セグメント ルーティングの設定

この章では、セグメント ルーティングの設定方法について説明します。

- セグメント ルーティングについて （1 ページ）
- セグメント ルーティングの注意事項と制限事項 （3 ページ）
- セグメント ルーティングの設定 （7 ページ）
- IS-IS プロトコルでのセグメント ルーティングの設定 （19 ページ）
- OSPFv2 プロトコルでのセグメント ルーティングの設定 （20 ページ）
- トラフィック エンジニアリング用のセグメント ルーティングの設定 （27 ページ）
- SR-TE 手動プレファレンス選択の設定 （41 ページ）
- SRTE フローベース トラフィック ステアリングの構成 （45 ページ）
- SRTE ポリシーの MPLS OAM モニタリングの構成 （63 ページ）
- セグメント ルーティングでの出力ピア エンジニアリングの設定 （75 ページ）
- セグメント ルーティング MPLS 上のレイヤ 2 EVPNの設定 （84 ページ）
- セグメント ルーティングの VNF の比例マルチパスの設定 （98 ページ）
- vPC マルチホーミング （100 ページ）
- セグメント ルーティング MPLS を介したレイヤ 3 EVPN およびレイヤ 3 VPN の構成 （102 ページ）
- セグメント ルーティング MPLS および GRE トンネルの設定 （117 ページ）
- レイヤ 3 EVPN の SR-TE の確認 （120 ページ）
- セグメント ルーティングの設定の確認 （122 ページ）
- SRTE 明示パス エンドポイント置換の構成 （124 ページ）
- デフォルト VRF を介した SRTE の構成 （128 ページ）
- その他の参考資料 （147 ページ）

セグメント ルーティングについて

セグメント ルーティングは、ソース ルーティングと同様に、パケットがたどるパスをパケット自体にエンコードする手法です。ノードは、制御された一連の命令（セグメント）によってパケットをステアリングするために、パケットの前にセグメント ルーティング ヘッダーを付

加する各セグメントを識別するセグメント ID（SID）は、フラットな 32 ビットの符号なし整数からなる

セグメントのサブクラスであるボーダー ゲートウェイ プロトコル（BGP）セグメントは、BGP 転送命令を識別します。BGP セグメントには、プレフィックス セグメントと隣接セグメントの 2 つのグループがあります。プレフィックス セグメントは、利用可能なすべての等コスト マルチパス（ECMP）パスを使用して、宛先への最短パスを通るようパケットを誘導します。

隣接セグメントは、パケットをネイバーへの特定のリンクに誘導します。

セグメント ルーティング アーキテクチャは、MPLS データ プレーンに直接適用される

セグメント ルーティング アプリケーション モジュール

セグメントルーティングアプリケーション（SR-APP）モジュールは、セグメントルーティング機能を構成するために使用されます。セグメントルーティングアプリケーション（SR-APP）は、セグメントルーティングに関連するすべてのCLIを処理する独立した内部プロセスです。SRGB範囲を予約し、それについてクライアントに通知する役割を担います。また、プレフィックスから SID へのマッピングの維持も担当します。SR-APP サポートは、BGP、IS-IS、および OSPF プロトコルでも利用できます。

SR-APP モジュールは、以下の情報を保持します。

- セグメント ルーティングの動作状態
- セグメント ルーティングのグローバル ブロック範囲
- プレフィックス SID マッピング

詳細については、「セグメント ルーティングの設定 （7 ページ）」を参照してください。

MPLS の NetFlow

NetFlow は入力 IP パケットについてパケット フローを識別し、これらのパケット フローに基づいて統計情報を提供します。NetFlowのためにパケットやネットワーキング デバイスを変更する必要はありません。フロー用に NetFlow が収集したデータをエクスポートするには、フロー エクスポータを使用し、このデータを Cisco Stealthwatch などのリモート NetFlow コレクタにエクスポートします。Cisco NX-OS は、NetFlow エクスポート用のユーザ データグラム プロトコル（UDP）データグラムの一部としてフローをエクスポートします。フロー用にNetFlow が収集したデータをエクスポートするには、フローエクスポータを使用し、このデータをCisco Stealthwatch などのリモートNetFlow コレクタにエクスポートします。Cisco NX-OS は、NetFlow エクスポート用のユーザ データグラム プロトコル（UDP）データグラムの一部としてフローをエクスポートします。

Cisco NX-OS リリース 9.3(1) 以降、セグメント ルーティング上の NetFlow Collector は、Cisco Nexus 9300-EX、9300-FX、9300-FX2、9500-EX、および9500-FX プラットフォーム スイッチでサポートされます。

Cisco NX-OS リリース 9.3(5) 以降、セグメント ルーティング上の NetFlow Collector は、Cisco Nexus 9300-FX3 プラットフォーム スイッチでサポートされます。

Netflow は Cisco Nexus 9300-GX プラットフォーム スイッチではサポートされません。

NetFlow Collector は、シングルおよびダブル MPLS ラベルの両方をサポートします。エクスポータの宛先設定のデフォルトおよび非デフォルト VRF の両方がサポートされます。NetFlow は、MPLS データ パスをサポートしていません。

セグメントルーティングは単一のラベルをサポートしないため、BGP ネイバーで **address-family ipv4labeled-unicast** コマンドを設定し、bgp 設定で **allocate-label** コマンドを設定する必要があります。

sFlow コレクタ

サンプリングされた Flow（sFlow）を使用すると、スイッチやルータを含むデータネットワーク内のリアルタイムトラフィックをモニターできます。sFlow では、トラフィックをモニタするためにスイッチとルータ上の sFlow エージェント ソフトウェアでサンプリング メカニズムを使用して、サンプル データを中央のデータ コレクタに転送します。

Cisco NX-OS リリース 9.3(1) 以降、セグメント ルーティング上の sFlow コレクタは Cisco Nexus 9300-EX、9300-FX、9300-FX2、9500-EX、および9500-FX プラットフォーム スイッチでサポートされます。

Cisco NX-OS リリース 9.3(5) 以降、セグメント ルーティング上の sFlow コレクタは Cisco Nexus 9300-FX3 プラットフォーム スイッチでサポートされます。

sFlow は Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX スイッチではサポートされていません。

sFlow 設定の詳細については、「*sFlow* の設定」のセクションを参照してください。『*Cisco Nexus 9000* シリーズ *NX-OS* システム管理設定ガイド、リリース *9.3(x)*』に掲載されています。

セグメント ルーティングの注意事項と制限事項

セグメント ルーティングに関する注意事項および制約事項は、次のとおりです。

- MPLS セグメント ルーティングは、FEX モジュールではサポートされていません。
- Cisco NX-OS リリース 9.3(1) 以降、**segment-routing mpls** コマンドは **segment-routing** に変更されました。
- -R シリーズ ラインカードを搭載した Cisco Nexus 9504 および 9508 プラットフォーム スイッチで MPLS セグメント ルーティングを有効にすると、BFD セッションがダウンしたり、戻ったりする場合があります。BGP ピアリングも、BFD で構成されている場合、ダウンしてからアップします。BGPセッションがダウンすると、ハードウェアからルートが取り消されます。これにより、BGPセッションが再確立されてルートが再インストールされるまで、パケット損失が発生します。ただし、いったん BFD が起動すると、追加のフラップは発生しません。

- セグメント ルーティングは、IGP (OSPF など) の下で、または BGP での AF ラベル付きユニキャストによって実行できます。
- セグメント ルーティングは、Cisco Nexus 9300-FX プラットフォーム スイッチおよび Cisco Nexus N9K-X9736C-FX ラインカードでサポートされています。
- セグメント ルーティングと SR-EVPN は、Cisco Nexus C31108PC-V、C31108TC-V、および C3132Q-V スイッチでサポートされています。
- Cisco NX-OS リリース 9.3(3) 以降、Cisco Nexus 9300-GX プラットフォーム スイッチ上ではレイヤ 3 VPN を設定できます。
- Cisco NX-OS リリース 9.3(3) 以降、セグメント ルーティングと SR-EVPN は Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX プラットフォーム スイッチでサポートされています。
- Cisco NX-OS リリース 9.3(3) 以降、隣接関係 SID と OSPF は Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX プラットフォーム スイッチでサポートされています。
- Cisco NX-OS リリース 9.3(3) 以降、OSPF でのセグメント ルーティング、IS-IS アンダーレイ、および BGP ラベル付きユニキャストは Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX プラットフォーム スイッチでサポートされています。
- BGP は、next-hop-self が有効な場合にのみ、iBGP ルート リフレクタ クライアントに SRGB ラベルを割り当てます（たとえば、プレフィックスは、RR 上のローカル IP/IPv6 アドレスの 1 つであるネクスト ホップでアドバタイズされます）。RR で next-hop-self を設定すると、影響を受けるルートのネクスト ホップが変更されます（ルート マップ フィルタリングの対象）。
- Cisco Nexus 9300-EX および 9300-FX プラットフォーム スイッチの MPLS 機能では、無停止の ISSU はサポートされていません。
- スタティック MPLS、MPLS セグメント ルーティング、および MPLS ストリッピングを同時に有効にすることはできません。
- Cisco NX-OS リリース 9.3(5) 以降、MPLS ストリッピングは Cisco Nexus 9300-GX プラットフォーム スイッチでサポートされます。以下の注意事項が当てはまります。
  - MPLS ストリップ機能を動作させるには、スイッチのリロード後に、**mpls strip** および **hardware acl tap-agg** コマンドを設定する必要があります。
  - Cisco Nexus 9300-GX プラットフォーム スイッチで MPLS ストリップが有効になっている場合、ACL ログ プロセスは表示されません。
  - dot1q VLAN を使用した MPLS ストリップはサポートされていません。
  - すべての二重 VLAN タグについて、2 番目の VLAN 範囲は 2 ～ 510 である必要があります。
  - dot1q を使用した MPLS ストリップはサポートされていません。

- PACL リダイレクトをサポートするには、入力タップ インターフェイスで **mode tap-aggregation** コマンドを実行する必要があります。

- スタティック MPLS、MPLS セグメント ルーティング、および MPLS ストリッピングは相互に排他的であるため、マルチホップ BGP の唯一のセグメント ルーティング アンダーレイはシングルホップ BGP です。 eBGP をオーバーレイとして実行する iBGP マルチホップトポロジはサポートされていません。
- 特定のインターフェイスへの転送がその後に続く MPLS ポップはサポートされていません。最後から 2 番目のホップ ポップ（PHP）は、コントロール プレーンが IPv4 黙示的 NULL ラベルをインストールした場合でも、ラベル FIB (LFIB) のアウトラベル として明示的 NULL ラベルをインストールすれば回避できます。
- BGP ラベル付きユニキャストおよび BGP セグメント ルーティングは、IPv6 プレフィックスではサポートされていません。
- BGP ラベル付きユニキャストおよび BGP セグメント ルーティングは、トンネル インターフェイス（GRE および VXLAN を含む）または vPC アクセス インターフェイスではサポートされていません。
- MTU パス ディスカバリ（RFC 2923）は、MPLS ラベル スイッチド パス（LSP）またはセグメント ルーテッド パスではサポートされていません。
- Cisco Nexus 9200 シリーズ スイッチの場合、レイヤ 3 または MPLS 隣接の隣接統計は維持されません。
- Cisco Nexus 9500 シリーズ スイッチの場合、MPLS LSP およびセグメント ルーテッド パスは、サブインターフェイス（ポート チャネルまたは通常のレイヤ 3 ポートのいずれか）ではサポートされていません。
- Cisco Nexus 9500 プラットフォーム スイッチの場合、セグメント ルーティングは非階層ルーティング モードでのみサポートされます。
- BGP 設定コマンドの **neighbor-down fib-accelerate** および **suppress-fib-pending** は、MPLS プレフィックスではサポートされていません。
- RFC 2973 および RFC 3270 で定義されている統一モデルはサポートされていません。したがって、IP DSCP ビットはインポーズされた MPLS ヘッダーにコピーされません。
- セグメント ルーティング グローバル ブロック（SRGB）を再構成すると、BGP プロセスが自動的に再起動され、既存の URIB および ULIB エントリが更新されます。トラフィックの損失は数秒間発生するため、本番環境で SRGB を再構成しないでください。
- セグメント ルーティング グローバル ブロック（SRGB）が範囲に設定されているが、ルート マップ ラベル インデックス デルタ値が設定された範囲外にある場合、割り当てられたラベルは動的に生成されます。たとえば、ルート マップのラベル インデックスが 9000 に設定されているときに SRGB が 16000 ～ 23999 の範囲に設定されている場合、ラベルは動的に割り当てられます。

- ネットワークの拡張性のため、トップオブラック（ToR）または境界リーフ スイッチから接続されているプレフィクスをアドバタイズするマルチホップ BGP とともに階層型ルーティング設計を使用することを推奨します。
- BGP セッションは、MPLS LSP またはセグメント ルーテッド パスではサポートされていません。
- レイヤ 3 転送整合性チェッカーは、MPLS ルートではサポートされていません。
- Cisco Nexus 9000 シリーズ スイッチのオンデマンド ネクスト ホップを使用して、セグメント ルーティング トラフィック エンジニアリングを設定できます。
- セグメント ルーティングのレイヤ 3 VPN およびレイヤ 3 EVPN ステッチングは、Cisco Nexus 9000 シリーズ スイッチでサポートされています。
- Cisco NX-OS リリース 9.3(3) 以降、セグメント ルーティング用のレイヤ 3 VPN およびレイヤ 3 EVPN ステッチングは、9300-GX プラットフォーム スイッチでサポートされています。
- OSPFv2 は、Cisco Nexus 9000 シリーズ スイッチのセグメント ルーティングの IGP コントロール プレーンとして設定できます。
- セグメント ルーティングのレイヤ 3 VPN およびレイヤ 3 EVPN ステッチングは、-EX ラインカードを備えた Cisco Nexus 9364C、9200、9300-EX、および 9500 プラットフォーム スイッチではサポートされていません。
- OSPF セグメント ルーティング コマンドおよびオンデマンド ネクスト ホップを使用したセグメント ルーティング トラフィック エンジニアリングは、Cisco Nexus 9364C スイッチではサポートされていません。
- セグメント ルーティングは、Cisco Nexus 9300-FX2 および 9300-FX3 プラットフォーム スイッチでサポートされています。
- セグメント ルーティングのためのレイヤ 3 VPN およびレイヤ 3 EVPN ステッチング、OSPF セグメント ルーティング コマンド、およびオンデマンド ネクスト ホップを使用したセグメント ルーティング トラフィック エンジニアリングは、Cisco Nexus 9364C スイッチでサポートされています。
- セグメント ルーティングを介したレイヤ 3 VPN は、Cisco Nexus 3100、3200、9200、9300、9300-EX/FX/FX2/FX3 プラットフォーム スイッチ、および EX\FX と R ライン カードを搭載した Cisco Nexus 9500 プラットフォーム スイッチでサポートされています。
- セグメント ルーティング設定を削除すると、MPLS およびトラフィック エンジニアリング設定を含む、関連するすべてのセグメント ルーティング設定が削除されます。
- ブート変数を設定してスイッチをリロードすることによって、Cisco Nexus デバイスを Cisco NX-OS リリース 9.3(1) から以前の NX-OS リリースにダウングレードすると、セグメント ルーティング MPLS の以前の設定がすべて失われます。

- Cisco NX-OS リリース 9.3(1) から ISSD を実行する前に、セグメント ルーティング設定を無効にする必要があります。そうしないと、既存のセグメントルーティング構成が失われます。
- セグメント ルーティング MPLS 隣接統計は、出力ラベル スタックと中間ノードのネクスト ホップに基づいて収集されます。ただし、PHP モードでは、同じスタックがすべての FEC で共有されるため、統計はすべての隣接で表示されます。
- スイッチでセグメント ルーティングが有効になっている場合、dot1Q タグ付き MPLS パケットの Q-in-Q タギングはサポートされておらず、パケットは外部タグのみで出力されます。

  例：VLAN 100 を使用する、アクセス dot1q トンネル モードの入力ポートについて考えます。着信 MPLS トラフィックには、200 の dot1Q タグがあります。通常、トラフィックは外部タグ 100、内部タグ 200 (着信パケットのタグと同じ) で送信されます。ただし、パケットは外部タグ付きで送信され、内部タグは失われます。
- 着信 MPLS パケットにタグが付いておらず、入力ポートがアクセス VLAN モードの場合、セグメント ルーティングが有効になっていれば、パケットはタグなしで出力されます。
- BGP、OSPF、および IS-IS アンダーレイを同時に使用してセグメントルーティングを構成しないことをお勧めします。
- Cisco NX-OS リリース 10.2(1q)F 以降、SR-MPLS は N9K-C9332D-GX2B プラットフォーム スイッチでサポートされます。ただし、SR PBR および MPLS ストリップ dot1q 機能は、GX2 スイッチではまだサポートされていません。

セグメント ルーティングの設定

セグメント ルーティングの設定

始める前に

セグメントルーティングを設定する前に、以下の条件を満たしていることを確認してください。

- **segment-routing** コマンドを構成する前に、**install feature-set mpls**、**feature-set mpls** および **feature mpls segment-routing** コマンドが存在している必要があります。
- グローバルブロックが構成されている場合、指定された範囲が使用されます。それ以外の場合は、デフォルトの 16000 ～ 23999 の範囲が使用されます。
- BGP は、**set label-index**<*value*> 構成と新しい**connected-prefix-sid-map** CLI の両方を使用するようになりました。競合が発生した場合は、SR-APP の構成が優先されます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**segment-routing**
	例：
	`switch(config)# segment-routing`
	`switch(config-sr)# mpls`
	`switch(config-sr-mpls)#`
		MPLS セグメント ルーティング機能を有効にします。このコマンドの **no** 形式は、MPLS セグメント ルーティング機能を無効化します。

ステップ 3	**connected-prefix-sid-map**
	例：
	`switch(config-sr-mpls)#`
	`connected-prefix-sid-map`
	`switch(config-sr-mpls)#`
		接続されたプレフィックス セグメント ID マッピングを設定します。

ステップ 4	**global-block** *<min> <max>*
	例：
	`switch(config-sr-mpls)# global-block`
	`<min> <max>`
	`switch(config-sr-mpls)#`
		セグメント ルーティング バインディングのグローバル ブロック範囲を指定します。

ステップ 5	**connected-prefix-sid-map**
	例：
	`switch(config-sr-mpls)#`
	`connected-prefix-sid-map`
	`switch(config-sr-mpls-conn-pfsid)#`
		接続されたプレフィックス セグメント ID マッピングを設定します。

ステップ 6	**address-family ipv4**
	例：
	`switch(config-sr-mpls-conn-pfsid)#address-family`
	`ipv4`
		IPv4 アドレス ファミリを設定します。

ステップ 7	*<prefix>/<masklen>* [**index**|**absolute**] *<label>*
	例：
	`switch(config-sr-mpls)# 2.1.1.5/32`
	`absolute 201101`
		オプションのキーワード **index** または **absolute** は、入力されたラベル値を SRGB へのインデックスとして解釈するか、絶対値として解釈するかを示します。

例

showコマンドについては、次の設定例を参照してください。

```
switch# show segment-routing mpls
Segment-Routing Global info

Service Name: segment-routing

State: Enabled

Process Id: 29123

Configured SRGB: 17000 - 24999

SRGB Allocation status: Alloc-Successful

Current SRGB: 17000 - 24999

Cleanup Interval: 60

Retry Interval: 180
```

次の CLI は、SR-APP に登録されているクライアントを表示します。クライアントが関心を登録した VRF がリストされます。

```
switch# show segment-routing mpls clients
            Segment-Routing Mpls Client Info

Client: isis-1
    PIB index: 1    UUID: 0x41000118    PID: 29463    MTS SAP: 412
    TIBs registered:
        VRF: default Table: base

Client: bgp-1
    PIB index: 2    UUID: 0x11b    PID: 18546    MTS SAP: 62252
    TIBs registered:
        VRF: default Table: base

Total Clients: 2
```

show segment-routing mpls ipv4 connected-prefix-sid-map CLI コマンドの例では、SRGB は、プレフィックス SID が構成された SRGB 内にあるかどうかを示します。**Indx** フィールドは、構成されたラベルがグローバル ブロックへのインデックスであることを示します。**Abs** フィールドは、構成されたラベルが絶対値であることを示します。

SRGB フィールドに N が表示されている場合は、構成されたプレフィックス SID が SRGB 範囲内になく、SR-APP クライアントに提供されていないことを意味します。SRGB 範囲に入るプレフィックス SID のみが SR-APP クライアントに与えられます。

```
switch# show segment-routing mpls ipv4 connected-prefix-sid-map
            Segment-Routing Prefix-SID Mappings
Prefix-SID mappings for VRF default Table base
Prefix             SID   Type Range SRGB
13.11.2.0/24       713   Indx 1     Y
30.7.7.7/32        730   Indx 1     Y
59.3.24.0/30       759   Indx 1     Y
150.101.1.0/24     801   Indx 1     Y
150.101.1.1/32     802   Indx 1     Y
150.101.2.0/24     803   Indx 1     Y
1.1.1.1/32         16013 Abs  1     Y
```

次の CLI は **show running-config segment-routing** 出力を表示します。

```
switch# show running-config segment-routing ?

> Redirect it to a file
>> Redirect it to a file in append mode
all Show running config with defaults
| Pipe command output to filter

switch# show running-config segment-routing
switch# show running-config segment-routing

!Command: show running-config segment-routing
!Running configuration last done at: Thu Dec 12 19:39:52 2019
!Time: Thu Dec 12 20:06:07 2019

version 9.3(3) Bios:version 05.39
segment-routing
    mpls
        connected-prefix-sid-map
            address-family ipv4
                2.1.1.1/32 absolute 100100

switch#
```

インターフェイス上の MPLS のイネーブル化

MPLS はセグメント ルーティングで使用するインターフェイスで有効にすることができます。

始める前に

MPLS 機能セットは、**install feature-set mpls** および **feature-set mpls** コマンドを使用してインストールし、有効にする必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**interface** *type slot/port*
	例：
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		指定したインターフェイスのインターフェイス コンフィギュレーション モードを開始します。

ステップ 3	[**no**] **mpls ip forwarding** 例： `switch(config-if)# mpls ip forwarding`	指定されたインターフェイスで MPLS を有効にします。このコマンドの **no** 形式は、指定されたインターフェイスで MPLS を無効にします。

	コマンドまたはアクション	目的
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

セグメント ルーティング グローバル ブロックの設定

セグメント ルーティング グローバル ブロック（SRGB）の開始と終了 MPLS ラベルは設定できます。

始める前に

MPLS 機能セットは、**install feature-set mpls** および **feature-set mpls** コマンドを使用してインストールし、有効にする必要があります。

MPLS セグメント ルーティング機能を有効にする必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

ステップ 2	[**no**] **segment-routing**
	例：
	`switch(config)# segment-routing`
	`switch(config-sr)# mpls`
		セグメント ルーティング コンフィギュレーション モードを開始し、16000 ～ 23999 のデフォルトの SRGB を有効にします。このコマンドの **no** 形式は、そのラベル ブロックの割り当てを解除します。
		設定されたダイナミック レンジがデフォルトの SRGB を保持できない場合、エラー メッセージが表示され、デフォルトの SRGB は割り当てられません。必要に応じて、次の手順で別の SRGB を設定できます。

ステップ 3	[**no**] **global-block** *beginning-label ending-label* 例：	SRGB の MPLS ラベル範囲を指定します。このコマンドは、**segment-routing** コマンドで設定されたデフォルトの

