFC 2833 <i>Named Telephone Event (rtp-nte)</i> サポートがイネーブルになります。これを指定しないと、 <code>rtp-nte</code> サポートはディセーブルになります。
ステップ9	<code>account account_name</code>  例： RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# account h323-my_vrf1	SBE 上のアカウントに属するものとして H.323 隣接を定義します。
ステップ10	<code>media-bypass</code>  例： RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# media-bypass	(任意) メディアトラフィックがDBEをバイパスできるように隣接を設定します。  このコマンドは任意であり、1つの隣接でだけ機能します。
ステップ11	<code>media-bypass-forbid</code>  例： RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# media-bypass-forbid	(リリース 3.4.1) メディアトラフィックがDBEをバイパスするのを禁止するように H.323 隣接を設定します。  設定されていない場合は、この隣接で開始および終了する通話のメディアトラフィックはエンドポイント間を直接フローし、両方の隣接が同じVPNにある限りはDBEを経由しません。
ステップ12	<code>attach</code>  例： RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# attach	隣接をアタッチします。

## SIP 隣接と VRF との関連付け

このタスクでは、SIP 隣接を VPN と関連付けます。

### 手順の概要

1. `adjacency sip adjacency-name`
2. `vrf vrf_name`
3. `signaling-address ipv4 local_signaling_IP_address`
4. `signaling-port port_num`
5. `remote-address ipv4 local_signaling_IP_address/prefix`
6. `local-id host name`
7. `signaling-peer [gk] peer_address`

8. `signaling-peer-port port_num`
9. `account account-name`
10. `media-bypass` (任意)
11. `media-bypass-forbid`
12. `attach`

## 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<p><code>adjacency sip adjacency-name</code></p> <p>例:</p> <pre>RP/0/0/CPU0:router(config-sbc-sbe)# adjacency sip sipGW RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)#</pre>	<p>SBE SIP 隣接のモードを開始します。</p> <ul style="list-style-type: none"> <li><code>adjacency-name</code> 引数を使用して、サービスの名前を定義します。</li> </ul>
ステップ2	<p><code>vrf vrf_name</code></p> <p>例:</p> <pre>RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# vrf my_vrf1</pre>	<p>特定のバーチャル プライベート ネットワーク (VPN) に H.323 隣接を関連付けます。</p>
ステップ3	<p><code>signaling-address ipv4 ipv4_IP_address</code></p> <p>例:</p> <pre>RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# signaling-address ipv4 88.88.88.88.101.11</pre>	<p>SIP 隣接のローカル IPv4 シグナリング アドレスを指定します。</p>
ステップ4	<p><code>signaling-port port_num</code></p> <p>例:</p> <pre>RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# signaling-port 5060</pre>	<p>SIP 隣接のローカル シグナリング ポートを指定します。</p>
ステップ5	<p><code>remote-address ipv4 remote_IP_address/prefix</code></p> <p>例:</p> <pre>RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# remote-address ipv4 10.10.101.4/32</pre>	<p>隣接を通じてコンタクトされる一連のリモート シグナリング ピアを、特定の IP アドレス プレフィックスを持つものに限定します。</p>
ステップ6	<p><code>local-id host address</code></p> <p>例:</p> <pre>RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# local-id host 88.88.101.11</pre>	<p>SIP 隣接のローカル識別名を設定します。</p>
ステップ7	<p><code>signaling-peer [gk] peer_address</code></p> <p>例:</p> <pre>RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# signaling-peer 10.10.101.4</pre>	<p>SIP 隣接が使用するリモート シグナリング ピアを指定します。</p>