	コマンドまたはアクション	目的
	`switch(config-sr-mpls)# global-block 16000 471804`	SRGB ラベル範囲を変更する場合に使用します。 開始 MPLS ラベルと終了 MPLS ラベルの許容値は 16000 ～ 471804 です。**mpls label range** コマンドでは最小ラベルとして 16 が許可されますが、SRGB は 16000 からしか開始できません。 （注） **global-block** コマンドの最小値は 16000 から始まります。以前のリリースからアップグレードする場合は、アップグレードをトリガーする前に、サポートされている範囲内に収まるように SRGB を変更する必要があります。
ステップ 4	（任意） **show mpls label range** 例： `switch(config-sr-mpls)# show mpls label range`	SRGB の割り当てが成功した場合にのみ、SRGB を表示します。
ステップ 5	**show segment-routing**	設定されている SRGB を表示します。
ステップ 6	**show segment-routing mpls** 例： `switch(config-sr-mpls)# show segment-routing mpls`	設定されている SRGB を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config-sr-mpls)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

ラベル インデックスの構成

network コマンドにマッチするルートのラベル インデックスを設定できます。これにより、**set label-index** コマンドを含むルート マップで構成されているローカル プレフィックスに対して BGP プレフィックス SID がアドバタイズされます。ただし、ローカル プレフィックスを指定する **network** コマンドでルート マップが指定されていることが必要です。（**network** コマンドの詳細については、Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide の「Configuring Basic BGP」の章を参照してください)。

（注） セグメントルーティングアプリケーション（SR-APP）モジュールは、セグメントルーティング機能を設定するために使用されます。BGP は、プレフィックス SID の設定のために、ルート マップの下の **set label-index** <*value*> 設定と、新しい **connected-prefix-sid-map** CLI の両方を使用するようになりました。競合が発生した場合には、SR-APP の設定が優先されます。

（注） ルート マップが **network** コマンド以外のコンテキストで指定されている場合、ルートマップ ラベル インデックスは無視されます。また、プレフィックスが **allocate-label route-map** *route-map-name* コマンドで設定されているかどうかに関係なく、ルートマップ ラベル インデックスを使用してプレフィックスにラベルが割り当てられます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**route-map** *map-name*
	例：
	`switch(config)# route-map SRmap`
	`switch(config-route-map)#`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ設定モードを開始します。

ステップ 3	[**no**] **set label-index** *index* 例： `switch(config-route-map)# set label-index 10`	**network** コマンドにマッチするルートのラベル インデックスを設定します。範囲は 0 ～ 471788 です。デフォルトでは、ラベル インデックスはルートに追加されません。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ設定モードを終了します。

ステップ 5	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

	コマンドまたはアクション	目的

ステップ 6	必須: **address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		IPv4アドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 7	**network** *ip-prefix* [**route-map** *map-name*] **例：** `switch(config-router-af)# network 10.10.10.10/32 route-map SRmap`	ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。
ステップ 8	（任意） **show route-map** [*map-name*] **例：** `switch(config-router-af)# show route-map`	ラベル インデックスなど、ルート マップに関する情報を表示します。
ステップ 9	（任意） **copy running-config startup-config** **例：** `switch(config-router-af)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

セグメント ルーティングの構成例

このセクションの例は、2 台のルータ間の一般的な BGP プレフィックス SID 構成を示しています。

この例は、10.10.10.10/32 と 20.20.20.20/32 の BGP スピーカー構成を、それぞれ 10 と 20 のラベル インデックスでアドバタイズする方法を示しています。16000 ～ 23999 のデフォルトのセグメント ルーティング グローバル ブロック （SRGB）範囲を使用します。

```
hostname s1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing
  mpls
  vlan 1
segment-routing
  mpls
    connected-prefix-sid-map
    address-family ipv4
    2.1.1.1/32 absolute 100100

route-map label-index-10 permit 10
```

```
  set label-index 10
route-map label-index-20 permit 10
  set label-index 20

vrf context management
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management

interface loopback1
  ip address 10.10.10.10/32

interface loopback2
  ip address 20.20.20.20/32

line console
line vty

router bgp 1
  address-family ipv4 unicast
    network 10.10.10.10/32 route-map label-index-10
    network 20.20.20.20/32 route-map label-index-20
    allocate-label all
  neighbor 10.1.1.2 remote-as 2
    address-family ipv4 labeled-unicast
```

この例は、BGP スピーカーからの構成を受信する方法を示しています。

```
hostname s2
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.2/24
  ipv6 address 10:1:1::2/64
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management

interface loopback1
  ip address 2.2.2.2/32
```

```
line console

line vty

router bgp 2
  address-family ipv4 unicast
    allocate-label all
  neighbor 10.1.1.1 remote-as 1
    address-family ipv4 labeled-unicast
```

この例は、BGP スピーカーからの構成を表示する方法を示しています。この例の **show** コマンドは、16000 ～ 23999 の SRGB 範囲のラベル 16010 にマッピングされているラベル インデックス 10 のプレフィックス 10.10.10.10 を表示します。

```
switch# show bgp ipv4 labeled-unicast 10.10.10.10/32

BGP routing table information for VRF default, address family IPv4 Label Unicast
BGP routing table entry for 10.10.10.10/32, version 7
Paths: (1 available, best #1)
Flags: (0x20c001a) on xmit-list, is in urib, is best urib route, is in HW, , has label
  label af: version 8, (0x100002) on xmit-list
  local label: 16010

  Advertised path-id 1, Label AF advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
  AS-Path: 1 , path sourced external to AS
    10.1.1.1 (metric 0) from 10.1.1.1 (10.10.10.10)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Prefix-SID Attribute: Length: 10
        Label Index TLV: Length 7, Flags 0x0 Label Index 10

  Path-id 1 not advertised to any peer
  Label AF advertisement
  Path-id 1 not advertised to any peer
```

この例は、BGP スピーカーで出力ピア エンジニアリングを構成する方法を示しています。

```
hostname epe-as-1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface Ethernet1/2
  no switchport
```

```
  ip address 11.1.1.1/24
  no shutdown

interface Ethernet1/3
  no switchport
  ip address 12.1.1.1/24
  no shutdown

interface Ethernet1/4
  no switchport
  ip address 13.1.1.1/24
  no shutdown

interface Ethernet1/5
  no switchport
  ip address 14.1.1.1/24
  no shutdown
```

次に、show ip route vrf 2 コマンドの例を示します。

```
show ip route vrf 2
IP Route Table for VRF "2"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

41.11.2.0/24, ubest/mbest: 1/0
    *via 1.1.1.9%default, [20/0], 13:26:48, bgp-2, external, tag 11 (mpls-vpn)
42.11.2.0/24, ubest/mbest: 1/0, attached
    *via 42.11.2.1, Vlan2, [0/0], 13:40:52, direct
42.11.2.1/32, ubest/mbest: 1/0, attached
    *via 42.11.2.1, Vlan2, [0/0], 13:40:52, local
```

次に、**show forwarding route vrf 2** コマンドの例を示します。

```
slot  1
=======

IPv4 routes for table 2/base

------------------+----------------------------------+-----------------+--------------+--------------
Prefix            | Next-hop                         | Interface       |
Labels            | Partial Install
------------------+----------------------------------+-----------------+--------------+--------------
0.0.0.0/32          Drop                               Null0
127.0.0.0/8         Drop                               Null0
255.255.255.255/32  Receive                            sup-eth1
*41.11.2.0/24       27.1.31.4                          Ethernet1/3
PUSH  30002 492529
                    27.1.32.4                          Ethernet1/21
PUSH  30002 492529
                    27.1.33.4                          port-channel23
PUSH  30002 492529
                    27.11.31.4                         Ethernet1/3.11
PUSH  30002 492529
                    27.11.33.4                         port-channel23.11
PUSH  30002 492529
                    37.1.53.4                          Ethernet1/53/1
```

```
PUSH  29002 492529
                    37.1.54.4                                 Ethernet1/54/1
PUSH  29002 492529
                    37.2.53.4                                 Ethernet1/53/2
PUSH  29002 492529
                    37.2.54.4                                 Ethernet1/54/2
PUSH  29002 492529
                    80.211.11.1                               Vlan801
PUSH  30002 492529
```

次に、**show bgp l2vpn evpn summary** コマンドの例を示します。

```
show bgp l2vpn evpn summary
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 2.2.2.3, local AS number 2
BGP table version is 17370542, L2VPN EVPN config peers 4, capable peers 1
1428 network entries and 1428 paths using 268464 bytes of memory
BGP attribute entries [476/76160], BGP AS path entries [1/6]
BGP community entries [0/0], BGP clusterlist entries [0/0]
476 received paths for inbound soft reconfiguration
476 identical, 0 modified, 0 filtered received paths using 0 bytes

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4    11       0       0        0    0    0 23:01:53 Shut (Admin)
1.1.1.9         4    11    4637    1836 17370542    0    0 23:01:40 476
1.1.1.10        4    11       0       0        0    0    0 23:01:53 Shut (Admin)
1.1.1.11        4    11       0       0        0    0    0 23:01:52 Shut (Admin)
```

次に、**show bgp l2vpn evpn** コマンドの例を示します。

```
show bgp l2vpn evpn 41.11.2.0
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 14.1.4.1:115
BGP routing table entry for [5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224, version 17369591
Paths: (1 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, received and used, is best path
             Imported to 2 destination(s)
  AS-Path: 11 , path sourced external to AS
    1.1.1.9 (metric 0) from 1.1.1.9 (14.1.4.1)
      Origin incomplete, MED 0, localpref 100, weight 0
      Received label 492529
      Extcommunity: RT:2:20

  Path-id 1 not advertised to any peer

Route Distinguisher: 2.2.2.3:113
BGP routing table entry for [5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224, version 17369595
Paths: (1 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path
             Imported from 14.1.4.1:115:[5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224
```

```
AS-Path: 11 , path sourced external to AS
  1.1.1.9 (metric 0) from 1.1.1.9 (14.1.4.1)
```

IS-IS プロトコルでのセグメント ルーティングの設定

IS-IS について

IS-IS は、ISO（国際標準化機構）/IEC（国際電気標準化会議）10589 および RFC 1995 に基づく IGP（内部ゲートウェイ プロトコル）です。Cisco NX-OS は、インターネット プロトコル バージョン 4（IPv4）および IPv6 をサポートします。IS-IS はネットワーク トポロジの変化を検出し、ネットワーク上の他のノードへのループフリー ルートを計算できる、ダイナミック リンクステート ルーティング プロトコルです。各ルータは、ネットワークの状態を記述するリンクステートデータベースを維持し、設定された各リンクにパケットを送信してネイバーを検出します。IS-IS はネットワークを介して各ネイバーにリンクステート情報をフラッディングします。ルータもすべての既存ネイバーを通じて、リンクステートデータベースのアドバタイズメントおよびアップデートを送信します。

IS-IS プロトコルでのセグメント ルーティングは、次をサポートしています。

- IPv4
- レベル 1、レベル 2、およびマルチレベルのルーティング
- プレフィックス SID
- ドメイン ボーダー ノード用の同じループバック インターフェイス上の複数の IS-IS インスタンス
- 隣接関係用の隣接関係 SID

IS-IS プロトコルでのセグメント ルーティングの設定

セグメント ルーティングは IS-IS プロトコルで設定できます。

始める前に

次の条件が満たされると、IS-IS セグメント ルーティングが完全に有効になります。

- **mpls segment-routing** 機能が有効になっていること。
- IS-IS 機能が有効になっていること。
- セグメント ルーティングが、IS-IS の下で少なくとも 1 つのアドレス ファミリに対して有効になっていること。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal**	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router isis** *instance-tag*	instance tag を設定して、新しい IS-IS インスタンスを作成します。
ステップ 3	**net** *network-entity-title*	この IS-IS インスタンスに対応する NET を設定します。
ステップ 4	**address-family** *ipv4* **unicast**	アドレス ファミリ設定モードを開始します。
ステップ 5	**segment-routing mpls**	セグメント ルーティングを IS-IS プロトコルで設定します。 （注） • IS-IS コマンドは、IPv4 アドレス ファミリでのみサポートされます。IPv6 アドレス ファミリではサポートされていません。 • SR プレフィックスの他のプロトコルから ISIS への再配布はサポートされていません。すべてのプレフィックス SID インターフェイスで **ip router isis** コマンドを有効にする必要があります。

OSPFv2 プロトコルでのセグメント ルーティングの設定

OSPF について

Open Shortest Path First（OSPF）は、Internet Engineering Task Force（IETF）の OSPF ワーキング グループによって開発された内部ゲートウェイ プロトコル（IGP）です。OSPF は特に IP ネットワーク向けに設計されており、IP サブネット化、および外部から取得したルーティング情報のタギングをサポートしています。OSPF を使用するとパケット認証も可能になり、パケットを送受信するときに IP マルチキャストが使用されます。

OSPF プロトコルのセグメント ルーティング設定は、プロセス レベルまたはエリア レベルで適用できます。プロセス レベルでセグメント ルーティングを設定すると、すべてのエリアで有効になります。ただし、エリア レベルごとに有効または無効にすることもできます。

OSPF プロトコルでのセグメント ルーティングは、次をサポートしています。

- OSPFv2 のコントロール プレーン
- マルチエリア
- ループバック インターフェイス上のホスト プレフィックスの IPv4 プレフィックス SID
- 隣接関係用の隣接関係 SID

隣接関係 SID のアドバタイズメント

OSPF は、セグメント ルーティング隣接関係 SID のアドバタイズメントをサポートしています。隣接関係セグメント識別子（Adj-SID）は、セグメント ルーティングにおけるルータ隣接関係を表します。

セグメントルーティング対応ルータは、隣接関係ごとにAdj-SIDを割り当てることができ、この SID を拡張不透明リンク LSA で伝送するように Adj-SID サブ TLV が定義されます。

OSPF は、OSPF 隣接関係が 2 つの方法または完全な状態にある場合、各 OSPF ネイバーに隣接関係 SID を割り当てます。OSPF は、セグメント ルーティングが有効になっている場合にのみ隣接関係 SID を割り当てます。隣接関係 SID のラベルは、システムによって動的に割り当てられます。これにより、ローカルでしか有効でないため、設定ミスの可能性がなくなります。

接続されたプレフィックス SID

OSPFv2 は、ループバック インターフェイスに関連付けられたアドレスのプレフィックス SID のアドバタイズをサポートします。これを実現するために、OSPFは、不透明な拡張プレフィックス LSA で拡張プレフィックス サブ TLV を使用します。OSPF がネイバーからこの LSA を受信すると、SR ラベルは、拡張プレフィックス サブ TLV に存在する情報に基づいて、受信したプレフィックスに対応する RIB に追加されます。

設定では、セグメント ルーティングを OSPF で有効にする必要があり、OSPF で設定されたループバック インターフェイスに対応して、セグメントルーティングモジュールでプレフィックス-SID マッピングが必要です。

（注） SID は、ループバック アドレスに対してのみ、またエリア内およびエリア間プレフィックス タイプに対してのみアドバタイズされます。外部プレフィックスまたはNSSAプレフィックスの SID 値はアドバタイズされません。

エリア間のプレフィックス伝播

エリア境界を越えたセグメント ルーティング サポートを提供するには、エリア間で SID 値を伝播するために OSPF が必要です。OSPF は、エリア間のプレフィックス到達可能性をアドバタイズするときに、プレフィックスの SID がアドバタイズされているかどうかを確認します。通常、SID 値はルータから取得され、送信元エリアのプレフィックスへの最適なパスに寄与します。この場合、OSPF はその SID を使用してエリア間でアドバタイズを行います。SID 値がエリア内のベスト パスに寄与するルータによってアドバタイズされない場合、OSPF は送信元エリア内の他のルータからの SID 値を使用します。

セグメント ルーティングのグローバル範囲の変更

OSPF は、SID/ラベル範囲 TLV のアドバタイズに関して、そのセグメント ルーティング機能をアドバタイズします。OSPFv2 では、SID/ラベル範囲 TLV はルータ情報 LSA で伝えられます。

セグメント ルーティングのグローバル範囲設定は、「segment-routing mpls」設定の下にあります。OSPF プロセスが来たら、segment-routing からグローバル範囲の値を取得し、その後の変更はそれに伝播する必要があります。

OSPF セグメント ルーティングが設定されている場合、OSPF は、OSPF セグメント ルーティングの動作状態を有効にする前に、セグメント ルーティング モジュールとのインタラクションをリクエストする必要があります。SRGB 範囲が作成されていない場合、OSPF は有効になりません。SRGB 変更イベントが発生した場合、OSPF は、そのサブブロック エントリで対応する変更を行います。

SID エントリの競合処理

理想的な状況では、各プレフィックスに一意の SID エントリが割り当てられている必要があります。

SID エントリと関連付けられているプレフィックス エントリの間に競合がある場合は、次のいずれかの方法を使用して競合を解決します。

- 1 つのプレフィックスに複数の SID：同じプレフィックスが異なる SID を持つ複数の送信元によってアドバタイズされる場合、OSPF はそのプレフィックスのラベルのないパスをインストールします。OSPF は、到達可能なルータからの SID のみを考慮し、到達不能なルーターからの SID は無視します。1 つのプレフィックスに対して複数の SID がアドバタイズされると、競合と見なされ、そのプレフィックスの接続領域に SID はアドバタイズされません。同様のロジックは、バックボーン エリアと非バックボーン エリアの間でエリア間プレフィックスを伝搬するときにも使用されます。
- SID の範囲外：SID 範囲に収まらない SID の場合、RIB の更新時にラベルは使用されません。

インターフェイスでの MPLS 転送

セグメントルーティングがインターフェイスを使用する前に、MPLS転送を有効にする必要があります。OSPF は、インターフェイスでの MPLS 転送を有効にする役割を担います。

セグメントルーティングが OSPF トポロジに対して有効になっている場合、または OSPF セグメントルーティングの動作状態が有効になっている場合、OSPF は、OSPF トポロジがアクティブである任意のインターフェイスに対して MPLS を有効にします。同様に、OSPF トポロジのセグメント ルーティングが無効になっている場合、OSPF は、そのトポロジのすべてのインターフェイスで MPLS 転送を無効にします。

MPLS転送は、IPIP/GRE トンネルを終端するインターフェイスではサポートされていません。

OSPFv2 でのセグメント ルーティングの設定

セグメント ルーティングを OSPFv2 プロトコルで設定します。

始める前に

OSPFv2 でセグメント ルーティングを設定する前に、次の条件が満たされていることを確認してください。

- OSPFv2 機能が有効になっている。
- セグメント ルーティング機能が有効になっている。
- セグメント ルーティングが OSPF で有効になっている。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	[**no**]**router ospf** *process* 例： `switch(config)# router ospf test`	OSPF モードを有効にします。
ステップ 3	**segment-routing** 例： `switch(config-router)# segment-routing mpls`	OSPF でのセグメント ルーティング機能を設定します。

OSPF ネットワークでのセグメント ルーティングの設定：エリア レベル

始める前に

OSPF ネットワークでセグメント ルーティングを設定する前に、ネットワーク上で OSPF を有効にする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**router ospf process** 例： `switch(config)# router ospf test`	OSPF モードを有効にします。
ステップ 2	**area** <*area id*> **segment-routing** [**mpls** | **disable**] 例： `switch(config-router)# area 1 segment-routing mpls`	特定の領域にセグメント ルーティング MPLS モードを設定します。
ステップ 3	[**no**]**area** <*area id*> **segment-routing** [**mpls** | **disable**] 例： `switch(config-router)#area 1 segment-routing disable`	指定されたエリアのセグメントルーティング mpls モードを無効にします。
ステップ 4	**show ip ospf** プロセス **segment-routing** 例： `switch(config-router)# show ip ospf test segment-routing`	OSPF の下で SR を設定するための出力を示します。

OSPF のプレフィックス SID の設定

ここでは、各インターフェイスでプレフィックスセグメント ID（SID）を設定する方法について説明します。

始める前に

セグメント ルーティングを対応するアドレス ファミリでイネーブルにする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します
ステップ 2	[**no**]**router ospf** *process* 例： `switch(config)# router ospf test`	OSPF を設定します。

ステップ 3	**segment-routing**
	例：
	`switch(config-router)# segment-routing`
	`switch(config-sr)#mpls`
	`switch(config-sr-mpls)#`
		OSPF でのセグメント ルーティング機能を設定します。

ステップ 4	**interface loopback** *interface_number* 例： `switch(config-sr-mpls)# Interface loopback 0`	OSPF が有効になっているインターフェイスを指定します。
ステップ 5	**ip address 1.1.1.1/32** 例： `switch(config-sr-mpls)# ip address 1.1.1.1/32`	ospf インターフェイスで設定された IP アドレスを指定します。
ステップ 6	**ip router ospf 1 area 0** 例： `switch(config-sr-mpls)# ip router ospf 1 area 0`	エリア内のインターフェイスで有効になっている OSPF を指定します。
ステップ 7	**segment-routing** 例： `switch(config-router)#segment-routing (config-sr)#mpls`	SR モジュールの下でプレフィックス SID マッピングを設定します。

ステップ 8	**connected-prefix-sid-map**
	例：
	`switch(config-sr-mpls)# connected-prefix-sid-map`
	`switch(config-sr-mpls-conn-pfxsid)#`
		セグメント ルーティング モジュールの下でプレフィックス SID マッピングを設定します。

	コマンドまたはアクション	目的

ステップ 9	**address-family ipv4**
	例：
	`switch(config-sr-mpls-conn-pfxsid)# address-family ipv4`
	`switch(config-sr-mpls-conn-pfxsid-af)#`
		OSPF インターフェイスで設定されている IPv4 アドレス ファミリを指定します。

ステップ 10	**1.1.1.1/32 index 10** 例： `switch(config-sr-mpls-conn-af)# 1.1.1.1/32 index 10`	SID 100 にアドレス 1.1.1.1/32 を関連付けます。
ステップ 11	**exit** 例： `switch(config-sr-mpls-conn-af)# exit`	セグメントルーティングモードを終了し、コンフィギュレーション端末モードに戻ります。

プレフィックス属性 N-flag-clear の設定

OSPF は、その不透明 LSA に拡張プレフィックス TLV を介してプレフィックス SID をアドバタイズします。これはプレフィックスのフラグを伝送します。そのうちの 1 つは N フラグ（ノード）で、プレフィックスに沿って送信されたトラフィックが、LSA を発信するルータ宛てであることを示します。このフラグは通常、ルータのループバックのホスト ルートをマークします。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**interface loopback3** 例： `switch(config)# interface loopback3`	インターフェイス ループバックを指定します。
ステップ 3	**ip ospf prefix-attributes n-flag-clear** 例： `switch#(config-if)# ip ospf prefix-attributes n-flag-clear`	プレフィックス N-flag をクリアします。

OSPF のプレフィックス SID の設定例

この例は、OSPF のプレフィックス SID の設定を示しています。

```
Router ospf 10
     Segment-routing mpls
Interface loop 0
    Ip address 1.1.1.1/32
    Ip router ospf 10 area 0
Segment-routing
   Mpls
      connected-prefix-sid-m
          address-family ipv4
              1.1.1.1/32 index 10
```

トラフィック エンジニアリング用のセグメント ルーティングの設定

トラフィック エンジニアリング用のセグメント ルーティングについて

トラフィック エンジニアリング用のセグメント ルーティング（SR-TE）は、送信元と宛先のペア間のトンネルを通じて行われます。トラフィック エンジニアリング用のセグメント ルーティングでは、送信元ルーティングの概念が使用されます。送信元はパスを計算し、パケットヘッダーでセグメントとしてエンコードします。トラフィック エンジニアリング（TE）トンネルは、トンネルの入力とトンネルの宛先との間でインスタンス化された TE LSP のコンテナです。TE トンネルは、同じトンネルに関連付けられた 1 つ以上の SR-TE LSP をインスタンス化できます。

トラフィック エンジニアリング用のセグメント ルーティング（SR-TE）では、ネットワークはアプリケーション単位およびフロー単位の状態を維持する必要はありません。代わりに、パケットで提供されている転送指示に従うだけです。

SR-TE は、すべてのセグメント レベルで ECMP を使用することにより、従来の MPLS-TE ネットワークよりも効果的にネットワーク帯域幅を利用します。単一のインテリジェント ソースを使用し、残りのルータをネットワーク経由で必要なパスを計算するタスクから解放します。