	コマンドまたはアクション	目的
ステップ 8	<b>signaling-peer-port</b> <i>port_num</i>  例： RP/0/0/CPU0:router (config-sbc-sbe-adj-sip) # signaling-peer-port 5060	SIP 隣接が使用するリモート シグナリング ピア ポートを指定します。
ステップ 9	<b>account</b> <i>account_name</i>  例： RP/0/0/CPU0:router (config-sbc-sbe-adj-sip) # account sipmy_vrf2	SIP 隣接を SBE のアカウントに所属しているものとして定義します。
ステップ 10	<b>media-bypass</b>  例： RP/0/0/CPU0:router (config-sbc-sbe-adj-sip) # media-bypass	(任意) メディア トラフィックが DBE をバイパスできるように隣接を設定します。  このコマンドは任意であり、1 つの隣接だけで動作します。
ステップ 11	<b>media-bypass-forbid</b>  例： RP/0/0/CPU0:router (config-sbc-sbe-adj-sip) # media-bypass-forbid	(リリース 3.4.1) メディア トラフィックが DBE をバイパスするのを禁止するように SIP 隣接を設定します。  設定されていない場合は、この隣接で開始および終了する通話のメディア トラフィックはエンドポイント間を直接フローし、両方の隣接が同じ VPN にある限りは DBE を経由しません。
ステップ 12	<b>attach</b>  例： RP/0/0/CPU0:router (config-sbc-sbe-adj-sip) # attach	隣接をアタッチします。

## DBE への VRF の設定 (分散モデル専用)

このタスクでは、分散モデルで VRF を使用して DBE を設定します。

### 手順の概要

1. **configure**
2. **interface sbc number**
3. **vrf vrf-name**
4. **ipv4 address address prefix**
5. **service-location preferred-active node-id [preferred-standby node-id]**
6. **interface sbc number**
7. **ipv4 address address prefix**
8. **service-location preferred-active node-id [preferred-standby node-id]**
9. **sbc service-name**
10. **service-location preferred-active node-id [preferred-standby node-id]**
11. **dbe**

12. `vdbe [global]`
13. `control-address h248 ipv4 IPv4_H.248_IP_address`
14. `controller h248 controller-index`
15. `transport [udp | tcp]`
16. `remote-address ipv4 remote-address`
17. `attach-controllers`
18. `media-address ipv4 ipv4_address [vrf vrf_name]`
19. `activate`

## 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<code>configure</code>  例： RP/0/0/CPU0:router# <code>configure</code> RP/0/0/CPU0:router(config)#	グローバル コンフィギュレーション モードをイネーブルにします。
ステップ2	<code>interface sbcnumber</code>  例： RP/0/0/CPU0:router(config)# <code>interface SBC1</code> RP/0/0/CPU0:router(config-if)#	SBC インターフェイス（必要に応じて作成）のモードを開始します。 <code>number</code> 引数は、1 ~ 2000 の値である必要があります。
ステップ3	<code>vrf vrf_name</code>  例： RP/0/0/CPU0:router(config-if)# <code>vrf my_vrf1</code>	特定のバーチャルプライベートネットワーク（VPN）に H.323 隣接を関連付けます。
ステップ4	<code>ipv4 address address prefix</code>  例： RP/0/0/CPU0:router(config-if)# <code>ipv4 address</code> 88.88.130.10 255.255.255.0	インターフェイスに IPv4 アドレスを割り当てます。
ステップ5	<code>service-location preferred-active node-id</code> [ <code>preferred standby node-id</code> ]  例： RP/0/0/CPU0:router(config-if)# <code>service-location</code> <code>preferred-active 0/1/CPU0</code>	サービスカードが SBC 機能をプライマリとして、およびオプションでセカンダリ ロケーションとして実行できるようにします。
ステップ6	<code>interface sbc number</code>  例： RP/0/0/CPU0:router(config-if)# <code>interface SBC2</code>	SBC インターフェイス（必要に応じて作成）のモードを開始します。 <code>number</code> 引数は、1 ~ 2000 の値である必要があります。