SR-TE ポリシー

トラフィック エンジニアリングを実現するためのセグメント ルーティング（SR-TE）では、ネットワークを介してトラフィックを誘導する「ポリシー」を使用します。SR-TE ポリシーは、セグメントまたはラベルのセットを含むコンテナです。このセグメントのリストは、ステートフル PCE であるオペレータによってプロビジョニングされます。ヘッドエンドは、SR-TE ポリシ - を介して伝送されるトラフィック フローに、対応する MPLS ラベル スタックを付します。SR-TE ポリシー パスに沿った各通過ノードは、パケットが最終的な宛先に到達するま

で、着信トップ ラベルを使用してネクストホップを選択し、ラベルをポップまたはスワップし、ラベル スタックの残りの部分を使用して次のノードにパケットを転送します。

SR-TEポリシーは、タプル（カラー、エンドポイント）によって一意に識別されます。カラーは 32 ビットの数値で表され、エンドポイントは IPv4 です。すべての SR-TE ポリシーにはカラー値があります。同じノード ペア間の各ポリシーには、一意のカラー値が必要です。ポリシーに異なるカラーを選択することで、同じ2つのエンドポイント間で複数のSR-TEポリシーを作成できます。

Cisco Nexus 9000 シリーズ スイッチは、次の 2 種類の SR-TE ポリシーをサポートしています。

- ダイナミック SR-TE ポリシー：SR-TE ポリシー構成またはオンデマンド カラー構成でダイナミック パス プリファレンスを構成すると、パス計算エンジン（PCE）が宛先アドレスへのパスを計算します。PCE でのダイナミック パス計算の結果、ヘッドエンド SR-TE ポリシーに適用されるセグメント/ラベルのリストが生成されます。したがって、トラフィックは、SR-TEポリシーが保持するセグメントにヒットすることによってネットワークを介してルーティングされます。
- 明示SR-TEポリシー：明示パスはラベルのリストであり、明示パスのノードまたはリンクを示します。この機能をイネーブルにするには、**explicit-path** コマンドを使用します。このコマンドにより、明示パスを作成し、パスを指定するためのコンフィギュレーション サブモードを開始できます。

SR-TE ポリシー パス

SR-TEポリシー パスは、セグメントID（SID）リストと呼ばれるパスを指定するセグメントのリストです。すべての SR-TE ポリシーは、動的パスまたは明示パスのいずれかである 1 つ以上の候補パスで構成されます。SR-TE ポリシーは1 つのパスをインスタンス化します。この選択されたパスが優先される有効な候補パスとなります。

動的パス オプションを使用してオンデマンドでカラーを追加し、同じカラーとエンドポイントに対して明示的なパス オプションを使用して明示的なポリシー構成を追加することもできます。この場合、単一のポリシーがヘッドエンドで作成され、設定された優先番号が最も高いパスがトラフィックの転送に使用されます。

SR-TE ポリシー パスの計算には、以下の 2 つの方法が使用されます。

- 動的パス：オンデマンド カラー構成またはポリシー構成でパス プリファレンスを構成するときに動的 PCEP オプションを指定すると、パス計算はパス計算エンジン（PCE）委任されます。
- 明示的なパス：このパスは明示的に指定された SID リストまたは SID リストのセットです。

Cisco NX-OS リリース 10.2(2)F 以降では、SR-TE ポリシーをロックダウンまたはシャットダウンするか、その両方を実行すること、 SR-TE ポリシーまたはオンデマンド カラー テンプレートのシャットダウン設定を行うこと、特定の優先順位を SRTE ポリシーのアクティブ パス オプションに強制すること、または、すべてまたは特定の SRTE ポリシーのパスの再最適化を強制することができます。この機能は、Cisco Nexus 9300-EX、9300-FX、9300-FX2、9300-GX、

および N9K-C9332D-GX2B プラットフォーム スイッチでサポートされています。詳細については、SR-TE 手動プレファレンス選択の設定 （41 ページ）を参照してください。

リリース 7.0(3)I7(1) から現在のリリースまでのさまざまな機能をサポートする Cisco Nexus 9000 スイッチの詳細については、Nexus スイッチ プラットフォーム サポート マトリックスを参照してください。

アフィニティおよびディスジョイント制約について

アフィニティ制約：パス計算エンジン（PCE）にアドバタイズされるリンクには、属性を割り当てることができます。SRTE プロセスは、アフィニティ マップとインターフェイス レベルの構成をホストします。ルーティング プロトコル (IGP) はインターフェイスの更新を登録し、SRTE は IGP にインターフェイスの更新を通知します。IGP tlv は BGP に渡され、外部ピアにアドバタイズされます。アフィニティ制約には 3 つのタイプがあります。

- exclude-any: 指定されたアフィニティ カラーのいずれかを持つリンクをパスが通過してはならないことを指定します。
- include-any: 指定されたアフィニティ カラーのいずれかを持つリンクのみをパスが通過しなければならないことを指定します。したがって、指定されたアフィニティ カラーを持たないリンクを使用してはなりません。
- include-all: 指定されたアフィニティ カラーをすべて持つリンクのみをパスが通過しなければならないことを指定します。したがって、指定されたアフィニティ カラーのすべてを持たないリンクを使用してはなりません。

ディスジョイント制約 - PCE にアドバタイズされる SR-TE ポリシーにディスジョイント制約を割り当てることができます。次に、PCE は、同じアソシエーション グループ ID およびディスジョイントのディスジョイントネス タイプを共有するポリシーに、ディスジョイント パスを提供します。

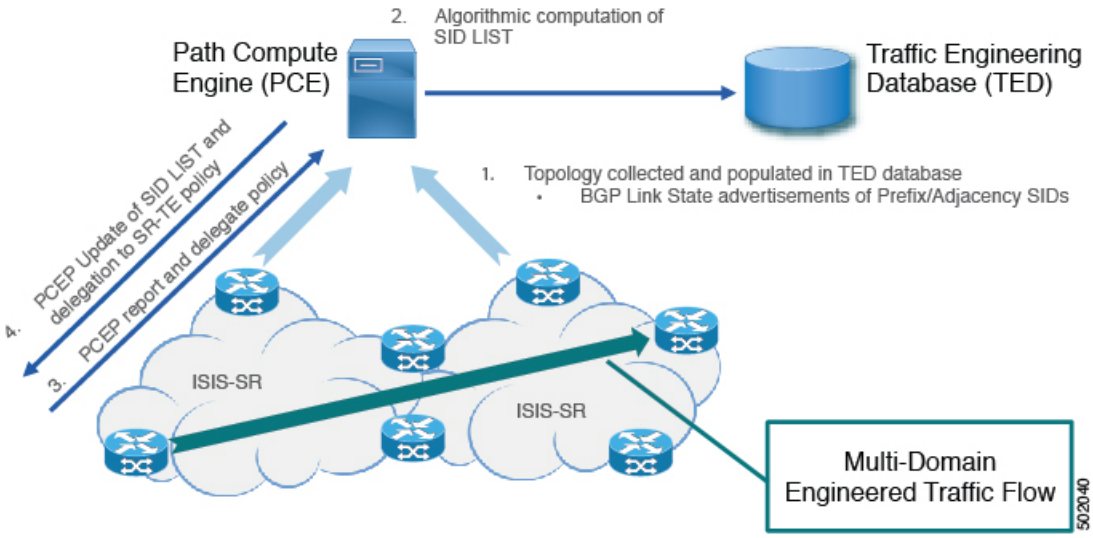
Cisco NX-OS リリース 9.3(1) は、次のディスジョイント パス レベルをサポートします。

- リンク：パスは異なるリンクを通過します（ただし、同じノードを通過する場合があります）。
- ノードのディスジョイントネス：パスは異なるリンクを通過しますが、同じノードを通過する場合があります。

セグメント ルーティング オン デマンド ネクスト ホップ

オン デマンド ネクスト ホップ（ODN）は、BGP ダイナミック SR-TE 機能を活用し、要件に基づいてエンド ツー エンド パスを検索してダウンロードするためのパス計算（PCE）機能を追加します。ODN は定義された BGP ポリシーに基づいて SR-TE 自動トンネルをトリガーします。次の図に示すように、ToR1 と AC1 間のエンド ツー エンド パスは、IGP メトリックに基づいて両端から確立できます。ODN のワークフローは次のようにまとめられます。

図 *1 : ODN* 操作



SR-TE に関する注意事項と制限事項

SR-TE には、次の注意事項と制限事項 があります。

- IPv4 および IPv6 オーバーレイの両方の SR-TE ODN がサポートされています。
- SR-TE ODN は、IS-IS アンダーレイでのみサポートされます。
- 転送では、再帰ネクスト ホップがバインド SID を持つルートに解決される場合、再帰ネクスト ホップを持つルートはサポートされません。
- 転送は、同じルートに対するバインディング ラベルを持つパスとバインディング ラベルのないパスの混合をサポートしていません。
- アフィニティとディスジョイントの制約は、動的な PCEP オプションを持つ SR-TE ポリシーにのみ適用されます。
- XTC は、同じグループ内でディスジョイントになっている 2 つのポリシーのみをサポートします。
- SR-TE アフィニティ インターフェイスを構成する場合、インターフェイス範囲はサポートされません。
- プリファレンスは、動的 PCEP と明示的なセグメント リストの両方を同じプリファレンスに対し一緒に設定することはできません。
- ポリシーごとに動的 PCEP オプションを持つことができるプリファレンスは 1 つだけです。
- 明示的なポリシーについては、同じプリファレンスで ECMP パスを構成する場合、最初のホップ（NHLFE）が両方の ECMP パスで同じであるなら、ULB はスイッチングに 1 つの

パスのみをインストールします。このことは、NHLFE が両方で同じであるため、両方の ECMP パスが同じ SRTE FEC を構築するので発生します。

- Cisco NX-OS リリース 9.3(1) では、アフィニティ設定による非保護モードは PCE（XTC）でサポートされていません。
- Cisco NX-OS リリース 9.3(3) 以降、SR-TE ODN、ポリシー、ポリシー パス、およびアフィニティとディスジョイントの制約は、Cisco Nexus 9364C-GX、Cisco Nexus 9316D-GX、および Cisco Nexus 93600CD-GX スイッチでサポートされています。
- Cisco NX-OS リリース 10.2(2)F 以降、SR-TE ポリシーの新しい show コマンドがいくつか導入されました。また、既存の SR-TE ポリシー コマンドの一部にオートコンプリート機能が提供され、使いやすさが向上しています。この機能は、Cisco Nexus 9300-EX、9300-FX、9300-FX2、9300-GX、および N9K-C9332D-GX2B プラットフォーム スイッチでサポートされています。

（注） リリース 7.0(3)I7(1) から現在のリリースまでのさまざまな機能をサポートする Cisco Nexus 9000 スイッチの詳細については、Nexus スイッチ プラットフォーム サポート マトリックスを参照してください。

SR-TE の設定

トラフィック エンジニアリング用にセグメント ルーティングを設定することができます。

始める前に

mpls セグメント ルーティング機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal**	グローバル コンフィギュレーション モードを開始します
ステップ 2	**segment-routing**	セグメントルーティング モードを開始します。
ステップ 3	**traffic-engineering**	トラフィック エンジニアリング モードに入ります。
ステップ 4	**encapsulation mpls source ipv4** *tunnel_ip_address*	SR-TE トンネルの送信元アドレスを設定します。
ステップ 5	**pcc**	PCC モードに入ります。

	コマンドまたはアクション	目的
ステップ 6	**source-address ipv4** *pcc_source_address*	PCC の送信元アドレスを設定する
ステップ 7	**pce-address ipv4 pce_source_address** *precedence* **num**	PCE の IP アドレスを設定します。最も小さい番号の PCE が優先され、その他はバックアップとして使用されます。
ステップ 8	**on-demand color** *color_num*	オンデマンドモードに入り、カラーを設定します。
ステップ 9	**candidate-paths**	ポリシーの候補パスを指定します。
ステップ 10	**preference** *preference_number*	候補パスの優先順位を指定します。
ステップ 11	**dynamic**	パス オプションを指定します。
ステップ 12	**pcep**	PCE から実行する必要があるパス計算を指定します。

アフィニティ制約の設定

SR-TE ポリシーに対するアフィニティ制約を設定できます。

始める前に

mpls セグメント ルーティング機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**segment-routing**
	例：
	`switch(config)# segment-routing`
	`switch(config-sr)#`
		MPLS セグメント ルーティング機能を有効にします。

ステップ 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィック エンジニアリング モードに入ります。

ステップ 4	**pcc**	PCC モードに入ります。

	コマンドまたはアクション	目的
ステップ 5	**source-address ipv4** *pcc_source_address*	PCC の送信元アドレスを設定する
ステップ 6	**pce-address ipv4 pce_source_address** *precedence* **num**	PCE の IP アドレスを設定します。 最も小さい番号の PCE が優先され、その他はバックアップとして使用されます。

ステップ 7	**affinity-map**
	例：
	`switch(config-sr-te)#affinity-map`
	`switch(config-sr-te-affmap)#`
		アフィニティマップ コンフィギュレーション モードを設定します。

ステップ 8	**color** *name* **bit-position** *position*
	例：
	`switch(config-sr-te-affmap)# color red bit-position 2`
	`switch(config-sr-te-affmap)#`
		アフィニティ ビットマップ内の特定のビット位置へのユーザー定義名のマッピングを構成します。

ステップ 9	**interface** *interface-name*
	例：
	`Enter SRTE interface config mode`
	`switch(config-sr-te-if)#interface eth1/1`
	`switch(config-sr-te-if)#`
		インターフェイスの名前を指定します。これは、アフィニティ ビットマップの特定のビットを参照するアフィニティ マッピング名です。

ステップ 10	**affinity**
	例：
	`switch(config-sr-te-if)# affinity`
	`switch(config-sr-te-if-aff)#`
	`switch(config-sr-te-if-aff)# color red`
	`switch(config-sr-te-if-aff)#`
		インターフェイスにアフィニティ カラーを追加します。

ステップ 11	**policy** *name* | **on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 211`
	または
	`switch(config-sr-te-color)# policy test_policy`
		ポリシーを設定します。

ステップ 12	**color** *color end-point address* 例： `switch(config-sr-te-pol)#color 200 endpoint 2.2.2.2`	ポリシーのカラーとエンドポイントを設定します。これは、「ポリシー名」設定モードを使用してポリシーを設定するときに必要です。

	コマンドまたはアクション	目的

ステップ 13	**candidate-path**
	例：
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		ポリシーの候補パスを指定します。

ステップ 14	**preference** *preference_number*
	例：
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 15	**dynamic**
	例：
	`switch(cfg-pref)# dynamic`
	`switch(cfg-dyn)#`
		パス オプションを指定します。

ステップ 16	**pcep**
	例：
	`switch(cfg-dyn)# pcep`
	`switch(cfg-dyn)#`
		ヘッドエンドが PCEP を使用して、それ自体からセグメントルーティングのポリシーのエンドポイントまでのパスを計算するようにPCEに要求することを指定します。

ステップ 17	**constraints**
	例：
	`switch(cfg-dyn)# constraints`
	`switch(cfg-constraints)#`
		候補パス優先制約モードに入ります。

ステップ 18	**affinity**
	例：
	`switch(cfg-constraints)# affinity`
	`switch(cfg-const-aff)#`
		ポリシーのアフィニティ制約を指定します。

ステップ 19	**exclude-any** |**include-all** | **include-any**
	例：
	`switch(cfg-const-aff)# include-any`
	`switch(cfg-aff-inclany)#`
		アフィニティ制約タイプを指定します。次のアフィニティタイプを使用できます。
		• exclude-any - 指定されたアフィニティカラーのいずれかを持つリンクをパスが通過してはならないことを指定します。
		• include-any - 指定されたアフィニティカラーのいずれかを持つリンクのみをパスが通過する必要があることを指定します。

	コマンドまたはアクション	目的
		• include-all - 指定されたアフィニティ カラーをすべて持つリンクのみをパスが通過する必要があることを指定します。

ステップ 20	**color** *color_name*
	例：
	`switch(cfg-aff-inclany)# color blue`
	`switch(cfg-aff-inclany)#`
		アフィニティ カラーの定義を指定します。

ディスジョイント パスの構成

SR-TE ポリシーに対するディスジョイント制約を設定できます。

始める前に

mpls セグメント ルーティング機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**segment-routing**
	例：
	`switch(config)# segment-routing`
	`switch(config-sr)#`
		MPLS セグメント ルーティング機能を有効にします。

ステップ 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィック エンジニアリング モードに入ります。

ステップ 4	**pcc**	PCC モードに入ります。
ステップ 5	**source-address ipv4** *pcc_source_address*	PCC の送信元アドレスを設定する
ステップ 6	**pce-address ipv4 pce_source_address** *precedence* **num**	PCE の IP アドレスを設定します。

	コマンドまたはアクション	目的
		最も小さい番号のPCEが優先され、その他はバックアップとして使用されます。

ステップ 7	**policy** *name* | **on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 211`
	または
	`switch(config-sr-te-color)# policy test_policy`
		ポリシーを設定します。

ステップ 8	**color** *color end-point address* 例： `switch2(config-sr-te-pol)# color 200 endpoint 2.2.2.2`	ポリシーのカラーとエンドポイントを設定します。これは、「ポリシー名」設定モードを使用してポリシーを設定するときに必要です。

ステップ 9	**candidate-path**
	例：
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		ポリシーの候補パスを指定します

ステップ 10	**preference** *preference_number*
	例：
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 11	**dynamic**
	例：
	`switch(cfg-pref)# dynamic`
	`switch(cfg-dyn)#`
		パス オプションを指定します。

ステップ 12	**pcep**
	例：
	`switch(cfg-dyn)# pcep`
	`switch(cfg-dyn)#`
		ヘッドエンドがPCEPを使用して、それ自体からセグメントルーティングのポリシーのエンドポイントまでのパスを計算するようにPCEに要求することを指定します。

ステップ 13	**constraints**
	例：
	`switch(cfg-dyn)# constraints`
	`switch(cfg-constraints)#`
		候補パス優先制約モードに入ります。

ステップ 14	**association-group** 例：	アソシエーション グループ タイプを指定します。

	コマンドまたはアクション	目的
	`switch(cfg-constraints)# association-group switch(cfg-assoc)#`	
ステップ 15	**disjoint** 例： `switch(cfg-assoc)# disjoint switch(cfg-disj)#`	ディスジョイントネスアソシエーション グループに属するパスを指定します。
ステップ 16	**type** | **link** | **node** 例： `switch(config-if)#type link`	ディスジョイントネス グループ タイプを指定します。
ステップ 17	**id** *number* 例： `switch(config-if)#id 1`	アソシエーション グループの識別子を指定します。

SR-TE の設定例

このセクションの例は、アフィニティおよびディスジョイントの設定を示しています。

この例は、ユーザー定義名から管理グループへのマッピングを示しています。

```
segment-routing
 traffic-eng
  affinity-map
   color green bit-position 0
   color blue bit-position 2
   color red bit-position 3
```

この例では、eth1/1 の隣接のアフィニティ リンクの色が赤と緑、eth1/2 の隣接のアフィニティ リンクの色が緑であることを示しています。

```
segment-routing
 traffic-eng
  interface eth1/1
   affinity
    color red
    color green
  !
  interface eth1/2
   affinity
    color green
```

この例は、ポリシーのアフィニティ制約を示しています。

```
segment-routing
  traffic-engineering
    affinity-map
      color blue bit-position 0
      color red  bit-position 1
    on-demand color 10
      candidate-paths
        preference 100
          dynamic
```

```
          pcep
        constraints
          affinity
            [include-any|include-all|exclude-any]
              color <col_name>
              color <col_name>
    policy new_policy
      color 201 endpoint 2.2.2.0
      candidate-paths
        preference 200
          dynamic
            pcep
          constraints
            affinity
              include-all
                color red
```

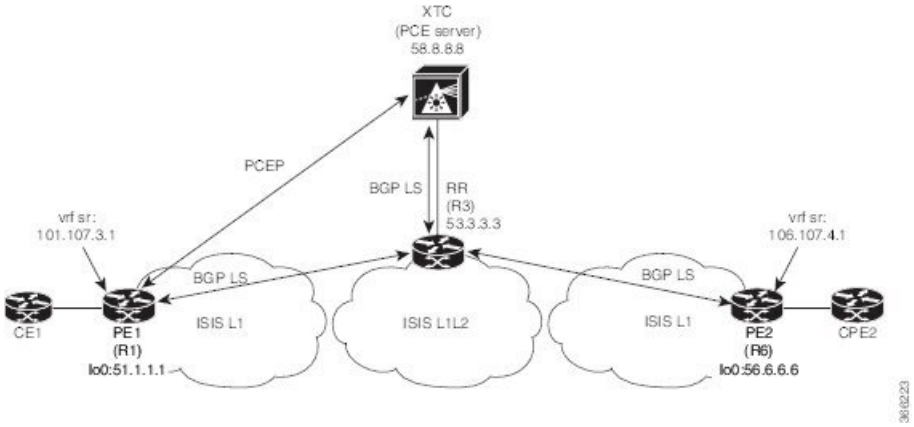
この例は、ポリシーのディスジョイント制約を示しています。

```
segment-routing
 traffic-eng
  on-demand color 99
   candidate-paths
     preference 100
       dynamic
         pcep
       constraints
         association-group
           disjoint
             type link
             id 1
```

SR-TE ODN の設定例 - ユースケース

SR-TE の ODN を設定するには、次のステップを実行します。設定ステップを説明するため、次の図を参考として使用します。

図 *2:* 参照トポロジ



1. PE1 から PE2 への IS-IS ポイントツーポイントセッションですべてのリンクを設定します。また、上記のトポロジーに従ってドメインを設定します。
2. R1、R3、および R6 の IS-IS セッションに対して「リンク状態の配布」を有効にします。

```
router isis 1
  net 31.0000.0000.0000.712a.00
  log-adjacency-changes
  distribute link-state
  address-family ipv4 unicast
    bfd
    segment-routing mpls
    maximum-paths 32
    advertise interface loopback0
```

3. ルータ R1（ヘッドエンド）と R6（テールエンド）に VRF インターフェイスを設定します。

   R1 上の VRF 設定：

```
interface Ethernet1/49.101
encapsulation dot1q 201
  vrf member sr
  ip address 101.10.1.1/24
  no shutdown

vrf context sr
  rd auto
  address-family ipv4 unicast
    route-target import 101:101
    route-target import 101:101 evpn
    route-target export 101:101
    route-target export 101:101 evpn
router bgp 6500
  vrf sr
    bestpath as-path multipath-relax
    address-family ipv4 unicast
      advertise l2vpn evpn
```

4. R6（テールエンド）での BGP コミュニティで VRF プレフィックスをタグ付けします。

```
route-map color1001 permit 10
  set extcommunity color 1001
```

5. R6（テールエンド）およびR1（ヘッドエンド）上のBGPを有効にしてVRF SRプレフィックスのアドバタイズと受信を行い、R6（テールエンド）上のコミュニティ設定とマッチングします。

   R6 < EVPN > R3 < EVPN > R1

   BGP の設定 R6：

```
router bgp 6500
  address-family ipv4 unicast
     allocate-label all
  neighbor 53.3.3.3
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
     route-map Color1001 out
      encapsulation mpls
```

   BGP の設定 R1：

```
router bgp 6500
  address-family ipv4 unicast
```

```
     allocate-label all
  neighbor 53.3.3.3
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
       encapsulation mpls
```

6. R3 での BGP 構成と、R1、R3.abd での XTC による BGP LS の有効化

   BGP の設定 R3：

```
router bgp 6500
  router-id 2.20.1.2
address-family ipv4 unicast
allocate-label all
address-family l2vpn evpn
retain route-target all
  neighbor 56.6.6.6
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
       route-reflector-client
       route-map NH_UNCHANGED out
      encapsulation mpls
  neighbor 51.1.1.1
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
      route-reflector-client
      route-map NH_UNCHANGED out
      encapsulation mpls
neighbor 58.8.8.8
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family link-state

route-map NH_UNCHANGED permit 10
  set ip next-hop unchanged
```