	コマンドまたはアクション	目的
ステップ7	<b>ipv4 address address prefix</b>  例： RP/0/0/CPU0:router(config-if)# ipv4 address 88.88.130.10 255.255.255.0	インターフェイスに IPv4 アドレスを割り当てます。
ステップ8	<b>service-location preferred active node-id</b> <b>[preferred standby node-id]</b>  例： RP/0/0/CPU0:router(config-if)# service-location preferred-active 0/1/CPU0	サービスカードが SBC 機能をプライマリとして、およびオプションでセカンダリ ロケーションとして実行できるようにします。
ステップ9	<b>sbc service-name</b>  例： RP/0/0/CPU0:router(config-if)# sbc lite_1 RP/0/0/CPU0:router(config-sbc)#	SBC サービスのモードを開始します。 <ul style="list-style-type: none"> <li>• <i>service-name</i> 引数を使用して、サービスの名前を定義します。</li> </ul>
ステップ10	<b>service-location preferred active node-id</b> <b>[preferred standby node-id]</b>  例： RP/0/0/CPU0:router(config-sbc)# service-location preferred-active 0/1/CPU0	サービスカードが SBC 機能をプライマリとして、およびオプションでセカンダリ ロケーションとして実行できるようにします。
ステップ11	<b>dbe</b>  例： RP/0/0/CPU0:router(config-sbc)# dbe RP/0/0/CPU0:router(config-sbc-dbe)#	SBC サービス内で DBE エンティティを設定するモードを開始します。
ステップ12	<b>vdbe [global]</b>  例： RP/0/0/CPU0:router(config-sbc-dbe)# vdbe RP/0/0/CPU0:router(config-sbc-dbe-vdbe)#	仮想 DBE パラメータを設定するモードを開始します。
ステップ13	<b>control-address h248 ipv4 ipv4_H.248_IP_address</b>  例： RP/0/0/CPU0:router(config-sbc-dbe-vdbe)# control-address h248 ipv4 88.88.130.100	特定の IPv4 H.248 制御アドレスを使用するように DBE を設定します。
ステップ14	<b>controller h248 controller-index</b>  例： RP/0/0/CPU0:router(config-sbc-dbe-vdbe)# controller h248 1 RP/0/0/CPU0:router(config-sbc-dbe-vdbe-h248)#	DBE の H.248 コントローラを設定するモードを開始します。

	コマンドまたはアクション	目的
ステップ 15	<code>transport [udp   tcp]</code>  例： RP/0/0/CPU0:router(config-sbc-dbe-vdbe-h248)# transport udp	H.248 コントローラで、H.248 制御シグナリングに UDP または TCP を使用するように DBE を設定します。
ステップ 16	<code>remote-address ipv4 remote-address</code>  例： RP/0/0/CPU0:router(config-sbc-dbe-vdbe-h248)# remote-address ipv4 88.88.101.21	H.248 コントローラの SBE 上に接続するリモートアドレスを定義します。
ステップ 17	<code>attach-controllers</code>  例： RP/0/0/CPU0:router(config-sbc-dbe-vdbe-h248)# attach-controllers RP/0/0/CPU0:router(config-sbc-dbe-vdbe)#	コントローラに接続する DBE を設定します。
ステップ 18	<code>media-address ipv4 ipv4_address [vrf vrf_name]</code>  例： RP/0/0/CPU0:router(config-sbc-dbe-vdbe)# media-address ipv4 88.88.130.2 vrf my_vrf	DBE メディア アドレス プールを設定するモードを開始し、VRF で使用するアドレス プールを作成します。
ステップ 19	<code>activate</code>  例： RP/0/0/CPU0:router(config-sbc-dbe-vdbe)# activate	SBC サービスを開始します。

## マルチ VRF 実装の設定例

ここでは、次の設定例について説明します。

- [マルチ VRF の設定 : 例](#)
- [H.323 隣接への VRF の関連付け : 例](#)
- [SIP 隣接と VRF との関連付け : 例](#)
- [DBE への VRF の設定 \(分散モデル専用\) : 例](#)

## マルチ VRF の設定 : 例

この例では、Service Virtual Interface (SVI; サービス仮想インターフェイス) と隣接を追加して、VPN を関連付ける方法について示します。callgen7 からコールが発信されると、sbc1 はそれを取得し、sbc2 にブリッジングし、callgen8 に渡します。