   BGP の設定 R1：

```
router bgp 6500
neighbor 58.8.8.8
              remote-as 6500
               log-neighbor-changes
               update-source loopback0
               address-family link-state
```

   BGP の設定 R6：

```
outer bgp 6500
   neighbor 58.8.8.8
    remote-as 6500
       log-neighbor-changes
       update-source loopback0
       address-family link-state
```

7. R1 で PCE および SR-TE トンネル設定を有効にします。

```
segment-routing
  traffic-engineering
    pcc
      source-address ipv4 51.1.1.1
      pce-address ipv4 58.8.8.8
    on-demand color 1001
      metric-type igp
```

SR-TE 手動プレファレンス選択の設定

このセクションでは、手動プレファレンス選択機能をサポートするために導入された設定および実行コマンドについて説明します。

SR-TE 手動優先順位選択の注意事項と制限事項

次の注意事項と制限事項は、SR-TE 手動優先順位選択機能に適用されます。

- Cisco NX-OS リリース 10.2(2)F 以降、SR-TE の手動優先順位選択機能により、SRTE ポリシーまたはオンデマンド カラー テンプレートの両方でロックダウン、シャットダウン、またはその両方を実行できます（SR-TE ポリシーまたはオンデマンド カラー テンプレートのシャットダウン優先順位）。さらに、この機能により、SR-TE ポリシーに対して特定の優先順位を強制的にアクティブにし、すべてまたは特定の SR-TE ポリシーに対してパスの再最適化を強制することもできます。

  この機能は、Cisco Nexus 9300-EX、9300-FX、9300-FX2、9300-GX、および N9K-C9332D-GX2B プラットフォーム スイッチでサポートされています。

SR-TE 手動設定について：ロックダウンとシャットダウン

Cisco NX-OS リリース 10.2(2)F 以降、必要に応じて次のアクションを実行できます。

- SRTE ポリシーのロックダウン：オンデマンドのカラー テンプレートまたは明示的なポリシーでロックダウンを有効にできます。ロックダウンは、ポリシーのパス設定の自動再最適化を無効にします。ロックダウンされたポリシーに対して新しい優先パスが発生した場合、新しいパスを使用するように自動的に切り替えることはなく、有効になるまで現在のアクティブなパス オプションを使用し続けます。

（注） オンデマンドテンプレートと同じカラーの明示ポリシー構成が存在する場合、ポリシー構成はロックダウンのテンプレート構成よりも優先されます。

例

ポリシーに複数の設定があるシナリオを考えてみましょう。ネットワークの障害により、優先度の高いパスがダウンしたと仮定します。障害は、優先度の高いパスにあるノードの

差し迫った障害である可能性があります。障害を調査して修正するとき、運用チームは問題のあるノードをリロードまたは無効にして、これが発生している間の中断を防ぐ必要がある場合があります。次に、優先度の低いパスをロックダウンし、優先度の高いパスに戻らないようにすることは、使用するのに適したオプションです。

- SRTE ポリシーのシャットダウン：オンデマンドのカラー テンプレートまたは明示ポリシーでシャットダウンを有効にすることができます。ポリシーの状態が管理状態ダウンに変わり、ポリシーに関係するすべてのクライアントにポリシー ダウン通知が送信されます。オンデマンドのカラー構成でシャットダウンを無効にすると、ポリシーのパスの有効性に基づいて、ポリシーの状態がアップまたはダウンに変更されます。

（注） オンデマンドテンプレートと同じ色の明示ポリシー設定が存在する場合、シャットダウンのテンプレート構成よりもポリシー構成が優先されます。

- SRTE ポリシーのシャットダウン設定 – オンデマンドのカラー テンプレート構成または明示ポリシー構成のパス設定で、パス設定をシャットダウンできます。これにより、そのパスプリファレンスが無効になり、プリファレンスが解除されるまで、将来のパスの再最適化が開始されなくなります。パスプリファレンスは、設定でシャットダウンされているかシャットダウンされていないかに基づいて、`show srte policy` の出力に管理状態ダウンまたはアップとして表示されます。

SR-TE 手動設定の構成 - ロックダウン/シャットダウン

SR-TEポリシーまたはオンデマンドカラーテンプレートで、ロックダウン、シャットダウン、またはその両方を構成できます。SR-TE ポリシーまたはオンデマンド カラー テンプレートの下で構成をシャットダウンすることもできます。

始める前に

mpls セグメント ルーティング機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal**	グローバル コンフィギュレーション モードを開始します
ステップ 2	**segment-routing**	セグメントルーティング モードを開始します。
ステップ 3	**traffic-engineering**	トラフィック エンジニアリング モードに入ります。

	コマンドまたはアクション	目的
ステップ 4	**on-demand color***color_num* または **policy** *name*	オンデマンド モードを開始し、カラーを構成します または SR-TE ポリシーを個別に構成します。
ステップ 5	（オプション） [**no**] **lockdown**	オンデマンドのカラー テンプレートまたは明示的なポリシー構成でロックダウンを有効にします。 （注） オンデマンド テンプレートと同じ色の明示的なポリシー構成が存在する場合、ポリシー構成がテンプレート構成よりも優先され、ポリシーがロックダウンされます。
ステップ 6	（オプション） [**no**] **shutdown**	必要に応じて、オンデマンド カラー テンプレートまたは構成済みの SR-TE ポリシーから作成されたポリシーをシャットダウンします。 （注） オンデマンド テンプレートと同じ色の明示的なポリシー構成が存在する場合、ポリシー構成がテンプレート構成よりも優先され、ポリシーがシャットダウンされます。
ステップ 7	**candidate-paths**	ポリシーの候補パスを指定します。
ステップ 8	**preference** *preference_number*	候補パスの優先順位を指定します。
ステップ 9	（オプション） [**no**] **shutdown**	SR-TE ポリシー構成またはオンデマンド カラー テンプレート構成の下でパス プリファレンスをシャットダウンします。

SRTE ポリシーの特定のパス設定を適用する

特定の設定を SRTE ポリシーのアクティブ パス オプションに適用するには、`segment-routing traffic-engineering switch name <policy_name> pref <preference_number>` 実行コマンドを使用します。このコマンドは、有効になるまで設定を使用します。

次のような出力例を示します。

```
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
Color: 801
Authorized: Y
Binding-sid Label: 22
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:180 ECMP path count: 1
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
NX2# segment-routing traffic-engineering switch name Green_White preference 170
NX2(cfg-pref)# show srte policy Green_white detail
Policy: 8.8.8.0|801
Name: Green_White
…..
Path type = MPLS Path options count: 4
Path-option Preference:180 ECMP path count: 1 Admin: UP Forced: No
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
Path-option Preference:170 ECMP path count: 1 Admin: UP Forced: Yes Active path option
1. Explicit Weighted: No
Name: Yellow
Index: 1 Label: 16006
Index: 2 Label: 16008
```

この手動で選択した設定を元に戻すには、次のオプションのいずれかを実行します。

- `segment-routing traffic-engineering reoptimize name <policy_name>` コマンドを使用します。詳細については、SRTE ポリシーまたはすべての SRTE ポリシーのパス再最適化の適用 （44 ページ）を参照してください。
- 別の設定に切り替えます
- このポリシーを閉じます
- 選択した設定を閉じます

SRTE ポリシーまたはすべての SRTE ポリシーのパス再最適化の適用

SRTE ポリシーに複数の設定がある場合、ポリシーを再最適化でき、利用可能な最適なパスを選択できます。

特定の SRTE ポリシーのパスの再最適化を適用するには、`segment-routing traffic-engineering reoptimize name <policy_name>` コマンドを使用します。`<policy_name>` は、ポリシー名またはエイリアス名にすることができます。このコマンドは、前のセクションで説明した設定スイッチ コマンドを取り消し、構成されている場合はロックダウンをオーバーライドします。

次のような出力例を示します。

```
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
```

```
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
Color: 801
Authorized: Y
Binding-sid Label: 22
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:170 ECMP path count: 1
1. Explicit Weighted: Yes Weight: 1
Name: Yellow
Index: 1 Label: 16006
Index: 2 Label: 16008
NX2# segment-routing traffic-engineering reoptimize name Green_White
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
Color: 801
Authorized: Y
Binding-sid Label: 22
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:180 ECMP path count: 1
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
```

すべてのSRTEポリシーのパスの再最適化を強制するには、`segment-routing traffic-engineering reoptimize all` コマンドを使用して、システムに存在するすべての SRTE ポリシーのパスの再最適化を適用します。このコマンドは、前のポイントで説明した設定スイッチ コマンドを取り消し、構成されている場合はロックダウンをオーバーライドします。

SRTE フローベース トラフィック ステアリングの構成

この章では、Cisco Nexus 9000-FX、9000-FX2、9000-FX3、9000-GX、および 9300 プラットフォーム スイッチで SRTE フローベースのトラフィック ステアリングを構成する方法について説明します。

SRTE フローベース トラフィック ステアリング

Cisco NX-OS リリース 10.1(2) のフローベースのトラフィック ステアリング機能は、直接的で柔軟な、ステアリングするトラフィックを選択する代替方法を提供します。この方法では、出力ノードではなく、ヘッドエンド ノードでソース ルーティングを直接構成できます。フローベースのトラフィック ステアリングにより、ユーザーは、宛先アドレス、UDP または TCP ポート、DSCP ビット、その他のプロパティなどの着信パケットのフィールドを一致させることにより、SRTEポリシーに誘導されるパケットを選択できます。一致は、パケットをポリシーに導くように ACL をプログラミングすることによって行われます。

トラフィックを一致させて誘導するために、ポリシーベース ルーティング（PBR）機能が拡張され、SRTE ポリシーをサポートするようになりました。現在の PBR 機能には、RPM、ACL Manager、および AclQoS コンポーネントが含まれます。Cisco NX-OS リリース 10.1(2) 以降、SRTE サポートを追加するために、RPM コンポーネントは SRTE および ULIB とも通信し、URIB との通信が強化されています。

したがって、SRTE のフローベースのトラフィック ステアリング機能には、次のものが含まれます。

- MPLS SR データプレーン
- IPv4 トラフィックのステアリングはデフォルト VRF でサポートされ、IPv4 および IPv6 トラフィックのステアリングはデフォルト以外の VRF でサポートされます
- 5 つのタプル フィールド（送信元アドレス、宛先アドレス、プロトコル、tcp/udp 送信元ポート、tcp/udp 宛先ポート）の組み合わせに基づく ACL によるトラフィックの一致
- 一致したトラフィックを SRTE ポリシーに導く
- IPv4 パケットのパケット内の DSCP/TOS ビットのマッチング
- IPv6 パケットのパケットのトラフィック クラス フィールドの一致
- 期間の定義に基づく ACL の自動有効化および無効化
- VRF ケースをステアリングするとき、ネクスト ホップを指定せずに SRTE ポリシーへのステアリングをサポートします。
- エニーキャスト エンドポイントを使用したオーバーレイ ECMP
- ACL に一致するパケットは、通常のルートよりも優先されます
- ToS/DSCP およびタイマーベースの ACL に基づくフロー選択
- next-hop-ip は、あるエンドポイントから別のエンドポイントへの SRTE ポリシーへのトラフィックのステアリングに使用されます。

SRTE のフローベース トラフィック ステアリングの注意事項と制限事項

次の注意事項と制限事項は、SRTE 機能のフローベース トラフィック ステアリングに適用されます。

- Cisco NX-OS リリース 10.1(2) 以降、SRTE のフローベースのトラフィック ステアリング機能は、Cisco Nexus 9000-FX、9000-FX2、9000-FX3、9000-GX、および 9300 プラットフォーム スイッチでサポートされます。
- SRTE ポリシーが VRF のインターフェイスに割り当てられたルート マップに適用されるとき（L3VPN/L3EVPN トラフィックを誘導するため）、set statement のネクスト ホップが BGP プレフィックスに解決され、その BGP プレフィックスがすでに SRTE を使用してトラフィックを誘導し、ルートマップはトラフィックを誘導しません。

- アンダーレイ ECMP は、ポリシー内のアクティブな各 SRTE パス（ECMP メンバー）のラベル スタックが同じ場合にのみサポートされます。9000-GX プラットフォームには、この制限はありません。
- ルートマップ トラッキング機能はサポートされていません。
- SRTE ポリシーを操作する場合、1 つのルート マップ シーケンス エントリに複数の**ネクスト ホップを設定**することはサポートされていません。
- SRTE ポリシーが VRF のインターフェイスに割り当てられたルート マップに適用される場合（L3VPN/L3EVPN トラフィックを誘導するため）、set ステートメントのネクスト ホップが RIB で複数のネクスト ホップを有する BGP ルート（オーバーレイ ルート）に対して解決される場合、 トラフィックはルートの最初のネクスト ホップにのみ誘導され、すべてのネクスト ホップで ECMP は行われません。
- SRTE ポリシー名がルート マップ セット ステートメントで使用されている場合、カラーとエンドポイントではなく、デフォルトの VRF ステアリングにのみ使用できます。そうでない場合は、明示的に定義されている SRTE パスを選択する必要があります。具体的には、これは、ラベルの代わりにポリシーエンドポイント キーワードを含むセグメントリストを使用するように定義された SRTE ポリシーを選択するためには使用できません。
- **set ip next-hop <>** で指定されたネクスト ホップ IP に適用される次のキーワードは、SRTE ポリシーにステアリングするときのルート マップではサポートされません。
  - verify-availability
  - drop-on-fail
  - force-order
  - load-share
- 必要な機能（セグメンティング ルーティング、l3 evpn または l3vpn）がデバイスで有効になっていない場合でも、srte-policy を使用したルート マップをインターフェイスに適用できます。ただし、srte-policy を使用した set-actions は抑制されます。つまり、これらのフローに対してデフォルト ルーティングが実行されます。
- ルート マップには、srte-policy ありおよび srte-policy なしの set コマンドを含めることができます。
- srte-policy 情報のない set-command の場合、ステアリングは next-hop-ip への到達可能性が MPLS ラベルを必要としない場合にのみ実行されます。
- ルート マップがデフォルト以外の VRF のインターフェイスに関連付けられており、そのルート マップにネクスト ホップ IP アドレス **N** と SRTE ポリシーを指定するシーケンスが含まれている場合、そのルートマップ上の他のすべてのシーケンスと、同じネクストホップ IP アドレスを使用する同じ VRF に関連付けられたその他すべてのルート マップにも SRTE ポリシーが必要です。同じネクスト ホップ IP と異なる SRTE ポリシーを使用して、別のルート マップまたはルート マップ シーケンスを同じ VRF に関連付けることはできません。

- 同様に、ルート マップがデフォルト以外の VRF のインターフェイスに関連付けられていて、そのルート マップが SRTE ポリシーを指定していないが、ネクスト ホップ IP アドレス **N** を指定している場合、同じネクスト ホップ IP アドレス N を使用し、SRTE ポリシーを指定する、そのルートマップまたは別のルートマップ内の別のシーケンスは適用されません。
- SRTE フローベースのトラフィック ステアリングは、VXLAN または EoMPLS PBR と同時に使用することはできません。
- SRTE 入力ノードのポリシー ベースのルーティング トラフィックでは、SR ラベル統計はサポートされていません。ただし、ACL リダイレクト統計はサポートされています。
- デフォルト VRF の IPv6 トラフィックは、SRTE ポリシーに誘導できません。MPLS SR アンダーレイは、IPv4 でのみサポートされます。ただし、IPv6 SR アンダーレイが必要な場合は、代わりに SRv6 を使用します。
- 9000-FX、9000-FX2、9000-FX3、および 9300 プラットフォーム ハードウェアは、ECMP メンバーごとに一意のアンダーレイ ラベル スタックをプッシュできず、これらのプラットフォームのアンダーレイ ECMP に影響します。つまり、セグメント リストの最初のホップが異なる SRTE ポリシーに複数のアクティブ セグメント リストがある場合（1 つの設定が複数のセグメント リストで構成されている場合）、そのような構成はサポートされません。このような場合、回避策として、エニーキャスト SID を構成して、すべての ECMP メンバーでラベル スタックが同じになるようにします。
- モジュラ プラットフォームは、Cisco NX-OS リリース 10.1(2) ではサポートされていません。
- Cisco NX-OS リリース 10.2(2)F 以降、SRTE のフローベースのトラフィック ステアリング機能は、Cisco N9K-C9332D-GX2B プラットフォーム スイッチでサポートされます。

構成プロセス：SRTE フローベース トラフィック ステアリング

SRTE フローベースのトラフィック ステアリング機能の構成プロセスは次のとおりです。

1. 特に IP アクセス リストの基準に一致する IP アクセス リストを構成します。

   詳細については、『*Cisco Nexus Series NX-OS* セキュリティ構成ガイド』の「*IP ACL* の構成」章を参照してください。

2. SRTE ポリシーを定義します。

   SRTE の設定の詳細については、『*Cisco Nexus 9000* シリーズ *NX-OS* ラベル スイッチ構成ガイド』の「トラフィック エンジニアリング用セグメント ルーティングの構成」の章を参照してください。

3. 一致（ステップ 1 で設定した IP アクセス リスト）とアクションをバインドするルート マップを構成します。一致は、パケットで一致するフィールドを参照し、アクションは、どの SRTE ポリシーを誘導するか、および使用する VPN ラベルを参照します（存在する場合）。

ToS/DSCP およびタイマーベース ACL に基づいたフロー選択の構成

SRTE フローベースのトラフィック ステアリング機能では、フロー選択は ToS/DSCP およびタイマー ベースの ACL に基づいています。

デフォルトおよびデフォルト以外の VRF のルート マップを、さまざまな基準によって選択されたポリシーに構成して正しく動作させるには、次の構成手順を実行します。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	[**ip** | **ipv6**] **access-list** *acl_name*
	例：
	`switch(config)# ip access-list L4_PORT`
	`switch(config)#`
		名前を使用して IP または IPv6 アクセス リストを定義し、 IP または IPv6 アクセスリストコンフィギュレーション モードを開始します。

ステップ 3	**10 permit ip** *ip_address* **any**
	例：
	`switch(config)# 10 permit ip any 5.5.0.0/16`
	`switch(config)#`
		スイッチで構成された IP または IPv6 アクセス リストを表示します。

ステップ 4	**20 permit tcp** *tcp_address* [**any**]
	例：
	`switch(config)# 20 permit tcp any 5.5.0.0/16`
	`switch(config)#`
		IPv6 アクセス リストに TCP 許可条件を設定します。
		（注）
		any キーワードは、IPv6 にのみ使用されます。

ステップ 5	[**ip** | **ipv6**] **access-list** *dscp_name*
	例：
	`switch(config)# ip access-list dscp`
	`switch(config)#`
		名前を使用して IP または IPv6 アクセス リストの DSCP 定義し、 IP または IPv6アクセスリストコンフィギュレーション モードを開始します。

ステップ 6	**10 permit tcp any** *tcp_address* **dscp** <*dscp value*> 例：	IP または IPv6 アクセス リストの DSCP 値を設定します。 （注）

	コマンドまたはアクション	目的
	`switch(config)# 10 permit tcp any 5.5.0.0/16 dscp af11` `switch(config)#`	**any** キーワードは、IPv6 にのみ使用されます。

ステップ 7	[**ip** | **ipv6**] **access-list** *acl_name*
	例：
	`switch(config)# ip access-list acll`
	`switch(config)#`
		名前を使用して IP または IPv6 アクセス リストを定義し、 IP または IPv6 アクセス リスト コンフィギュレーション モードを開始します。

ステップ 8	**10 permit tcp any** *tcp_address* **acl** *acl_name*
	例：
	`switch(config)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11`
	`switch(config)#`
		IPv6 アクセス リストに TCP 許可条件を設定します。
		（注）
		any キーワードは、IPv6 にのみ使用されます。

ステップ 9	[**ip** | **ipv6**] **access-list** *acl_name*
	例：
	`switch(config)# ip access-list acl1`
	`switch(config)#`
		名前を使用して IP または IPv6 アクセス リストを定義し、 IP または IPv6 アクセス リスト コンフィギュレーション モードを開始します。

ステップ 10	**10 permit tcp any** *any time - range tl*
	例：
	`switch(config-acl)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11`
	`switch(config)#`
		IP または IPv6 アクセス リストの TCP の時間範囲を定義する時間範囲値を設定します。
		（注）
		any キーワードは、IPv6 にのみ使用されます。

ステップ 11	**time-range** *name*
	例：
	`switch(config-acl)# time-range t1`
	`switch(config)#`
		名前を使用して、IP または IPv6 アクセス リストの時間範囲を定義します。

ステップ 12	**F2(config-time-range)#**
	WOLF2(config-time-range)#
	例：
	`switch(config-time-range)# 10 absolute start 20:06:56 8 february 2021`
	`end 20:10:56 8 february 2021`
		構成の時間範囲を定義します。

フローベース トラフィック ステアリングのデフォルトおよび非デフォルト VRF でのルート マップの構成

次のセクションでは、SRTE フローベースのトラフィック ステアリング機能のデフォルトおよび非デフォルト VRF でルート マップを構成する方法を示します。

カラーおよびエンドポイントによって選択されているポリシーへのデフォルト VRF のルート マップの構成

デフォルト VRF のトラフィックを、色とエンドポイントで選択されたポリシーに導くルート マップを構成するには、次の手順を実行します。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルート マップが一致する必要のあるフィールドを指定します。

ステップ 3	**set srte-policy color** *num* **endpoint** *ip address*
	例：
	`switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1`
	`switch(config-route-map)#`
		SRTE ポリシー カラーとポリシーのエンド ポイントを構成します。
		（注）
		IPv4 アドレスのみをエンドポイントにできます。

ステップ 4	**interface** *interface-type/slot/port*
	例：
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		インターフェイス設定モードを開始します。

ステップ 5	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルート マップが適用されます。

名前で選択されたポリシーへのデフォルト VRF のルート マップ構成例

デフォルト VRF のトラフィックを名前で選択されたポリシーに導くルート マップを構成するには、次の手順を実行します。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルート マップが一致する必要のあるフィールドを指定します。

ステップ 3	**set srte-policy name** *policy-name*
	例：
	`switch(config-route-map)# set srte-policy name policy1`
	`switch(config-route-map)#`
		SRTE ポリシー名を構成します。

ステップ 4	**interface** *interface-type/slot/port*
	例：
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		インターフェイス コンフィギュレーション モードを開始します。

ステップ 5	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルート マップが適用されます。

ネクストホップ、カラー、およびエンドポイントで選択されたポリシーへのデフォルト以外の VRF のルート マップ構成

デフォルト以外の VRF のトラフィックを、カラーとエンドポイントで選択されたポリシーに導くルートマップを構成するには、次の手順を実行します。この手順では、正しい MPLS VPN ラベルがトラフィックに適用されるようにネクストホップを指定します。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルート マップが一致する必要のあるフィールドを指定します。

ステップ 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy color** *num* **endpoint** *ip address*
	例：
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121 endpoint 10.0.0.1`
	`switch(config-route-map)#`
		srte-policy（カラーおよびエンドポイント）を介して、構成されたネクストホップにパケットをリダイレクトします。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ構成モードを終了し、グローバル構成モードに戻ります。

ステップ 5	**interface** *interface-type/slot/port*
	例：
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		インターフェイス コンフィギュレーション モードを開始します。

ステップ 6	**vrf member** *vrf-name*
	例：
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		このインターフェイスを VRF に追加します。

ステップ 7	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-if)# ip policy route-map FLOW1`
	`switch(config-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルートマップが適用されます。

ステップ 8	[**no**] **shutdown** 例：	インターフェイスをディセーブルにします。

	コマンドまたはアクション	目的
	`switch(config-if)# no shutdown` `switch(config-if)#`	

デフォルト以外の VRF のルート マップをネクストホップおよびカラー別に選択されたポリシーに構成する

次の手順を実行し、デフォルト VRF のトラフィックを色とエンドポイントで選択されたポリシーに誘導するルートマップを構成しますが、エンドポイントは明示的に構成されていません。 ネクストホップが指定されているため、正しい MPLS VPN ラベルがトラフィックに適用され、正しい SRTE エンドポイントがネクストホップに一致するルートから取得されます。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルート マップが一致する必要のあるフィールドを指定します。