(注) この設定では、ゲートウェイの隣接および番号分析が削除されています。

1. ラインカード インターフェイスを my\_vrf2 に関連付ける設定をします。

```
cdp
vrf my_vrf2
ipv4 address 192.168.229.1 255.255.255.0
negotiation auto
!
interface GigabitEthernet0/3/0/1
description vrf-lite 1 to node1
```

2. ラインカード インターフェイスを my\_vrf1 に関連付ける設定をします。

```
cdp
vrf my_vrf1
ipv4 address 1.0.4.1 255.255.255.0
negotiation auto
!
interface SBC1
vrf my_vrf1
```

3. SVI を my\_vrf1 に関連付ける設定をします。

```
ipv4 address 88.88.101.10 255.255.255.0
service-location preferred-active 0/4/CPU0 preferred-standby 0/5/CPU0
!
interface SBC2
vrf my_vrf2
```

4. SVI を my\_vrf2 に関連付ける設定をします。

```
ipv4 address 88.88.129.10 255.255.255.0
service-location preferred-active 0/4/CPU0 preferred-standby 0/5/CPU0
!
!
sbc lite_8
service-location preferred-active 0/4/CPU0 preferred-standby 0/5/CPU0
sbe
activate
adjacency h323 h323my_vrf1
vrf my_vrf1
```

5. H.323 の隣接を my\_vrf1 に関連付ける設定をします。

```
signaling-address ipv4 88.88.101.11
signaling-port 1720
remote-address ipv4 10.10.101.4/32
signaling-peer 10.10.101.4
signaling-peer-port 1720
dtmf-relay
account h323-my_vrf1
attach
!
adjacency h323 h323my_vrf2
vrf my_vrf2
```

6. H.323 の隣接を my\_vrf2 に関連付ける設定をします。

```
signaling-address ipv4 88.88.129.11
signaling-port 1720
remote-address ipv4 10.10.115.4/32
signaling-peer 10.10.115.4
signaling-peer-port 1720
dtmf-relay
account h323-my_vrf2
attach
!
adjacency sip sipmy_vrf1
vrf my_vrf1
```

7. SIP の隣接を my\_vrf1 に関連付ける設定をします。

```
signaling-address ipv4 88.88.101.11
signaling-port 5060
remote-address ipv4 10.10.101.4/32
local-id host 88.88.101.11
signaling-peer 10.10.101.4
signaling-peer-port 5060
account sip-my_vrf1
attach
!
adjacency sip sipmy_vrf2
vrf my_vrf2
```

8. SIP の隣接を my\_vrf2 に関連付ける設定をします。

```
signaling-address ipv4 88.88.129.11
signaling-port 5060
remote-address ipv4 10.10.115.4/32
local-id host 88.88.129.11
signaling-peer 10.10.115.4
signaling-peer-port 5060
account sip-my_vrf2
attach
!
!
dbe
media-address ipv4 88.88.101.2 vrf my_vrf1
```

9. メディア アドレスを my\_vrf1 に関連付ける設定をします。

```
media-address ipv4 88.88.129.2 vrf my_vrf2
```

10. メディア アドレスを my\_vrf2 に関連付ける設定をします。

```
activate
!
!
end
```

## H.323 隣接への VRF の関連付け：例

この設定例では、VPN に関連付けられた H.323 隣接を作成します。

```
adjacency h323 h323my_vrf1
  vrf my_vrf1
  signaling-address ipv4 88.88.101.11
  signaling-port 1720
  remote-address ipv4 10.10.101.4/32
  signaling-peer 10.10.101.4
  signaling-peer-port 1720
  dtmf-relay
  account h323-my_vrf1
  attach
```

## SIP 隣接と VRF との関連付け：例

この設定では、VPN に関連付けられた SIP 隣接を作成する例を示します。

```
adjacency sip sipmy_vrf1
  vrf my_vrf1
  signaling-address ipv4 88.88.101.11
  signaling-port 5060
  remote-address ipv4 10.10.101.4/32
  local-id host 88.88.101.11
  signaling-peer 10.10.101.4
  signaling-peer-port 5060
  account sip-my_vrf1
  attach
```