ステップ 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy color** *num*
	例：
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121`
	`switch(config-route-map)#`
		srte-policy（カラー）を介して、構成されたネクストホップにパケットをリダイレクトします。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ構成モードを終了し、グローバル構成モードに戻ります。

ステップ 5	**interface** *interface-type/slot/port*
	例：
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		インターフェイス コンフィギュレーション 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 6	**vrf member** *vrf-name*
	例：
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		このインターフェイスを VRF に追加します。

ステップ 7	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-if)# ip policy route-map FLOW1`
	`switch(config-if-route-map)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルート マップが適用されます。

ステップ 8	[**no**] **shutdown**
	例：
	`switch(config-if-route-map)# no shutdown`
	`switch(config-if-route-map)#`
		インターフェイスをディセーブルにします。

デフォルト以外の VRF のルート マップをネクストホップおよび名前別に選択されたポリシーに構成する

次の手順を実行して、デフォルト以外の VRF のトラフィックを名前別に選択されたポリシーに誘導するルート マップを構成します。ネクストホップは、正しい MPLS VPN ラベルがトラフィックに課されるように指定されます

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルート マップが一致する必要のあるフィールドを指定します。

ステップ 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy** *name* 例：	srte-policy（名前）を介して、構成されたネクストホップにパケットをリダイレクトします。

	コマンドまたはアクション	目的
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy policy1` `switch(config-route-map)#`	

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ構成モードを終了し、グローバル構成モードに戻ります。

ステップ 5	**interface** *interface-type/slot/port*
	例：
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		インターフェイス コンフィギュレーション モードを開始します。

ステップ 6	**vrf member** *vrf-name*
	例：
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		このインターフェイスを VRF に追加します。

ステップ 7	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-if)# ip policy route-map FLOW1`
	`switch(config-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルートマップが適用されます。

ステップ 8	[**no**] **shutdown**
	例：
	`switch(config-if)# no shutdown`
	`switch(config-if)#`
		インターフェイスをディセーブルにします。

カラーとエンドポイントで選択されたポリシーへのデフォルト以外の VRF のルート マップ構成例

デフォルト以外の VRF のトラフィックを、カラーとエンドポイントで選択されたポリシーに導くルートマップを構成するには、次の手順を実行します。この手順では、指定するネクストホップは必要ありません。VPN ラベルは、ローカル スイッチで VRF に割り当てられたラベルを検索することによって取得されます。 これは、すべてのスイッチの VRF の BGP 割り当てインデックス構成を使用して、すべてのスイッチの VRF に同じラベルが割り当てられている場合にのみ構成可能です。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルートマップが一致する必要のあるフィールドを指定します。

ステップ 3	**set srte-policy color** *num* **endpoint** *ip address*
	例：
	`switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1`
	`switch(config-route-map)#`
		SRTE ポリシー カラーとポリシーのエンド ポイントを構成します。
		（注）
		IPv4 アドレスのみをエンドポイントにできます。

ステップ 4	**interface** *interface-type/slot/port*
	例：
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		インターフェイス コンフィギュレーション モードを開始します。

ステップ 5	**vrf member** *vrf-name*
	例：
	`switch(config-route-map-if)# vrf member vrf1`
	`switch(config-route-map-if)#`
		このインターフェイスを VRF に追加します。

ステップ 6	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルート マップが適用されます。

ステップ 7	[**no**] **shutdown**
	例：
	`switch(config-route-map-if)# no shutdown`
	`switch(config-route-map-if)#`
		インターフェイスをディセーブルにします。

	コマンドまたはアクション	目的

ステップ 8	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ構成モードを終了し、グローバル構成モードに戻ります。

ステップ 9	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能を開始します。

ステップ 10	**router bgp as-number**
	例：
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		BGP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。

ステップ 11	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vrf1`
	`switch(config-router-vrf)#`
		BGP プロセスを VRF に関連付けます。

ステップ 12	**allocate-index** *index*
	例：
	`switch(config-router-vrf)#`
	`allocate-index 10`
		VRF にインデックスを割り当てます。これにより、VRF にスタティック MPLS ローカル VPN ラベルを割り当てるように BGP に指示されます。VRF に割り当てられた MPLS VPN ラベルは、指定された値から取得されます。インデックスは、MPLS ラベル値の特別な範囲へのオフセットとして使用されます。 指定されたインデックス値の場合、同じローカル ラベルが常に許可されます。

名前で選択されたポリシーへのデフォルト以外のルート マップ構成例

次の手順を実行して、デフォルト以外の VRF のトラフィックを名前別に選択されたポリシーに誘導するルートマップを構成します。この手順では、指定するネクストホップは必要ありません。VPN ラベルは、ローカル スイッチで VRF に割り当てられたラベルを検索することによって取得されます。 これは、すべてのスイッチの VRF の BGP 割り当てインデックス構成を使用して、すべてのスイッチの VRF に同じラベルが割り当てられている場合にのみ構成可能です。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリングおよび PBR 機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**route-map FLOW1** *seq_num*
	例：
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		ルート マップに FLOW1 という名前を付けます。

ステップ 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	例：
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		フィールドを説明する ACL を追加することにより、ルートマップが一致する必要のあるフィールドを指定します。

ステップ 3	**set srte-policy** *name*
	例：
	`switch(config-route-map)# set srte-policy policy1`
	`switch(config-route-map)#`
		SRTE ポリシー名を構成します。

ステップ 4	**interface** *interface-type/slot/port*
	例：
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		インターフェイス設定モードを開始します。

ステップ 5	**vrf member** *vrf-name*
	例：
	`switch(config-route-map-if)# vrf member vrf1`
	`switch(config-route-map-if)#`
		このインターフェイスを VRF に追加します。

ステップ 6	[**ip** | **ipv6**] **policy route-map FLOW1**
	例：
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		IP または IPv6 ポリシーベース ルーティングをインターフェイスに割り当てます。これにより、インターフェイスに入力するすべてのトラフィックのルート マップが適用されます。

ステップ 7	[**no**] **shutdown**
	例：
	`switch(config-route-map-if)# no shutdown`
	`switch(config-route-map-if)#`
		インターフェイスをディセーブルにします。

ステップ 8	**exit** 例：	ルートマップ構成モードを終了し、グローバル構成モードに戻ります。

	コマンドまたはアクション	目的
	`switch(config-route-map)# exit` `switch(config)#`	

ステップ 9	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能を開始します。

ステップ 10	**router bgp as-number**
	例：
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		BGP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。

ステップ 11	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vrf1`
	`switch(config-router-vrf)#`
		BGP プロセスを VRF に関連付けます。

ステップ 12	**allocate-index** *index*
	例：
	`switch(config-router-vrf)#`
	`allocate-index 10`
		VRF にインデックスを割り当てます。これにより、VRF にスタティック MPLS ローカル VPN ラベルを割り当てるように BGP に指示されます。VRF に割り当てられた MPLS VPN ラベルは、指定された値から取得されます。インデックスは、MPLS ラベル値の特別な範囲へのオフセットとして使用されます。 指定されたインデックス値の場合、同じローカル ラベルが常に許可されます。

SRTE フローベース トラフィック ステアリングの構成例

このセクションには、SRTE フローベースのトラフィック ステアリングを構成するための次の例が含まれています。

ToS/DSCP および時間ベース ACL に基づくフロー選択の構成例

```
switch# configure terminal
switch(config)# ip access-list L4_PORT
switch(config)# 10 permit ip any 5.5.0.0/16
switch(config)# 20 permit tcp any 5.5.0.0/16
switch(config)# ip access-list dscp
switch(config)# 10 permit tcp any 5.5.0.0/16 dscp af11
switch(config)# ip access-list acll
switch(config)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11
switch(config)# ip access-list acl1
switch(config-acl)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11
```

```
switch(config-acl)# time-range t1
start 20:06:56 8 february 2021 end 20:10:56 8 february 2021
```

カラーおよびエンドポイントで選択されたポリシーへのデフォルトVRFのルートマップ構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map-if)# ip policy route-map FLOW1
```

名前別に選択されたポリシーへのデフォルトの VRF でのルートマッピング構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set srte-policy name policy1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map-if)# ip policy route-map FLOW1
```

ネクストホップ、カラー、エンドポイントで選択されたポリシーへのデフォルト以外の VRF のルートマップ構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121 endpoint 10.0.0.1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
```

ネクストホップおよびカラーで選択されたポリシーへのデフォルト以外の VRF のルートマップの構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
```

ネクストホップ名別に選択されたポリシーへのデフォルト以外の VRF でのルートマッピング構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy policy1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
```

デフォルト以外のVRFでのルートマップの構成例を色とエンドポイントで選択したポリシーにマッピングする

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
```

```
switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
switch(config)# feature bgp
switch(config)# router bgp 1.1
switch(config-router)# vrf vrf1
switch(config-router-vrf)# allocate-index 10
```

名前別に選択されたポリシーへのデフォルト以外の VRF でのルートマッピング構成例

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set srte-policy policy1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
switch(config)# feature bgp
switch(config)# router bgp 1.1
switch(config-router)# vrf vrf1
switch(config-router-vrf)# allocate-index 10
```

SRTE のフローベース トラフィック ステアリング構成の確認

SRTE 構成のフローベースのステアリングに関する適切な詳細を表示するには、次のいずれかのタスクを実行します。

表 *1 : SRTE* のフローベース トラフィック ステアリング構成の確認

コマンド	目的
show srte policy	許可されたポリシーのみを表示します。
show srte policy [**all**]	SR-TE で使用可能なすべてのポリシーのリストを表示します。
show srte policy [**detail**]	要求されたすべてのポリシーの詳細ビューを表示します。
show srte policy <*name*>	SR-TE ポリシーを名前でフィルタリングし、SR-TE でその名前で使用できるすべてのポリシーのリストを表示します。 （注） このコマンドには、ポリシー名のオートコンプリート機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。

コマンド	目的
show srte policy color <*color*> **endpoint** <*endpoint*>	カラーとエンドポイントの SR-TE ポリシーを表示します。 （注） このコマンドには、カラーとエンドポイントのオートコンプリート機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。
show route-map [**name**]	ルート マップの情報を表示します。
show forwarding mpls srte module	転送情報ベース - FIB モジュールの SRTE 情報を表示します。

SRTE ポリシーの MPLS OAM モニタリングの構成

SRTE ポリシーの MPLS OAM モニタリングについて

Cisco NX-OS リリース 10.1(2) 以降、MPLS OAM モニタリングにより、1 つ以上の SRTE ポリシーが構成されているスイッチで、SRTE ポリシーのアクティブ パスに障害が発生したかどうかをプロアクティブに検出できます。現在アクティブな優先度の高いパスがすべて失敗した場合、SRTE はその優先度の高いパスがダウンしていると見なし、そのような優先順位があれば、ポリシーで次に高い優先順位をアクティブにします。そうでない場合は、ポリシーをダウンとしてマークします。

この機能の前は、SRTE 優先順位とポリシーの状態は、優先順位内のパスの最初のホップ（最初の MPLS ラベル）の状態によってのみ決定されていました。ラベルがプログラムされている場合、パスは稼働していると見なされ、ラベルがないか無効な場合、パスは停止していると見なされます。

MPLS OAM モニタリングは、MPLS LSPV Nil-FEC ping 要求を SRTE パスに沿って継続的に送信することにより、この検証を強化します。各 ping 要求には、SRTE ポリシーに従うトラフィックに課されるものと同じラベル スタックが含まれているため、ping は同じパスをたどります。ping は、各 ping 間の構成可能な間隔で送信され、パスの最終ノードからの ping への応答は間隔内で期待されます。最終ノードから障害応答が返ってきた場合、または間隔内に応答がなかった場合は、失敗間隔としてカウントされます。構成可能な数の失敗間隔が連続して発生すると、パスはダウンしていると見なされます。優先順位のすべてのパスがダウンしている場合、優先順位はダウンしていると見なされます。

モニタされたパス

CLI がプロアクティブなモニタリングを使用してパスをモニタできる場合にのみ、OAM を使用してパスがモニタされます。ポリシーに関連付けられているパスのみがモニタされます。た

とえば、セグメントリストが作成されポリシーに関連付けられていない場合、それはモニタされません。また、同じパスが複数のポリシーで使用されている場合、そのパスに対して作成されるモニタリングセッションは1つだけです。これは、パスがポリシーの基本設定に関連付けられたセグメントリストであるか、ヘッドエンドでパス補完を使用して計算されたものであるかに関係なく適用されます。

デフォルトでは、イメージがOAMモニタリング サポートのないバージョンからモニタリング サポートのあるバージョンにアップグレードされた場合、ポリシーのモニタリング方式は従来のファーストホップ方式になります。

MPLS OAM モニタリングは、すべての SRTE ポリシーに対してグローバルに有効にすることができます。グローバルに有効になっている場合、ポリシーごとに選択的に無効にすることができます。グローバルに有効化されていない場合は、個々のポリシーに対して選択的に有効化できます。

インデックス制限

index-limit X CLI は、パス全体ではなく、パスの最初のサブセットのみを ping するために使用されます。指定された index-limit 以下のセグメントリスト内のインデックスのみが、モニタするパスの一部です。たとえば、セグメント リストが次のようになっているとします。

```
index 100 mpls label 16001
index 200 mpls label 16002
index 300 mpls label 16003
```

次に、index-limit が指定されていない場合、ping されるパスは 16001、16002、16003 になります。index-limit が 250 の場合、ping されるパスは 16001、16002 になります。index-limit が 200 の場合、ping されるパスも 16001、16002 になります。

SRTE ポリシーの MPLS OAM モニタリングに関する注意事項と制限事項

SRTE ポリシーの MPLS OAM モニタリングには、次のガイドラインと制限事項があります。

- Cisco NX-OS リリース 10.1(2) 以降、MPLS OAM モニタリング（継続的かつ予防的なパス）が導入され、Cisco Nexus 9300 EX、9300-FX、9300-FX2、および 9300-GX プラットフォーム スイッチでサポートされています。
- SRTE ポリシーが構成されているヘッドエンド ノードでは、SRTE と MPLS OAM の両方を、それぞれ feature mpls segment-routing traffic-engineering および feature mpls oam の一部として個別に有効にする必要があります。そうでない場合、ユーザーは OAM を使用して SRTE ポリシーのモニタリングを構成できません。さらに、SR ファブリックの残りのノードでは、MPLS OAM モニタリングによって送信された ping に応答するために、feature mpls oam を使用して MPLS OAM を有効にする必要があります。
- SRTE は、モニタリング セッションの最大数を 1000 に制限します。
- ping の最小間隔は 1000 ミリ秒です。

- SRTE OAM モニタリング ポリシーがデバイスで実行されている場合、`feature mpls oam` を無効にすることはできません。すべての SRTE OAM モニタリング ポリシーが無効になっている場合にのみ、デバイスから `feature mpls oam` を無効にできます。それ以外の場合、次のエラー メッセージが表示されます。

  「SRTE MPLS 活性検出は、すべてのポリシーに対して有効になっているか、少なくとも 1 つのポリシーに対して有効になっているか、またはオンデマンド カラーに対して有効になっています。MPLS OAM を無効にする前に、活性検出が完全に無効になっていることを確認してください。」

- Cisco NX-OS リリース 10.1(2) では、SRTE OAM モニタリングは、スタティック ポリシーと、明示パスが構成されているオンデマンド カラーに対してサポートされています。
- OAM セッションは、PCEP を使用してダイナミック オプションで構成されたパスでは実行されません。

MPLS OAM モニタリングの構成

このセクションでは、ポリシーのプロアクティブなパス モニタリングを有効にするために必要な CLI について説明します。

- **グローバル設定**

  この構成により、構成されたすべてのポリシーの OAM パス モニタリングが有効になります。

- **ポリシー固有の構成**

  この構成により、特定のポリシーの OAM パス モニタリングが有効になります。

グローバル設定

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリング機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**segment-routing** 例：	セグメント ルーティング構成モードを開始します。

	コマンドまたはアクション	目的
	`switch(config)#segment-routing` `switch(config-sr)#`	

ステップ 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィック エンジニアリング モードに入ります。

ステップ 4	[**liveness-detection**]
	例：
	`switch(config-sr-te)#`
	`liveness-detection`
	`switch(config-sr-te-livedet)#`
		活性検出構成モードを開始します。

ステップ 5	**interval** *num*
	例：
	`switch(config-sr-te-livedet)# interval`
	`6000`
	`switch(config-sr-te-livedet)#`
		間隔はミリ秒です。デフォルトは3000 ms です。

ステップ 6	**multiplier** *num*
	例：
	`switch(config-sr-te-livedet)#`
	`multiplier 5`
	`switch(config-sr-te-livedet)#`
		乗数は、乗数は、ダウンと見なされるためにアップしているパスの失敗する必要がある連続間隔数と、アップとみなされるためにダウンしているパスの連続間隔数を設定します。デフォルトは 3 です。

ステップ 7	**mpls**
	例：
	`switch(config-sr-te-livedet)# mpls`
	`switch(config-sr-te-livedet-mpls)#`
		mpl を介したセグメント ルーティングを有効にします。

ステップ 8	[**no**]**oam**
	例：
	`switch(config-sr-te-livedet-mpls)#`
	`oam`
	`switch(config-sr-te-livedet-mpls)#`
		すべてのSRTEポリシーに対してMPLS OAMモニタリングをグローバルに有効にします。
		このコマンドの no 形式で、OAM モニタリングを無効にします。

ステップ 9	**segment-list name** *sidlist-name*
	例：
	`switch(config-sr-te)# segment-list name blue`
	`index 10 mpls label 16004`
	`index 10 mpls label 16005`
		明示 SID リストを作成します。
		（注）
		このコマンドは、sidlist-nameの自動入力機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。

	コマンドまたはアクション	目的

ステップ 10	**policy** *policy name*
	例：
	`switch(config-sr-te)# policy 1`
	`switch(config-sr-te-pol)`
		ポリシーを設定します。

ステップ 11	**color** *numberIP-end-point*
	例：
	`switch(config-sr-te-pol)# color 1 endpoint 5.5.5.5`
	`switch(config-sr-te-pol)`
		ポリシーのカラーとエンドポイントを設定します。

ステップ 12	**candidate-paths**
	例：
	`switch(config-sr-te-pol)# candidate-paths`
	`switch(config-expcndpaths)#`
		ポリシーの候補パスを指定します。

ステップ 13	**preference** *preference-number*
	例：
	`switch(config-expcndpaths)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 14	*sidlist-name*explicit segment-list
	例：
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		明示リストを指定します。
		（注）
		このコマンドは、sidlist-name の自動入力機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。

ステップ 15	**on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 211`
	`switch(config-sr-te-color)#`
		オンデマンド色テンプレートモードを開始し、特定の色のオンデマンド色を構成します。

ステップ 16	**candidate-paths**
	例：
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		ポリシーの候補パスを指定します。

ステップ 17	**preference** *preference-number*
	例：
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

	コマンドまたはアクション	目的

ステップ 18	*sidlist-name*explicit segment-list
	例：
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		明示リストを指定します。
		（注）
		このコマンドは、sidlist-name の自動入力機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。

ポリシー固有の構成

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリング機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**segment-routing**
	例：
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		セグメントルーティング構成モードを開始します。

ステップ 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィックエンジニアリングモードに入ります。

ステップ 4	[**liveness-detection**]
	例：
	`switch(config-sr-te)# liveness-detection`
	`switch(config-sr-te-livedet)#`
		活性検出構成モードを開始します。

ステップ 5	**interval** *num*
	例：
	`switch(config-sr-te-livedet)# interval 6000`
	`switch(config-sr-te-livedet)#`
		間隔はミリ秒です。デフォルトは3000 ms です。

	コマンドまたはアクション	目的

ステップ 6	**multiplier** *num*
	例：
	`switch(config-sr-te-livedet)# multiplier 5`
	`switch(config-sr-te-livedet)#`
		乗数は、乗数は、ダウンと見なされるためにアップしているパスの失敗する必要がある連続間隔数と、アップとみなされるためにダウンしているパスの連続間隔数を設定します。デフォルトは 3 です。

ステップ 7	**segment-list name** *sidlist-name*
	例：
	`switch(config-sr-te)# segment-list name blue`
	`index 10 mpls label 16004`
	`index 10 mpls label 16005`
		明示 SID リストを作成します。
		（注）
		このコマンドは、sidlist-name の自動入力機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。

ステップ 8	**policy** *policy name*
	例：
	`switch(config-sr-te)# policy 1`
	`switch(config-sr-te-pol)`
		ポリシーを設定します。

ステップ 9	**color** *numberIP-end-point*
	例：
	`switch(config-sr-te-pol)# color 1 endpoint 5.5.5.5`
	`switch(config-sr-te-pol)`
		ポリシーのカラーとエンドポイントを設定します。

ステップ 10	**candidate-paths**
	例：
	`switch(config-sr-te-pol)# candidate-paths`
	`switch(config-expcndpaths)#`
		ポリシーの候補パスを指定します。

ステップ 11	**preference** *preference-number*
	例：
	`switch(config-expcndpaths)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

ステップ 12	*sidlist-name*explicit segment-list
	例：
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		明示リストを指定します。
		（注）
		このコマンドは、sidlist-name の自動入力機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。

	コマンドまたはアクション	目的

ステップ 13	[**liveness-detection**]
	例：
	`switch(config-sr-te)# liveness-detection`
	`switch(config-sr-te-livedet)#`
		活性検出構成モードを開始します。

ステップ 14	[**no**]**index-limit** *num*
	例：
	`switch(config-sr-te-livedet)# index-limit 20`
	`switch(config-sr-te-livedet)#`
		ユーザーが指定した数以下のインデックスを持つ SID のみをモニタします。

ステップ 15	[**no**]**shutdown**
	例：
	`switch(config-sr-te-livedet)# shutdown`
	`switch(config-sr-te-livedet)#`
		活性検出を無効にします。これは、関連するすべての構成を完全に削除せずに、活性検出を一時的に無効にする場合に便利です。
		このコマンドの no 形式で、OAM モニタリングを無効にします。

ステップ 16	**mpls**
	例：
	`switch(config-sr-te-livedet)# mpls`
	`switch(config-sr-te-livedet-mpls)#`
		mpl を介したセグメント ルーティングを有効にします。

ステップ 17	[**no**]**oam**
	例：
	`switch(config-sr-te-livedet-mpls)# oam`
	`switch(config-sr-te-livedet-mpls)#`
		すべてのSRTEポリシーに対してMPLS OAMモニタリングをグローバルに有効にします。
		このコマンドの no 形式で、OAM モニタリングを無効にします。

ステップ 18	**on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 211`
	`switch(config-sr-te-color)#`
		オンデマンド色テンプレートモードを開始し、特定の色のオンデマンド色を構成します。

ステップ 19	**candidate-paths**
	例：
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		ポリシーの候補パスを指定します。

ステップ 20	**preference** *preference-number*
	例：
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		候補パスの優先順位を指定します。

	コマンドまたはアクション	目的

ステップ 21	*sidlist-name*explicit segment-list
	例：
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		明示リストを指定します。
		（注）
		このコマンドは、sidlist-name の自動入力機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。

ステップ 22	[**liveness-detection**]
	例：
	`switch(config-sr-te-color)# liveness-detection`
	`switch(config-sr-te-color-livedet)#`
		活性検出構成モードを開始します。

ステップ 23	[**no**]**index-limit** *num*
	例：
	`switch(config-sr-te-color-livedet)# index-limit 20`
	`switch(config-sr-te-color-livedet)#`
		ユーザーが指定した数以下のインデックスを持つ SID のみをモニタします。

ステップ 24	[**no**]**shutdown**
	例：
	`switch(config-sr-te-color-livedet)# shutdown`
	`switch(config-sr-te-color-livedet)#`
		活性検出を無効にします。これは、関連するすべての構成を完全に削除せずに、活性検出を一時的に無効にする場合に便利です。
		このコマンドの no 形式で、OAM モニタリングを無効にします。

ステップ 25	**mpls**
	例：
	`switch(config-sr-te-color-livedet)# mpls`
	`switch(config-sr-te-color-livedet-mpls)#`
		mpl を介したセグメント ルーティングを有効にします。

ステップ 26	[**no**]**oam**
	例：
	`switch(config-sr-te-color-livedet-mpls)# oam`
	`switch(config-sr-te-color-livedet-mpls)#`
		すべての SRTE ポリシーに対して MPLS OAM モニタリングをグローバルに有効にします。
		このコマンドの no 形式で、OAM モニタリングを無効にします。

MPLS OAM モニタリングの構成の確認

MPLS OAM モニタリングの構成情報を表示するには、次のタスクのいずれかを実行します。

表 *2 : MPLS OAM* モニタリングの構成の確認

コマンド	目的
show srte policy	許可されたポリシーのみを表示します。
show srte policy [**all**]	SR-TE で使用可能なすべてのポリシーのリストを表示します。
show srte policy [**detail**]	要求されたすべてのポリシーの詳細ビューを表示します。
show srte policy <*name*>	SR-TE ポリシーを名前でフィルタリングし、SR-TE でその名前で使用できるすべてのポリシーのリストを表示します。 （注） このコマンドには、ポリシー名のオートコンプリート機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。
show srte policy color <*color*> **endpoint** <*endpoint*>	カラーとエンドポイントの SR-TE ポリシーを表示します。 （注） このコマンドには、カラーとエンドポイントのオートコンプリート機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。

コマンド	目的
show srte policy proactive-policy-monitoring	promon データベースに存在するすべてのアクティブなプロアクティブ ポリシー モニタリング セッションのリストを表示します。 （注） このコマンドの最後に疑問符オプションを使用して、次のオプションのいずれかを指定するか、Enter キーを押してすべてのセッションを表示できます。 • brief：セッションに関する簡単な情報を表示します • color：ポリシーのカラーに関連する promon セッションを示します • name：ポリシー名に関連する Promon セッションを表示します • セッション ID：セッション ID の Promon セッションを表示します
show srte policy proactive-policy-monitoring [**brief**]	セッション ID のリストとプロアクティブ ポリシー モニタリング セッションの状態のみを表示します。
show srte policy proactive-policy-monitoring [**session** <*session-id*>]	セッション ID を使用してフィルタリングし、そのセッションに関する情報を詳細に表示します。 （注） このコマンドには、セッション ID の自動入力機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。
show srte policy proactive-policy-monitoring color <*color*> **endpoint**<*endpoint*>	カラーとエンドポイントを使用してフィルタリングし、プロアクティブなポリシー モニタリング セッションを表示します。 （注） このコマンドには、カラーとエンドポイントのオートコンプリート機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。

MPLS OAM モニタリングの構成例

次に、MPLS OAM モニタリングの構成例を示します。

- ユーザー指定の乗数と間隔によるグローバル有効化の構成例：

```
segment-routing
  traffic-engineering
    liveness-detection
        interval 6000
        multiplier 5
      mpls
        oam
    segment-list name blue
      index 10 mpls label 16004
      index 20 mpls label 16005
    segment-list name green
      index 10 mpls label 16003
      index 20 mpls label 16006
    segment-list name red
      index 10 mpls label 16002
      index 20 mpls label 16004
      index 30 mpls label 16005
    policy customer-1
      color 1 endpoint 5.5.5.5
      candidate-paths
        preference 100
          explicit segment-list red
    on-demand color 211
      candidate-paths
        preference 100
          explicit segment-list green
```

- ユーザー指定の乗数、間隔、インデックス制限、およびシャットダウン オプションを使用したポリシー有効化の構成例：

```
segment-routing
  traffic-engineering
    liveness-detection
        interval 6000
        multiplier 5
    segment-list name blue
      index 10 mpls label 16004
      index 20 mpls label 16005
    segment-list name green
      index 10 mpls label 16003
      index 20 mpls label 16006
    segment-list name red
      index 10 mpls label 16002
      index 20 mpls label 16004
      index 30 mpls label 16005
    policy customer-1
      color 1 endpoint 5.5.5.5
      candidate-paths
        preference 100
          explicit segment-list red
      liveness-detection
        index-limit 20
        shutdown
        mpls
          oam
    on-demand color 211
      candidate-paths
```

```
  preference 100
    explicit segment-list green
liveness-detection
    index-limit 20
    shutdown
    mpls
      oam
```

セグメントルーティングでの出力ピアエンジニアリングの設定

BGP プレフィックス SID

セグメント ルーティングをサポートするためには、BGP が BGP プレフィックスのセグメント ID（SID）をアドバタイズできなければなりません。BGP プレフィックス SID は常にセグメント ルーティング BGP ドメイン内でグローバルであり、命令を識別し、BGP によって計算された ECMP 対応のベストパスを介して、パケットを関連するプレフィックスに転送します。BGP プレフィックス SID は、BGP プレフィックス セグメントを識別します。

隣接 SID

隣接関係セグメント識別子（SID）は、特定のインターフェイスとそのインターフェイスからの次のホップを指す、ローカル ラベルです。隣接関係 SID を有効にするために必要な特定の設定はありません。アドレス ファミリの BGP を介してセグメント ルーティングが有効になると、BGP が実行されるすべてのインターフェイスに対して、アドレス ファミリがそのインターフェイスのすべてのネイバーに対して隣接 SID を自動的に割り当てます。

セグメント ルーティングのための高可用性

インサービス ソフトウェア アップグレード (ISSU) は、BGP グレースフル リスタートで最低限サポートされます。すべての状態（セグメント ルーティング状態を含む）は、BGP ルータのピアから再学習する必要があります。グレースフル リスタート期間中、以前に学習したルートとラベルの状態は保持されます。

セグメント ルーティングを使用した BGP 出力ピア エンジニアリングの概要

Cisco Nexus 9000 シリーズ スイッチは、多くの場合、大規模データセンター（MSDC）に導入されます。このような環境では、セグメント ルーティング（SR）で BGP 出力ピア エンジニアリング（EPE）をサポートすることが要件となります。

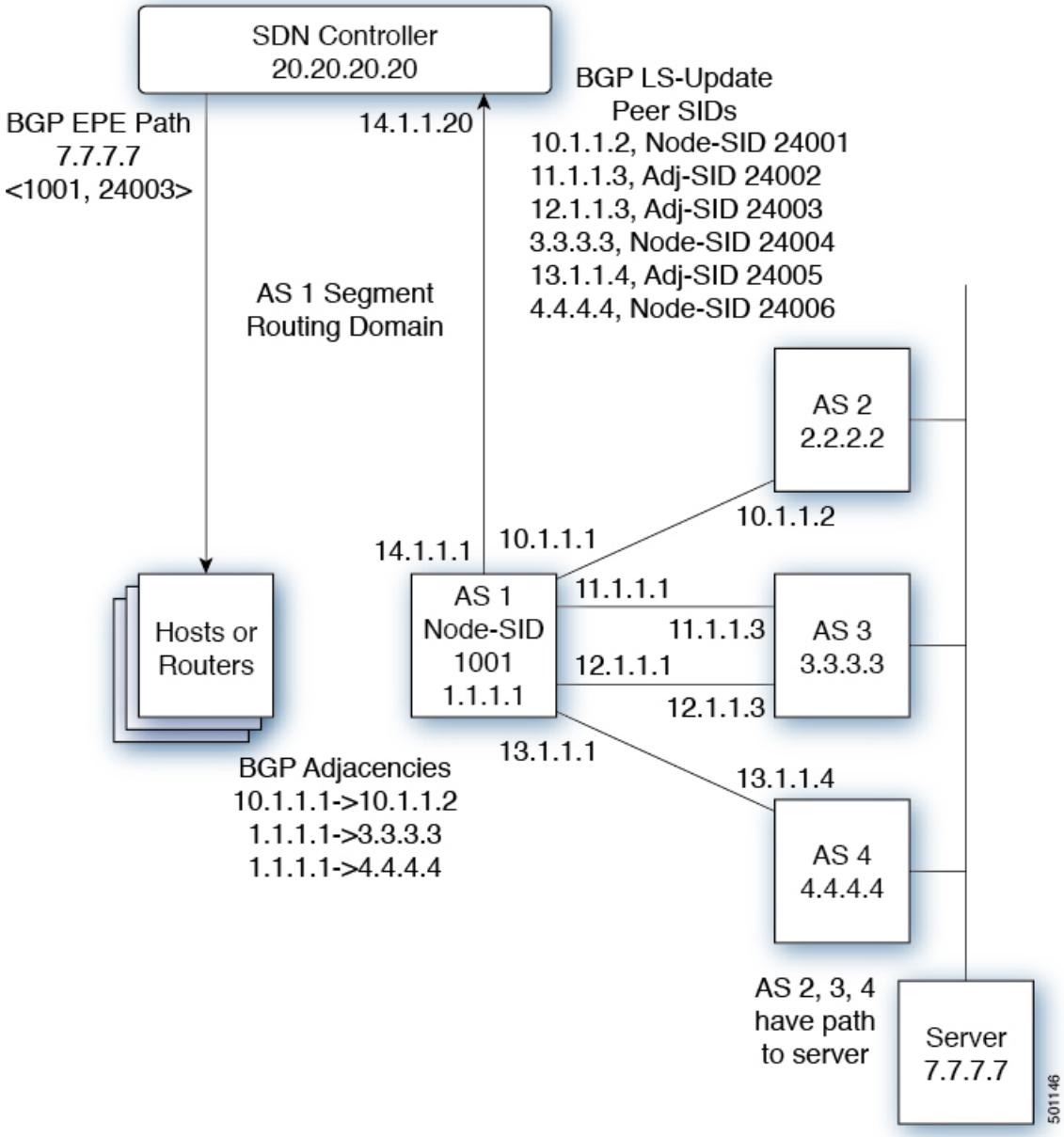
セグメント ルーティング（SR）はソース ルーティングを利用します。ノードは、制御された一連の命令（セグメント）によってパケットを操作するために、パケットの前に SR ヘッダーを付加します。セグメントは、トポロジまたはサービスベースの命令を表すことができます。SR では、SR ドメインの入力ノードでのみフローごとの状態を維持しながら、トポロジパスまたはサービス チェーンを介してフローを操作できます。この機能の場合、セグメント ルーティング アーキテクチャは、MPLS データ プレーンに直接適用されます。

セグメント ルーティングをサポートするためには、BGP が BGP プレフィックスのセグメント ID（SID）をアドバタイズできなければなりません。BGP プレフィックスは常に SR または BGP ドメイン内でグローバルであり、命令を識別し、BGP によって計算された ECMP 対応のベストパスを介して、パケットを関連するプレフィックスに転送します。BGP プレフィックスは、BGP プレフィックス セグメントの識別子です。

SR ベースの出力ピア エンジニアリング（EPE）ソリューションにより、集中型（SDN）コントローラは、ドメイン内の入力境界ルータまたはホストで任意の出力ピア ポリシーをプログラムできます。

次の例では、3 つのルータすべてが iBGP を実行し、NRLI を相互にアドバタイズします。また、ルータはループバックをネクストホップとしてアドバタイズし、再帰的に解決します。これにより、図に示すように、ルータ間に ECMP が提供されます。

図 *3* : 出力ピア エンジニアリングの例



SDN コントローラは、そのピアおよび隣接のそれぞれについて、出力ルータ 1.1.1.1 からのセグメント ID を受信します。次に、出口ポイントをコントローラのルーティング ドメイン内の他のルータおよびホストにインテリジェントにアドバタイズできます。図に示すように、BGP ネットワーク層到達可能性情報（NLRI）には、ルータ 1.1.1.1 へのノード SID と、7.7.7.7 へのトラフィックがリンク 12.1.1.1->12.1.1.3 を介して出力されることを示すピア隣接 SID 24003 の両方が含まれています。 。

BGP 出力ピア エンジニアリングのガイドラインと制限事項

BGP 出力ピア エンジニアリングには、次のガイドラインと制限事項があります。

- BGP 出力ピア エンジニアリングは、IPv4 BGP ピアでのみサポートされています。IPv6 BGP ピアはサポートされていません。
- BGP 出力ピア エンジニアリングは、デフォルトの VPN ルーティングおよび転送（VRF）インスタンスでのみサポートされます。
- 出力ピア エンジニアリング（EPE）ピア セットには、任意の数の EPG ピアを追加できます。ただし、インストールされている復元力のある CE ごとの FEC は 32 ピアに制限されています。
- 特定の BGP ネイバーは、単一のピア セットのメンバーにしかなれません。ピアセットが構成されています。複数のピアセットはサポートされていません。オプションの**ピアセット**名を指定して、ネイバーをピアセットに追加できます。対応する RPC FEC は、ピアセット内のすべてのピア間でトラフィックを負荷分散します。ピアセット名は、最長 63 文字の文字列です（64 NULL で終了）。この長さは、NX-OS ポリシー名の長さと一致します。ピアは、単一のピアセットのメンバーにしかなれません。
- 特定のピアの隣接関係は、異なるピアセットに個別に割り当てることはできません。
- Cisco NX-OS リリース 9.3(3) 以降、BGP 出力ピア エンジニアリングは Cisco Nexus 9300-GX プラットフォーム スイッチでサポートされます。

BGP を使用したネイバー出力ピア エンジニアリングの設定

RFC 7752 および draft-ietf-idr-bgpls-segment-routing-epe の導入により、出力園児に名リングを設定できます。この機能は、外部 BGP ネイバーに対してのみ有効であり、デフォルトでは設定されていません。出力エンジニアリングでは、RFC 7752 エンコーディングを使用します。

始める前に

- BGPを有効にする必要があります。
- リリース 7.0(3)I3(1) またはリリース 7.0(3)I4(1) からアップグレードした後、Cisco Nexus 9000 シリーズ スイッチで出力ピア エンジニアリング（EPE）を設定する前に、次のコマンドを使用して、TCAM リージョンを設定します。
  1. switch# **hardware access-list tcam region vpc-convergence 0**
  2. switch# **hardware access-list tcam region racl 0**
  3. switch# **hardware access-list tcam region mpls 256 double-wide**
- 設定を保存して、スイッチをリロードします。

詳細については、*Cisco Nexus 9000 Series NX-OS Security Configuration Guide* の「Using Templates to Configure ACL TCAM Region Sizes」および「Configuring ACL TCAM Region Sizes」のセクションを参照してください。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**router bgp** <*bgp autonomous number*>	自律ルータ BGP 番号を指定します。
ステップ 3	**neighbor** <*IP address*>	ネイバーの IP アドレスを設定します。

ステップ 4	[**no**|**default**] **egress-engineering** [**peer-set** *peer-set-name*]
	例：
	`switch(config)# router bgp 1`
	`switch(config-router)# neighbor 4.4.4.4`
	`switch(config-router)#`
	`egress-engineering peer-set NewPeer`
		ピアノード SID がネイバーに割り当てられ、BGP リンク状態（BGP-LS）アドレス ファミリ リンク NLRI のインスタンスでアドバタイズされるかどうかを指定します。ネイバーがマルチホップ ネイバーである場合、BGP-LS リンク NLRI インスタンスもネイバーへの等コストマルチパス（ECMP）パスごとにアドバタイズされます。これには、一意の Peer-Adj-SID が含まれます。
		オプションで、ネイバーをピア セットに追加できます。ピアセット SID は、ピアノード SID と同じインスタンスの BGP-LS リンク NLRI でもアドバタイズされます。BGP リンクステート NLRI は、リンクステート アドレス ファミリが設定されているすべてのネイバーにアドバタイズされます。
		EPE の詳細については、RFC 7752 および draft-ietf-idr-bgpls-segment-routing-epe-05 を参照してください。

出力ピア エンジニアリングの設定例

BGP スピーカー 1.1.1.1 の出力ピア エンジニアリングのサンプル設定を参照してください。ネイバー 20.20.20.20 は SDN コントローラであることに注意してください。

```
 hostname epe-as-1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface Ethernet1/2
  no switchport
  ip address 11.1.1.1/24
  no shutdown

interface Ethernet1/3
  no switchport
  ip address 12.1.1.1/24
  no shutdown

interface Ethernet1/4
  no switchport
  ip address 13.1.1.1/24
  no shutdown

interface Ethernet1/5
  no switchport
  ip address 14.1.1.1/24
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management


interface loopback1
  ip address 1.1.1.1/32
line console

line vty
ip route 2.2.2.2/32 10.1.1.2
ip route 3.3.3.3/32 11.1.1.3
ip route 3.3.3.3/32  12.1.1.3
ip route 4.4.4.4/32  13.1.1.4
ip route 20.20.20.20/32 14.1.1.20

router bgp 1
  address-family ipv4 unicast
  address-family link-state
 neighbor 10.1.1.2
    remote-as 2
    address-family ipv4
```

```
    egress-engineering
 neighbor 3.3.3.3
   remote-as 3
   address-family ipv4
   update-source loopback1
   ebgp-multihop 2
   egress-engineering
 neighbor 4.4.4.4
   remote-as 4
   address-family ipv4
   update-source loopback1
   ebgp-multihop 2
   egress-engineering
neighbor 20.20.20.20
   remote-as 1
   address-family link-state
   update-source loopback1
   ebgp-multihop 2
neighbor 124.11.50.5
    bfs
    remote-as 6
    update-source port-channel50.11
    egress-engineering peer-set pset2 <<<<<<<
    address-family ipv4 unicast
neighbor 124.11.101.2
    bfd
    remote-as 6
    update-source Vlan2401
    egress-engineering
    address-family ipv4 unicast
```

次に、**show bgp internal epe** コマンドの出力例を示します。