## DBE への VRF の設定（分散モデル専用）：例

次の例では、分散モデルの DBE に VRF を設定する方法を示します。

my\_vrf1 にはメディア アドレスのみ含むことができ、制御アドレス シグナリングはデフォルトの VRF に含める必要があります。統合モデルの SBE および DBE と同様、メディア アドレスは SVI と、my\_vrf1 の物理インターフェイスに関連付けられます。



(注) 制御シグナリング用に 2 つめの SVI がデフォルト VRF が必要になります。

```
interface SBC1
  vrf my_vrf1
  ipv4 address 88.88.130.10 255.255.255.0
  service-location preferred-active 0/1/CPU0
  !
interface SBC2
  ipv4 address 88.88.130.10 255.255.255.0
  service-location preferred-active 0/1/CPU0
  !
```

```

sbclite_1
service-location preferred-active 0/1/CPU0
dbs
vdbs
control-address h248 ipv4 88.88.130.100
controller h248 1
transport udp
remote-address ipv4 88.88.101.21
!
attach-controllers
!
media-address ipv4 88.88.130.2 vrf my_vrf1
activate

```

## その他の関連資料

ここでは、SBC multi-VRF の実装に関する関連資料について説明します。

## 関連資料

関連項目	マニュアル タイトル
Cisco IOS XR マスター コマンド リファレンス	『Cisco IOS XR Master Commands List』
Cisco IOS XR SBC インターフェイス コンフィギュレーション コマンド	『Cisco IOS XR Session Border Controller Command Reference』
Cisco IOS XR ソフトウェアを使用するルータを初回に起動し設定するための情報	『Cisco IOS XR Getting Started Guide』
Cisco IOS XR コマンド モード	『Cisco IOS XR Command Mode Reference』

## 標準

標準	タイトル
この機能でサポートされる新規の標準または変更された標準はありません。また、既存の標準のサポートは変更されていません。	—



## MIB

MIB	MIB のリンク
—	<p>Cisco IOS XR ソフトウェアを使用して MIB の場所を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用して、[Cisco Access Products] メニューからプラットフォームを選択します。</p> <p><a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></p>

## RFC

RFC	タイトル
RFC 2685	『Virtual Private Networks Identifier』
RFC 1918	『Address Allocation for Private Internets』
RFC 2547	『BGP/MPLS VPNs』

## シスコのテクニカル サポート

説明	リンク
<p>シスコのテクニカル サポート Web サイトでは、製品、テクノロジー、ソリューション、技術的なヒント、およびツールへのリンクなどの、数千ページに及ぶ技術情報が検索可能です。Cisco.com に登録済みのユーザは、このページから詳細情報にアクセスできます。</p>	<p><a href="http://www.cisco.com/en/US/support/index.html">http://www.cisco.com/en/US/support/index.html</a></p>

## 関連コマンドのまとめ

ここでは、Cisco XR 12000 シリーズ ルータでの SBC multi-VRF の設定に関連するコマンドの一覧をアルファベット順に示します。コマンドの詳細については、『Cisco IOS XR Session Border Controller Command Reference』を参照してください。

コマンド	目的
<code>interface sbc number</code>	SBC インターフェイス（必要に応じて作成）のモードを開始します。 <i>number</i> 引数は、1 ～ 2000 の値である必要があります。
<code>ipv4 address address prefix</code>	インターフェイスに IPv4 アドレスを割り当てます。
<code>media-address ipv4 ipv4_address [vrf vrf_name]</code>	DBE メディア アドレス プール内に IPv4 アドレスを作成します。
<code>service-location preferred active node-id [preferred standby node-id]</code>	サービス カードが SBC 機能をプライマリとして、およびオプションでセカンダリ ロケーションとして実行できるようにします。
<code>vdbe [global]</code>	仮想 DBE パラメータを設定するモードを開始します。
<code>vrf vrf_name</code>	特定のバーチャル プライベート ネットワーク（VPN）に H.323 隣接を関連付けます。