```
switch# show bgp internal epe
BGP Egress Peer Engineering (EPE) Information:
Link-State Server: Inactive
Link-State Client: Active
Configured EPE Peers: 26
Active EPE Peers: 3
EPE SID State:
RPC SID Peer or Set Assigned
ID Type Set Name ID Label Adj-Info, iod
1 Node 124.1.50.5 1 1600
2 Set pset1 2 1601
3 Node 6.6.6.6 3 1602
4 Node 124.11.50.5 4 1603
5 Set pset2 5 1604
6 Adj 6.6.6.6 6 1605 124.11.50.4->124.11.50.5/0x1600b031, 80
7 Adj 6.6.6.6 7 1606 124.1.50.4->124.1.50.5/0x16000031, 78
EPE Peer-Sets:
IPv4 Peer-Set: pset1, RPC-Set 2, Count 7, SID 1601
Peers: 124.11.116.2 124.11.111.2 124.11.106.2 124.11.101.2
124.11.49.5 124.1.50.5 124.1.49.5
IPv4 Peer-Set: pset2, RPC-Set 5, Count 5, SID 1604
Peers: 124.11.117.2 124.11.112.2 124.11.107.2 124.11.102.2
124.11.50.5
IPv4 Peer-Set: pset3, RPC-Set 0, Count 4, SID unspecified
Peers: 124.11.118.2 124.11.113.2 124.11.108.2 124.11.103.2
IPv4 Peer-Set: pset4, RPC-Set 0, Count 4, SID unspecified
Peers: 124.11.119.2 124.11.114.2 124.11.109.2 124.11.104.2
IPv4 Peer-Set: pset5, RPC-Set 0, Count 4, SID unspecified
Peers: 124.11.120.2 124.11.115.2 124.11.110.2 124.11.105.2
switch#
```

BGP リンク ステート アドレス ファミリの設定

対応する SID をアドバタイズするコントローラを持つネイバー セッションに対し、 BGP リンク ステート アドレス ファミリを設定することができます。この機能は、グローバル コンフィギュレーション モードおよびネイバー アドレス ファミリ コンフィギュレーション モードで設定できます。

始める前に

BGPを有効にする必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**router bgp** <*bgp autonomous number*>	自律ルータ BGP 番号を指定します。

ステップ 3	[**no**] **address-family link-state**
	例：
	`switch(config)# router bgp 64497`
	`switch (config-router af)#`
	`address-family link-state`
		アドレスファミリ インターフェイス コンフィギュレーション モードを開始します。
		（注）
		このコマンドは、ネイバー アドレス ファミリ コンフィギュレーション モードでも設定できます。

ステップ 4	**neighbor** <*IP address*>	ネイバーの IP アドレスを設定します。

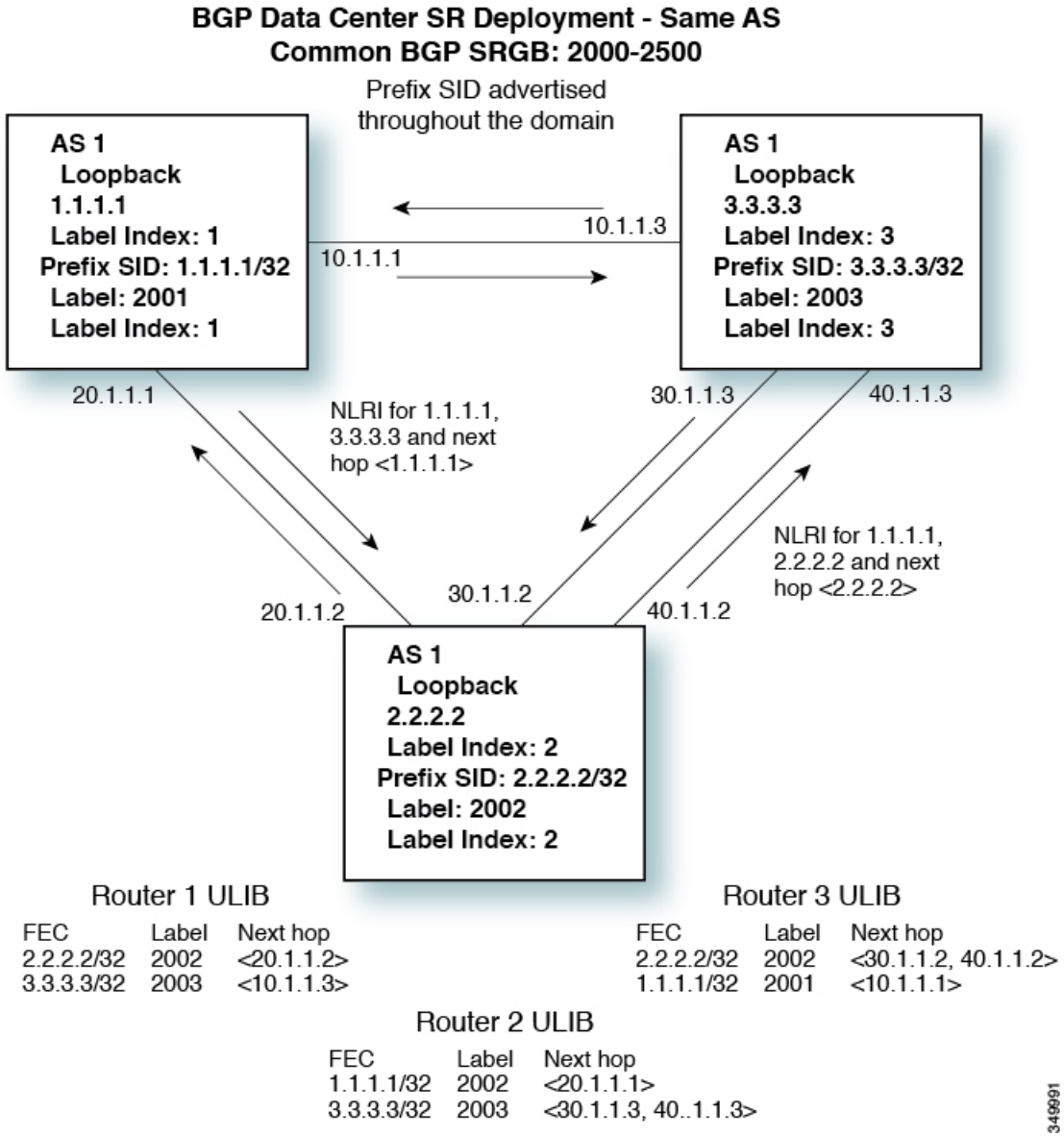
ステップ 5	[**no**] **address-family link-state**
	例：
	`switch(config)#router bgp 1`
	`switch(config-router)#address-family link-state`
	`switch(config-router)#neighbor 20.20.20.20`
	`switch(config-router)#address-family link-state`
		アドレスファミリ インターフェイス コンフィギュレーション モードを開始します。
		（注）
		このコマンドは、ネイバー アドレス ファミリ コンフィギュレーション モードでも設定できます。

BGP プレフィックス SID の展開例

以下の簡単な例では、3 つのルーターすべてが iBGP を実行し、ネットワーク層到達可能性情報（NRLI）を互いにアドバタイズしています。また、ルーターは、ルーター 2.2.2.2 と 3.3.3.3

の間に ECMP を提供するネクスト ホップとして、ループバック インターフェイスをアドバタイズしています。

図 *4 : BGP* プレフィックス *SID* の簡単な例



セグメント ルーティング MPLS 上のレイヤ 2 EVPNの設定

レイヤ 2 EVPN について

イーサネット VPN（EVPN）は、MPLS ネットワークを介してイーサネット マルチポイント サービスを提供する次世代のソリューションです。EVPN は、コアでコントロールプレーン ベースの MAC ラーニングを可能にする既存の仮想プライベート LAN サービス（VPLS）とは対照的に動作します。EVPN では、EVPN インスタンスに参加している PE が MP-BGP プロトコルを使用してコントロールプレーン内でカスタマー MAC ルートを学習します。コントロールプレーン MAC 学習には数多くの利点があり、フローごとのロードバランシングによるマルチホーミングのサポートなどにより、VPLS の弱点に EVPN で対処できるようにします。

EVPN コントロール プレーンでは、データセンター ネットワークにおいて、次のものを提供します。

- データセンター ネットワークの物理トポロジに制限されない、柔軟なワークロード配置。そのため、データセンター ファブリック内の任意の場所に仮想マシン（VM）を配置できます。
- データセンター内部およびデータセンター間における最適なサーバー間 East-West トラフィック。サーバ/仮想マシン間の East-West トラフィックは、ファースト ホップ ルータでのほぼ特定されたルーティングで達成されます。ファースト ホップ ルーティングはアクセス レイヤで行われます。ホスト ルートの交換は、サーバまたはホストへの流入と送出に関するルーティングがほぼ特定されるようにする必要があります。VM モビリティは、新しい MAC アドレスまたは IP アドレスがローカル スイッチに直接接続されている場合に、新しいエンドポイント接続を検出することでサポートされます。ローカル スイッチは、新しい MAC または IP アドレスを検出すると、ネットワークの残りの部分に新しいロケーションを通知します。
- レイヤ 2 およびレイヤ 3 トラフィックのセグメンテーション。トラフィック セグメンテーションは MPLS カプセル化を使用して実現され、ラベル（BD ごとのラベルおよび VRF ごとのラベル）はセグメント識別子として機能します。

セグメント ルーティング MPLS 上のレイヤ 2 EVPN の注意事項と制限事項

セグメント ルーティング MPLS 上のレイヤ 2 EVPN には、次の注意事項と制限事項があります。

- セグメント ルーティング レイヤ 2 EVPN フラッディングは、入力レプリケーション メカニズムに基づいています。MPLS コアはマルチキャストをサポートしていません。
- ARP 抑制はサポートされていません。
- vPC での整合性チェックはサポートされていません。

- 同じレイヤ 2 EVI とレイヤ 3 EVI を一緒に設定することはできません。
- Cisco NX-OSリリース 9.3(1) 以降、レイヤ 2 EVPN は Cisco Nexus 9300-FX2 プラットフォーム スイッチでサポートされます。
- Cisco NX-OSリリース 9.3(5) 以降、セグメント ルーティング MPLS 上のレイヤ 2 EVPN は、Cisco Nexus 9300-GX および Cisco Nexus 9300-FX3 プラットフォーム スイッチでサポートされます。

セグメント ルーティング MPLS 上のレイヤ 2 EVPNの設定

始める前に

次の手順を実行します。

- **install feature-set mpls** コマンドと **feature-set mpls** コマンドを使用して、MPLS 機能セットをインストールして有効にする必要があります。
- MPLS セグメント ルーティング機能を有効にする必要があります。
- **nv overlay** コマンドを使用して、nv オーバーレイ機能を有効にする必要があります。
- **nv overlay evpn** コマンドを使用して EVPN コントロール プレーンを有効にする必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**feature bgp** 例： `switch(config)#feature bgp`	BGP 機能と構成を有効にします。
ステップ 3	**install feature-set mpls** 例： `switch(config)#install feature-set mpls`	MPLS 構成コマンドを有効にします。
ステップ 4	**feature-set mpls** 例： `switch(config)#install feature-set mpls`	MPLS 構成コマンドを有効にします。

	コマンドまたはアクション	目的
ステップ 5	**feature mpls segment-routing** 例： `switch(config)#feature mpls segment-routing`	セグメントルーティング構成コマンドを有効にします。
ステップ 6	**feature mpls evpn** 例： `switch(config)#feature mpls evpn`	EVPN over MPLS 構成コマンドを有効にします。このコマンドは **feature-nv** CLI コマンドとは相互に排他的です。
ステップ 7	**feature nv overlay** 例： `switch(config)#feature nv overlay`	セグメント ルーティング レイヤ 2 EVPNに使用されるNVE機能を有効にします。
ステップ 8	**nv overlay evpn** 例： `switch(config)#nv overlay evpn`	EVPN を有効にします。
ステップ 9	**interface loopback** *Interface_Number* 例： `switch(config)#interface loopback 1`	NVEのループバック インターフェイスを設定します。
ステップ 10	**ip address** *address* 例： `switch(config-if)#ip address 192.168.15.1`	IP アドレスを設定します。
ステップ 11	**exit** 例： `switch(config-if)#exit`	グローバルアドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 12	**evpn** 例： `switch(config)#evpn`	EVPN コンフィギュレーション モードを開始します。

ステップ 13	**evi** *number*
	例：
	`switch(config-evpn)#evi 1000`
	`switch(config-evpn-sr)#`
		レイヤ 2 EVIを設定します。必要であれば、自動生成された EVI に基づいて RT を手動で構成できます。

ステップ 14	**encapsulation mpls** 例： `switch(config-evpn)#encapsulation mpls`	MPLSカプセル化と入力レプリケーションを有効にします。

	コマンドまたはアクション	目的
ステップ 15	**source-interface loopback** *Interface_Number* 例： `switch(config-evpn-nve-encap)#source-interface loopback 1`	NVE送信元インターフェイスを指定します。
ステップ 16	**exit** 例： `switch(config-evpn-nve-encap)#exit`	設定を終了します。
ステップ 17	**vrf context** *VRF_NAME* 例： `switch(config)#vrf context Tenant-A`	VRF を設定します。
ステップ 18	**evi** *EVI_ID* 例： `switch(config-vrf)#evi 30001`	L3 EVI を設定します。
ステップ 19	**exit** 例： `switch(config-vrf)#exit`	設定を終了します。
ステップ 20	**VLAN** *VLAN_ID* 例： `switch(config)#vlan 1001`	VLAN を設定します。
ステップ 21	**evi auto** 例： `switch(config-vlan)#evi auto`	L2 EVI を設定します。
ステップ 22	**exit** 例： `switch(config-vlan)#exit`	
ステップ 23	**router bgp** *autonomous-system-number* 例： `switch(config)#router bgp 1`	BGP コンフィギュレーション モードを開始します。
ステップ 24	**address-family l2vpn evpn** 例： `switch(config-router)#address-family l2vpn evpn`	EVPN アドレス ファミリをグローバルに有効にします。

	コマンドまたはアクション	目的
ステップ 25	**neighbor address** *remote-as autonomous-system-number* 例： `switch(config-router)#neighbor 192.169.13.1 remote as 2`	BGP ネイバーを設定します。
ステップ 26	**address-family l2vpn evpn** 例： `switch(config-router-neighbor)#address-family l2vpn evpn`	ネイバーの EVPN アドレス ファミリを有効にします。
ステップ 27	**encapsulation mpls** 例： `switch(config-router-neighbor)#encapsulation mpls`	MPLS カプセル化を有効にします。
ステップ 28	**send-community extended** 例： `switch(config-router-neighbor)#send-community extended`	BGP を設定し、拡張コミュニティ リストをアドバタイズします。
ステップ 29	**vrf** *VRF_NAME* 例： `switch(config-router)#vrf Tenant-A`	BGP VRF を設定します。
ステップ 30	**exit** 例： `switch(config-router)#exit`	設定を終了します。

EVI 用の VLAN の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**vlan** *number*	VLAN を設定します。
ステップ 2	**evi** *[auto]*	VLAN の BD ラベルを作成します。このラベルは、セグメント ルーティング レイヤ 2 EVPN 全体で VLAN の識別子として使用されます。

NVE インターフェイスの設定

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**interface loopback** *loopback_number* 例： `switch(config)# interface loopback 1`	IPアドレスをこのループバック インターフェイスに関連付け、この IP アドレスをセグメント ルーティング設定に使用します。
ステップ 3	**ip address** 例： `switch(config-if)#ip address 192.169.15.1/32`	IPv4 アドレス ファミリを指定し、ルータ アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 4	**evpn** 例： `switch(config)#evpn`	EVPN 設定モードを開始します。
ステップ 5	**encapsulation mpls** 例： `switch(config-evpn)# encapsulation mpls`	MPLS カプセル化と入力レプリケーションを有効にします。
ステップ 6	**source-interface** *loopback_number* 例： `switch(config-evpn-nve-encap)#source-interface loopback 1`	NVE 送信元インターフェイスを指定します。
ステップ 7	**exit** 例： `switch(config)# exit`	セグメント ルーティング モードを終了し、コンフィギュレーション端末モードに戻ります。

VRF 下での EVI の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**vrf context** テナント	VRF テナントを作成します。

	コマンドまたはアクション	目的
ステップ 2	**evi** *number*	VRF 下でレイヤ 3 EVI を設定します。

エニーキャスト ゲートウェイの設定

ファブリック転送の設定は、SVIがエニーキャストモードで設定されている場合にのみ必要です。

手順

	コマンドまたはアクション	目的
ステップ 1	**fabric forwarding anycast-gateway-mac 0000.aabb.ccdd**	分散ゲートウェイの仮想MACアドレスを設定します。
ステップ 2	**fabric forwarding mode anycast-gateway**	インターフェイスコンフィギュレーション モードで SVI をエニーキャスト ゲートウェイと関連付けます。

ループバック インターフェイスのラベル付きパスのアドバタイズ

レイヤ2 EVPNエンドポイントとしてアドバタイズされるループバック インターフェイスは、ラベル インデックスにマッピングする必要があります。これにより、BGP は、同じものに対応する MPLS ラベル付きパスをアドバタイズします。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	[**no**]**router ospf** *process* 例： `switch(config)# router ospf test`	OSPF モードを有効にします。
ステップ 3	**segment-routing** 例： `switch(config-router)# segment-routing mpls`	OSPF でのセグメント ルーティング機能を設定します。

	コマンドまたはアクション	目的
ステップ 4	**connected-prefix-sid-map** 例： `switch(config-sr-mpls)# connected-prefix-sid-map`	ローカル プレフィックスと SID のアドレス ファミリ固有のマッピングを設定できるサブモードを開始します。
ステップ 5	**address-family ipv4** 例： `switch(config-sr-mpls-conn)# address-family ipv4`	IPv4 アドレス プレフィックスを指定します。
ステップ 6	**1.1.1.1/32 index 100** 例： `switch(config-sr-mpls-conn-af)# 1.1.1.1/32 100`	SID 100 にアドレス 1.1.1.1/32 を関連付けます。
ステップ 7	**exit-address-family** 例： `switch(config-sr-mpls-conn-af)# exit-address-family`	アドレス ファミリを終了します。

SRv6 静的プレフィックス単位 TE について

SRv6 静的プレフィックス単位 TE 機能を使用すると、デフォルト以外の VRF にマッピングされたプレフィックスをマッピングおよびアドバタイズできます。この機能により、一致する VRF ルート ターゲットを使用して単一のインスタンスで複数のプレフィックスをアドバタイズでき、各プレフィックスを手動で入力する必要がなくなります。

Cisco NX-OS リリース 9.3(5) では、1 つの VNF だけが VM にサービスを提供できます。

SRv6 の静的なプレフィックスごとの TE の設定

始める前に

次の手順を実行します。

- **install feature-set mpls** コマンドと **feature-set mpls** コマンドを使用して、MPLS 機能セットをインストールして有効にする必要があります。
- MPLS セグメント ルーティング機能を有効にする必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**vrf context** *VRF_Name* 例： `switch(config)# vrf context vrf_2_7_8`	VRF を定義し、VRF コンフィギュレーション モードを開始します。
ステップ 3	**rd** *rd_format* 例： `switch(config-vrf)# rd 2.2.2.0:2`	RD を VRF に割り当てます。
ステップ 4	**address-family** {**ipv4** | **ipv6** } 例： `switch(config-vrf)# address-family ipv4 unicast`	VRF インスタンス用に IPv4 または IPv6 アドレスファミリを指定し、アドレスファミリ コンフィギュレーション モードを開始します。
ステップ 5	**route-target import** *route-target-id* 例： `switch(config-vrf)# route-target import 1:2`	VRF へのルートのインポートを設定します。
ステップ 6	**route-target import** *route-target-id* **evpn** 例： `switch(config-vrf)# route-target import 1:2 evpn`	一致するルート ターゲット値を持つ、レイヤ3 EVPN から VRF へのルートのインポートを設定します。
ステップ 7	**route-target export** *route-target-id* 例： `switch(config-vrf)# route-target export 1:2`	VRF からのルートのエクスポートを設定します。
ステップ 8	**route-target export** *route-target-id* **evpn** 例： `switch(config-vrf)# route-target export 1:2 evpn`	一致するルート ターゲット値を持つ、VPN から レイヤ3 EVPN からへのルートのエクスポートを設定します。
ステップ 9	**router bgp** *autonomous-system-number* 例： `switch(config)# router bgp 65000`	BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。

	コマンドまたはアクション	目的
ステップ 10	**router-id** *id* 例： `switch(config-router)# router-id 2.2.2.0`	ルータ ID を設定します。
ステップ 11	**address-family l2vpn evpn** 例： `switch(config-router-af)# address-family l2vpn evpn`	レイヤ 2 VPN EVPN のグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 12	**neighbor** *ipv4-address* **remote-as**
	例：
	`switch(config-router)# neighbor 7.7.7.0 remote-as 65000`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの IPv4 アドレスおよび AS 番号を設定します。

ステップ 13	**update-source loopback** *number* 例： `switch(config-router-neighbor)# update-source loopback0`	ループバック番号を指定します
ステップ 14	**address-family l2vpn evpn** 例： `switch(config-router-neighbor)#address-family l2vpn evpn`	ネイバーの EVPN アドレス ファミリを有効にします。
ステップ 15	**send-community extended** 例： `switch(config-router-neighbor)#send-community extended`	BGP を設定し、拡張コミュニティ リストをアドバタイズします。
ステップ 16	**encapsulation mpls** 例： `switch(config-router-neighbor)#encapsulation mpls`	MPLS カプセル化を有効にします。
ステップ 17	**exit** 例： `switch(config-router-neighbor)#exit`	設定を終了します。

例

次の例は、VRF VT を定義するために RPM 構成を設定する方法を示しています。

```
rf context vrf_2_7_8
          rd 2.2.2.0:2
```

```
          address-family ipv4 unicast
            route-target import 0.0.1.1:2
            route-target import 0.0.1.1:2 evpn
            route-target export 0.0.1.1:2
            route-target export 0.0.1.1:2 evpn
ip extcommunity-list standard vrf_2_7_8-test permit rt 0.0.1.1:2
            route-map Node-2 permit 4
            match extcommunity vrf_2_7_8-test
            set extcommunity color 204
```

RD Auto について

自動派生ルート識別子（rd auto）は、IETF RFC 4364 セクション 4.2 で説明されているタイプ 1 エンコーディング形式に基づいています。https://tools.ietf.org/html/rfc4364#section-4.2タイプ 1 エンコーディングでは、4 バイトの管理フィールドと 2 バイトの番号フィールドを使用できます。Cisco NX-OS 内では、自動導出 RD は、4 バイトの管理フィールド（RID）としての BGP ルータ ID の IP アドレスと、2 バイトの番号フィールド（VRF ID）の内部 VRF ID を使用して構築されます。

2 バイトの番号付けフィールドは常に VRF から取得されますが、IP-VRF または MAC-VRF での使用に応じて異なる番号付け方式になります。

- IP-VRF の 2 バイトの番号付けフィールドは、1 から始まる内部 VRF ID を使用します。VRF ID 1 および 2 は、それぞれデフォルト VRF および管理 VRF 用に予約されています。最初のカスタム定義 IP VRF は VRF ID 3 を使用します。
- MAC-VRF の 2 バイトの番号付けフィールドは、VLAN ID + 32767 を使用します。その結果、VLAN ID 1 は 32768 になります。

例：自動取得ルート識別子（RD）

- BGP ルータ ID 192.0.2.1 および VRF ID 6-RD 192.0.2.1:6 の IP-VRF
- BGP ルータ ID 192.0.2.1 および VLAN 20-RD 192.0.2.1:32787 の MAC-VRF

Route-Target Auto について

自動派生Route-Target（route-target import/export/both auto）は、IETF RFC 4364 セクション 4.2（https://tools.ietf.org/html/rfc4364#section-4.2）で説明されているタイプ 0 エンコーディング形式に基づいています。IETF RFC 4364 セクション 4.2 ではルート識別子形式について説明し、IETF RFC 4364 セクション 4.3.1では、Route-Target に同様の形式を使用することが望ましいとしています。タイプ 0 エンコーディングでは、2 バイトの管理フィールドと 4 バイトの番号フィールドを使用できます。Cisco NX-OS 内では、自動派生 Route-Target は、2 バイトの管理フィールドとして自律システム番号（ASN）、4 バイトの番号フィールドのサービス識別子（EVI）で構成されます。

2 バイト ASN

タイプ 0 エンコーディングでは、2 バイトの管理フィールドと 4 バイトの番号フィールドを使用できます。Cisco NX-OS 内では、自動派生 Route-Target は、2 バイトの管理フィールドとし

ての自律システム番号（ASN）と、4 バイトの番号フィールドのサービス識別子（EVI）で構成されます。

自動派生 Route-Target（RT）の例：

- ASN 65001 と L3EVI 50001 内の IP-VRF： Route-Target 65001:50001
- ASN 65001 と L2VNI 30001 内の MAC-VRF：Route-Target 65001:30001

Multi-AS 環境では、Route-Target を静的に定義するか、Route-Target の ASN 部分と一致するように書き換える必要があります。

（注） 4 バイト ASN の自動派生 Route-Target はサポートされていません。

4 バイト ASN

タイプ 0 エンコーディングでは、2 バイトの管理フィールドと 4 バイトの番号フィールドを使用できます。Cisco NX-OS 内では、自動派生 Route-Target は、2 バイトの管理フィールドとしての自律システム番号（ASN）と、4 バイトの番号フィールドのサービス識別子（EVI）で構成されます。4 バイト長の ASN 要求と 24 ビット（3 バイト）を必要とする EVI では、拡張コミュニティ内のサブフィールド長が使い果たされます（2 バイト タイプと 6 バイト サブフィールド）。長さと形式の制約、およびサービス識別子（EVI）の一意性の重要性の結果、4 バイトの ASN は、IETF RFC 6793 セクション 9（https://tools.ietf.org/html/rfc6793#section-9）で説明されているように、AS_TRANS という名前の 2 バイトの ASN で表されます。2 バイトの ASN 23456 は、4 バイトの ASN をエイリアスする特別な目的の AS 番号であるAS_TRANSとして IANA（https://www.iana.org/assignments/iana-as-numbers-special-registry/iana-as-numbers-special-registry.xhtml）によって登録されます。

4 バイトの ASN（AS_TRANS）を使用した自動派生 Route-Target（RT）の例：

- ASN 65656 と L3VNI 50001 内の IP-VR：Route-Target 23456:50001
- ASN 65656 とL2VNI 30001 内の MAC-VRF：Route-Target 23456:30001

BD 用の RD およびルート ターゲットの設定

VLAN で evi auto を設定すると、ブリッジ ドメイン (BD) RD およびルート ターゲットが自動的に生成されます。BD RD およびルート ターゲットを手動で設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	グローバル コンフィギュレーション モードを開始します

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	
ステップ 2	**evpn** 例： `switch(config)# evpn`	EVPN 設定モードを開始します。
ステップ 3	**evi** *VLAN_ID* 例： `switch(config-evpn)# evi 1001`	RD/ルート ターゲットを設定するための L2 EVI を指定します。
ステップ 4	**rd** *rd_format* 例： `switch(config-evpn-evi-sr)# rd 192.1.1.1:33768`	RD を設定します。
ステップ 5	**route-target both** *rt_format* 例： `switch(config-evpn-evi-sr)# route-target both 1:20001`	ルートターゲットを設定します。

VRF用 の RD およびルート ターゲットの設定

VRF で **evi** *evi_ID* を設定すると、VRF RD およびルート ターゲットが自動的に生成されます。VRF RD およびルート ターゲットを手動で設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**vrf context** *VRF_NAME* 例： `switch(config)# vrf context A`	VRF を設定します。
ステップ 3	**rd auto** または **rd_format** 例： `switch(config-vrf)# rd auto`	RD を設定します。

	コマンドまたはアクション	目的
ステップ 4	**address-family ipv4 unicast** 例： `switch(config-vrf)# address-family ipv4 unicast`	IPv4アドレスファミリを有効にします。
ステップ 5	**route-target both** *rt_format* **evpn** 例： `switch(config-vrf-af-ipv4)# route-target both 1:30001 evpn`	ルートターゲットを設定します。

セグメント ルーティング MPLS 上のレイヤ 2 EVPN の設定例

次の例は、セグメント ルーティング MPLS を介したレイヤ 2 EVPN の設定を示しています。

```
install feature-set mpls
feature-set mpls
nv overlay evpn
feature bgp
feature mpls segment-routing
feature mpls evpn
feature interface-vlan
feature nv overlay

fabric forwarding anycast-gateway-mac 0000.1111.2222

vlan 1001
  evi auto

vrf context Tenant-A
  evi 30001

interface loopback 1
  ip address 192.168.15.1/32

interface vlan 1001
  no shutdown
  vrf member Tenant-A
  ip address 111.1.0.1/16
  fabric forwarding mode anycast-gateway

router bgp 1
  address-family l2vpn evpn
    neighbor 192.169.13.1
      remote-as 2
      address-family l2vpn evpn
        send-community extended
        encapsulation mpls
    vrf Tenant-A

evpn
  encapsulation mpls
    source-interface loopback 1
```

セグメントルーティングのVNFの比例マルチパスの設定

セグメント ルーティングの VNF の比例マルチパスについて

ネットワーク機能仮想化インフラストラクチャ（NFVi）では、サービス ネットワーク（ポータブル IP）が仮想ネットワーク機能（VNF）によりアドバタイズされます。VNF は、ポータブル IP ゲートウェイ (PIP-GW) とも呼ばれ、VNF 内の VM 間でデータ パケットをルーティングします。セグメント ルーティング機能の VNF の比例マルチパスにより、EVPN アドレス ファミリでサービス ネットワーク（PIP）の VNF をアドバタイズできます。VNF の IP アドレスは、サービス ネットワークの EVPN IP プレフィックス ルート NLRI アドバタイズメントの「ゲートウェイ IP アドレス」フィールドでエンコードされます。

VNF の IP アドレスをアドバタイズすることにより、EVPN ファブリックの入力ノードは、VNF IP アドレスを VNF に接続されたリーフに再帰的に解決します。リーフは、サービス ネットワーク（PIP）をアドバタイズするのと同じノードである可能性があります。

ルートインジェクタは、IPv4 または IPv6 AF にルートを挿入する BGP プロトコルです。この場合、ルートインジェクタは、ネクスト ホップが VNF として設定されている VM にルートを挿入します。

ルート インジェクタとは異なり、VNF はルーティング プロトコルに参加して、VM の到達可能性をアドバタイズできます。サポートされているプロトコルは、eBGP、IS-IS、および OSPF です。

セグメント ルーティングの VNF の比例マルチパスの有効化

セグメント ルーティング機能の VNF の比例マルチパスを有効にして、ネクストホップ パスを保持することにより、IGP または静的ルートのルートを再配布できます。その後、再構築された EVPN タイプ 5 ルートのゲートウェイ IP をエクスポートしてアドバタイズできます。

Cisco NX-OS リリース 9.3(5) では、1 つの VNF だけが VM にサービスを提供できます。

始める前に

次の手順を実行します。

- **install feature-set mpls** コマンドと **feature-set mpls** コマンドを使用して、MPLS 機能セットをインストールして有効にします。
- MPLS セグメント ルーティング機能を有効化します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal switch(config)#`	グローバル コンフィギュレーション モードに入ります。
ステップ 2	**route-map** *export-l2evpn-rtmap* **permit** *10* 例： `switch(config)# route-map export-l2evpn-rtmap permit 10`	<<説明が必要>>
ステップ 3	**match ip address prefix-list** *pip-pfx-list* 例： `switch(config-route-map)# match ip prefix-list vm-pfx-list`	PIP-GW をゲートウェイとしてアドバタイズする必要があるプレフィックスを定義します。
ステップ 4	**set evpn gateway-ip use-nexthop** 例： `switch(config-route-map)# set evpn gateway-ip use-nexthop`	gateway-ip をアドバタイズするための特定のルートを定義します。
ステップ 5	**vrf context** *VRF_Name* 例： `switch(config-route-map)# vrf context vrf switch(config-route-map)# address-family ipv4 unicast switch(config-route-map)# export map export-l2evpn-rtmap`	ルート マップを vrf コンテキストに適用します。
ステップ 6	**address-family ipv4 unicast** 例： `switch(config-route-map)# address-family ipv4 unicast switch(config-route-map)# export map export-l2evpn-rtmap`	ルート マップを vrf コンテキストに適用します。
ステップ 7	**export map** *export-l2evpn-rtmap* 例： `switch(config-route-map)# export map export-l2evpn-rtmap`	ルート マップを vrf コンテキストに適用します。

	コマンドまたはアクション	目的
ステップ 8	**router bgp** *number* 例： `switch(config)# router bgp 100`	BGP を設定します。
ステップ 9	**vrf** *VRF_Name* 例： `switch(config-route-map)# vrf vrf3`	ルート マップを vrf コンテキストに適用します。
ステップ 10	**address-family ipv4 unicast** 例： `switch(config-router)# address-family ipv4 unicast`	IPv4 のアドレス ファミリを設定します。
ステップ 11	**export-gateway-ip** 例： `switch(config-route-map)# export-gateway-ip`	gateway-ip をエクスポートしてアドバタイズして、EVPN タイプ 5 ルートを再接続します。 （注） gateway-ip のエクスポートと EVPN ゲートウェイ構成の設定は同時に実行できます。同時に設定すると、すべてのプレフィックスがゲートウェイ IP とともにエクスポートされます。

vPC マルチホーミング

マルチホーミングについて

Cisco Nexus プラットフォーム スイッチは、vPC ベースのマルチホーミングをサポートします。このマルチホーミングでは、スイッチのペアが冗長性のために単一のデバイスとして機能し、両方のスイッチがアクティブ モードで機能します。EVPN 環境の Cisco Nexus プラットフォーム スイッチでは、レイヤ 2 マルチホーミングをサポートする 2 つのソリューションがあります。これらのソリューションは、MCT リンクが必要な従来の vPC（エミュレートまたは仮想 IP アドレス）と BGP EVPN 技術に基づいています。

BGP EVPN コントロール プレーンを使用している間、各 vPC ペアは共通の仮想 IP（VIP）を使用して、アクティブ/アクティブの冗長性を提供します。さらに、BGP EVPN ベースのマルチホーミングは、特定の障害シナリオで高速コンバージェンスを提供します。

vPC ピア上の BD ごとのラベル

vPC ピアが同じ BD ごとのラベルを持つようにするには、BD ごとのラベルに次の値を指定する必要があります。

```
Label value = Label_base + VLAN_ID
```

ラベル ベースは、同じ vPC ピアで設定されます。現在、VLAN 設定は両方の vPC ピアで同一であるため、両方の vPC ピアに同じラベルが付けられます。

Cisco NX-OS リリース 9.3(1) では、BD ごとのラベルの設定はサポートされていません。このリリースでは、evi auto のみがサポートされています。

vPC ピア上の VRF ごとのラベル

vPC ピアが同じ VRF ごとのラベルを持つようにするには、VRF ごとのラベルに次の値を指定する必要があります。

```
Label value = Label_base + vrf_allocate_index
```

vPC ピアの割り当てインデックスを設定するには、次の手順を実行します。

```
Router bgp 1
  vrf Tenant_A
    allocate-index 11
```

バックアップ リンクの設定

バックアップ リンクは、vPC ピア間で設定する必要があります。このリンクとしては、MCT に並列な任意のレイヤ 3 リンクが可能です。

例

```
interface vlan 100
    ip add 10.1.1.1/24
        mpls ip forwarding

< enable underlay protocol >
```

vPC マルチホーミング ピアリングの注意事項と制約事項

vPC マルチホーミング ピアリングには、次の注意事項と制約事項があります。

- ESI ベースのマルチホーミングはサポートされていません。
- 物理および仮想セカンダリ IP アドレスは、両方とも MPLS ラベル付きパスを介してアドバタイズされる必要があります。
- vPC 整合性チェックは、BD ごとのラベル設定ではサポートされていません。

vPC マルチホーミングの設定例

次の例は、vPC マルチホーミングの設定を示しています。

- vPC プライマリ

```
interface loopback1
  ip address 192.169.15.1/32
  ip address 192.169.15.15/32 secondary

evpn
  encapsulation mpls
    source-interface loopback1

vlan 101
  evi auto

vrf context A
  evi 301

router bgp 1
  vrf A
    allocate-index 1001
```

- vPC セカンダリ

```
interface loopback1
  ip address 192.169.15.2/32
  ip address 192.169.15.15/32 secondary

evpn
  encapsulation mpls
    source-interface loopback1

vlan 101
  evi auto

vrf context A
  evi 301

router bgp 1
  vrf A
    allocate-index 1001
```

セグメント ルーティング MPLS を介したレイヤ 3 EVPN およびレイヤ 3 VPN の構成

このセクションでは、レイヤ 3 EVPN を設定するタスクと、L3 EVPN および L3VPN ルータのスティッチングについて説明します。構成を完了するには、次の作業を実行します。

インポートおよびエクスポートルール用のVRFおよびルートターゲットの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal**	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**vrf** *vrf-name*	VPN ルーティングおよび転送（VRF）インスタンスを定義し、VRF コンフィギュレーション モードを開始します。
ステップ 3	**rd auto**	一意のルート識別子（RD）をVRFに自動的に割り当てます。
ステップ 4	**address-family** { **ipv4 | ipv6** } **unicast**	VRF インスタンス用に IPv4 または IPv6 アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション サブモードを開始します。
ステップ 5	**route-target import** *route-target-id*	一致するルートターゲット値を持つ、L3 VPN BGP NLRI から VRF へのルートのインポートを設定します。
ステップ 6	**route-target export** *route-target-id*	VRF から L3VPN BGP NLRI へのルートのエクスポートを設定し、指定されたルートターゲット識別子を L3VPN BGP NLRI に割り当てます。
ステップ 7	**route-target import** *route-target-id* **evpn**	一致するルートターゲット値を持つ L3 EVPN BGP NLRI からのルートのインポートを設定します。
ステップ 8	**route-target export** *route-target-id* **evpn**	VRF から L3 EVPN BGP NLRI へのルートのエクスポートを設定し、指定されたルートターゲット識別子を BGP EVPN NLRI に割り当てます。

BGP EVPN およびラベル割り当てモードの設定

encapsulation mpls コマンドを使用して MPLS トンネル カプセル化を使用できます。EVPN アドレス ファミリのラベル割り当てモードを設定できます。NX-OS の IP ルート タイプの EVPN でのデフォルトのトンネル カプセル化は VXLAN です。

BGP EVPN を介した Cisco Nexus 9000 シリーズ スイッチからの（IP またはラベル）バインディングのアドバタイズにより、リモート スイッチはルーティングされたトラフィックをその IP に送信できます。その際、MPLS を介して IP をアドバタイズしたスイッチへの IP のラベルを使用します。

IP プレフィックス ルート（タイプ 5）は次のとおりです。

- MPLS カプセル化によるタイプ 5 ルート

```
RT-5 Route - IP Prefix

RD:  L3 RD
IP Length:  prefix length
IP address:  IP (4 bytes)
Label1:  BGP MPLS Label
Route Target
RT for IP-VRF
```

デフォルトのラベル割り当てモードは、MPLS 上のレイヤ 3 EVPN の VRF 単位です。

BGP EVPN とラベル割り当てモードを設定するには、次の手順を実行します。

始める前に

install feature-set mpls コマンドと **feature-set mpls** コマンドを使用して、MPLS 機能セットをインストールして有効にする必要があります。

MPLS セグメント ルーティング機能を有効にする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal**	グローバル コンフィギュレーション モードを開始します

ステップ 2	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
		BGP プロセスおよび関連する設定を削除するには、このコマンドで **no** オプションを使用します。

ステップ 3	必須: **address-family l2vpn evpn** 例： `switch(config-router)# address-family`	レイヤ 2 VPN EVPN のグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	l2vpn evpn switch(config-router-af)#	
ステップ 4	必須: **exit** 例： switch(config-router-af)# exit switch(config-router)#	グローバルアドレスファミリ コンフィギュレーション モードを終了します。
ステップ 5	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number* 例： switch(config-router)# neighbor 10.1.1.1 remote-as 64497 switch(config-router-neighbor)#	リモート BGP ピアの IPv4 アドレスおよび AS 番号を設定します。
ステップ 6	**address-family l2vpn evpn** 例： switch(config-router-neighbor)# address-family l2vpn evpn switch(config-router-neighbor-af)#	ラベル付きのレイヤ 2 VPN EVPN をアドバタイズします。
ステップ 7	**encapsulation mpls** 例： router bgp 100 address-family l2vpn evpn neighbor NVE2 remote-as 100 address-family l2vpn evpn send-community extended encapsulation mpls vrf foo address-family ipv4 unicast advertise l2vpn evpn BGP セグメント ルーティング設定： router bgp 100 address-family ipv4 unicast network 200.0.0.1/32 route-map label_index_pol_100 network 192.168.5.1/32 route-map label_index_pol_101 network 101.0.0.0/24 route-map label_index_pol_103 allocate-label all neighbor 192.168.5.6 remote-as 20 address-family ipv4 labeled-unicast send-community extended	BGP EVPN アドレス ファミリを有効にし、EVPN タイプ 5 ルート アップデートをネイバーに送信します。 （注） NX-OS の IP ルート タイプの EVPN でのデフォルトのトンネル カプセル化は VXLAN です。これをオーバーライドするために、MPLS トンネルのカプセル化を示す新しい CLI が導入されています。
ステップ 8	**vrf** <*customer_name*>	VRF を設定します。

	コマンドまたはアクション	目的
ステップ 9	**address-family ipv4 unicast**	IPv4 アドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 10	**advertise l2vpn evpn**	レイヤ 2 VPN EVPN をアドバタイズします。
ステップ 11	**redistribute direct route-map DIRECT_TO_BGP**	直接接続されたルートを BGP-EVPN に再配布します。
ステップ 12	**label-allocation-mode per-vrf**	ラベル割り当てモードを VRF 単位に設定します。プレフィックス単位のラベル モードを設定する場合は、**no label-allocation-mode per-vrf** CLI コマンドを使用します。 EVPN アドレス ファミリの場合、デフォルトのラベル割り当ては VRF 単位です。一方、ラベル割り当て CLI がサポートされている他のアドレス ファミリではプレフィックス単位モードです。実行コンフィギュレーションでは、CLI の no 形式は表示されません。

例

プレフィックス単位のラベル割り当ての設定については、次の例を参照してください。

```
router bgp 65000
    [address-family l2vpn evpn]
    neighbor 10.1.1.1
        remote-as 100
        address-family l2vpn evpn
        send-community extended
    neighbor 20.1.1.1
        remote-as 65000
        address-family l2vpn evpn
        encapsulation mpls
        send-community extended
    vrf customer1
        address-family ipv4 unicast
            advertise l2vpn evpn
            redistribute direct route-map DIRECT_TO_BGP
            no label-allocation-mode per-vrf
```

BGP レイヤ 3 EVPN およびレイヤ 3 VPN スティッチングの構成

同じルーターでスティッチングを構成するには、レイヤー 3 VPN ネイバー関係とルーター アドバタイズメントを構成します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch# configure terminal`
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
		BGP プロセスおよび関連する設定を削除するには、このコマンドで **no** オプションを使用します。

ステップ 3	**address-family** {**vpnv4** | **vpnv6**} **unicast**
	例：
	`switch(config-router)# address-family vpnv4 unicast`
	`switch(config-router-af)# address-family vpnv6 unicast`
	`switch(config-router-af)#`
		レイヤ 3 VPNv4 または VPNv6 に対するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 4	**exit**
	例：
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		グローバルアドレスファミリ コンフィギュレーション モードを終了します。

ステップ 5	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number* 例： `switch(config-router)# neighbor 20.1.1.1 remote-as 64498`	リモート BGP L3VPN ピアの IPv4 アドレスおよび AS 番号を設定します。

ステップ 6	**address-family** {**vpnv4** | **vpnv6**} **unicast**
	例：
	`switch(config-router)# address-family vpnv4 unicast`
	`switch(config-router-af)# address-family vpnv6 unicast`
	`switch(config-router-af)#`
		VPNv4 または VPNv6 のアドレスファミリのネイバーを設定します。

ステップ 7	**send-community extended**	BGP VPN アドレス ファミリを有効にします

	コマンドまたはアクション	目的
ステップ 8	**import l2vpn evpn reoriginate**	標準のルートターゲット識別子と一致するルートターゲット識別子を持つレイヤ 3 BGP EVPN NLRI からのルーティング情報のインポートを設定し、このルーティング情報を、スティッチング ルートターゲット識別子に割り当てる再発信の後に、BGP EVPN ネイバーへエクスポートします。

ステップ 9	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
	例：
	`switch(config-router)# neighbor 10.1.1.1 remote-as 64497`
	`switch(config-router-neighbor)#`
		リモート レイヤ 3 EVPN BGP ピアの IPv4 アドレスおよび AS 番号を設定します。

ステップ 10	**address-family** {**l2vpn** | **evpn**
	例：
	`switch(config-router-neighbor)# address-family l2vpn evpn`
	`switch(config-router-neighbor-af)#`
		レイヤ 3 EVPN のネイバー アドレス ファミリを設定します。

ステップ 11	**import vpn unicast reoriginate**	スティッチング ルートターゲット識別子と一致するルートターゲット識別子を持つ BGP EVPN NLRI からのルーティング情報のインポートを有効にし、この再発信後のルーティング情報をレイヤ 3 VPN BGP ネイバーにエクスポートします。
ステップ 12	**vrf** <*customer_name*>	VRF を設定します。
ステップ 13	**address-family ipv4 unicast**	IPv4 アドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 14	**advertise l2vpn evpn**	レイヤ 2 VPN EVPN をアドバタイズします。

例

```
vrf context Customer1
    rd auto
    address-family ipv4 unicast
        route-target import 100:100
        route-target export  100:100
        route-target import 100:100 evpn
```

```
        route-target export 100:100 evpn

segment-routing
    mpls
      global-block 11000 20000
      connected-prefix-sid
         address-family ipv4 unicast
         200.0.0.1 index 101
!
int lo1
  ip address 200.0.0.1/32
!
interface e1/13
  description "MPLS interface towards Core"
  ip address 192.168.5.1/24
  mpls ip forwarding
  no shut

router bgp 100
address-family ipv4 unicast
allocate-label all
address-family ipv6 unicast
address-family l2vpn evpn
address-family vpnv4 unicast
address-family vpnv6 unicast
 neighbor 10.0.0.1 remote-as 200
    update-source loopback1
    address-family vpnv4 unicast
      send-community extended
      import l2vpn evpn reoriginate
    address-family vpnv6 unicast
      import l2vpn evpn reoriginate
      send-community extended
  neighbor 20.0.0.1 remote-as 300
    address-family l2vpn evpn
      send-community extended
      import vpn unicast reoriginate
      encapsulation mpls
  neighbor 192.168.5.6 remote-as 300
      address-family ipv4 labeled-unicast
  vrf Customer1
    address-family ipv4 unicast
      advertise l2vpn evpn
    address-family ipv6 unicast
      advertise l2vpn evpn
```

レイヤー 3 EVPN およびレイヤー 3 VPN を有効にする機能の設定

始める前に

VPN ファブリック ライセンスをインストールします。

feature interface-vlan コマンドが有効になっていることを確認してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**feature bgp**	BGP 機能と構成を有効にします。
ステップ 2	**install feature-set mpls**	MPLS 構成コマンドを有効にします。
ステップ 3	**feature-set mpls**	MPLS 構成コマンドを有効にします。
ステップ 4	**feature mpls segment-routing**	セグメント ルーティング構成コマンドを有効にします。
ステップ 5	**feature mpls evpn**	EVPN over MPLS 構成コマンドを有効にします。このコマンドは **feature-nv** CLI コマンドとは相互に排他的です。
ステップ 6	**feature mpls l3vpn**	EVPN over MPLS 構成コマンドを有効にします。このコマンドは **feature-nv** CLI コマンドとは相互に排他的です。

セグメント ルーティングを介した BGP L3 VPN の構成

始める前に

install feature-set mpls コマンドと **feature-set mpls** コマンドを使用して、MPLS 機能セットをインストールして有効にする必要があります。

MPLS セグメント ルーティング機能を有効にする必要があります。

feature mpls l3vpn コマンドを使用して、MPLS L3 VPN 機能を有効にする必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

	コマンドまたはアクション	目的
		BGP プロセスおよび関連する設定を削除するには、このコマンドで **no** オプションを使用します。

ステップ 3	**address-family** {**vpnv4** | **vpnv6**} **unicast**
	例：
	`switch(config-router)# address-family vpnv4 unicast`
	`switch(config-router-af)# address-family vpnv6 unicast`
	`switch(config-router-af)#`
		レイヤ 3 VPNv4 または VPNv6 に対するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 4	[**no**] **allocate-label option-b**	AS 間オプション b を無効にします

ステップ 5	必須: **exit**
	例：
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		グローバルアドレス ファミリ コンフィギュレーション モードを終了します。

ステップ 6	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
	例：
	`switch(config-router)# neighbor 20.1.1.1 remote-as 64498`
	`switch(config-router-neighbor)#`
		リモート BGP L3VPN ピアの IPv4 アドレスおよび AS 番号を設定します。

ステップ 7	**address-family** {**vpnv4** | **vpnv6** } **unicast**
	例：
	`switch(config-router-neighbor)# address-family vpnv4 unicast`
	`switch(config-router-neighbor-af)#`
		VPNv4 または VPNv6 のアドレスファミリのネイバーを設定します。

ステップ 8	**send-community extended**	BGP VPN アドレス ファミリを有効にします。
ステップ 9	**vrf** <*customer_name*>	VRF を設定します。
ステップ 10	**allocate-index x**	割り当てインデックスを設定します。
ステップ 11	**address-family ipv4 unicast**	IPv4 アドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 12	**redistribute direct route-map DIRECT_TO_BGP**	直接接続されたルートを BGP-L3VPN に再配布します。

SRTE 経由 BGP レイヤ 3 VPN

この機能により、データセンター相互接続（DCI）/WAN エッジ展開のセグメント ルーティング コアに対するトラフィック エンジニアリング機能が有効になります。DCI ハンドオフ（SR に基づき VxLAN から L3VPN へ、またはその逆）を可能にし、SR コアで SRTE 機能を使用できるため、さまざまなトラフィック クラスによって SLA を達成できます。SRTE 機能は、L3VPN プレフィックスに SR-Policy を適用することにより、DCI またはエッジ ルータに適用できます。L3VPN プレフィックスは、拡張コミュニティ カラーを設定した後（DCI またはエッジ ノードによって）アドバタイズでき、BGP L3VPN ネイバーは、そのカラーに基づいて SR ポリシーを適用して SRTE を作成できます。以下に、L3VPN プレフィックスで拡張コミュニティ カラーを構成するための構成を示します。

SRTE を介したレイヤ 3 VPN の構成に関する注意事項と制限事項

Cisco NX-OS リリース 10.1(2) 以降、セグメント ルーティング トラフィック エンジニアリングは、Cisco Nexus 9300-FX3、N9K-C9316D-GX、N9K-C93180YC-FX、N9K-C93240YC-FX2、および N9K- C9364C プラットフォーム スイッチ上でレイヤ 4 VPN を介してサポートされます。

この機能の制限は次のとおりです。

- アンダーレイ IPv6 はサポートされません。SRv6 は代替です。
- BGP の専用ファブリックにおける PCE の欠点のため、BGP アンダーレイを使用した PCE はサポートされていません。
- NXOS が BGP-LS で LSA をアドバタイズできないため、PCE を使用した OSPF-SRTE はサポートされていません。
- 合計 1000 の SRTE ポリシー スケール、BGP VPNv4 32K ルート、BGP VPNV6 32k ルート、および 1000 のアンダーレイ SR プレフィックスをサポートします。

Cisco NX-OS リリース 10.2(3)F 以降、カラー専用（CO）ビットのオプションがルート マップに追加されています。SRTE ポリシーを使用している特定のプレフィックスの CO ビットの値が変更された場合、BGP は古いポリシーを削除し、新しいポリシーを追加します。

拡張コミュニティ カラーの構成

このセクションは、次のトピックで構成されています。

入力ノードにおける拡張コミュニティ カラーの構成

SRTE ポリシーがインスタンス化される入力ノードによってプレフィックスが通知されるときに、入力ノードで拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**route-map** *map-name*
	例：
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ コンフィギュレーション モードを開始します。

ステップ 3	**set extcommunity color** *color-num*
	例：
	`switch(config-route-map)# set extcommunity color 20`
	`switch(config-route-map)#`
		カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ設定モードを終了します。

ステップ 5	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
		BGP プロセスおよび関連する設定を削除するには、このコマンドで no オプションを使用します。

ステップ 6	**neighbor** *ip-address*
	例：
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		BGP ネイバー テーブルまたはマルチプロトコル BGP ネイバー テーブルにエントリを追加します。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。

ステップ 7	**address-family vpnv4/vpnv6 unicast**
	例：
	`switch(config-router-neighbor)# address-family vpnv4/vpnv6 unicast`
	`switch(config-router-neighbor-af)#`
		vpnv4/vpnv6 アドレス ファミリ タイプのルータ アドレスファミリ構成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 8	**route-map** *map-name* **in**
	例：
	`switch(config-router-neighbor-af)# route-map ABC in`
	`switch(config-router-neighbor-af)#`
		構成された BGP ポリシーを受信ルートに適用します。
		マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

出力ノードでの拡張コミュニティ カラーの構成

プレフィックスが出力ノードによって通知されるときに、出力ノードで拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**route-map** *map-name*
	例：
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ コンフィギュレーション モードを開始します。

ステップ 3	**set extcommunity color** *color-num*
	例：
	`switch(config-route-map)# set extcommunity color 20`
	`switch(config-route-map)#`
		カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ設定モードを終了します。

ステップ 5	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

	コマンドまたはアクション	目的
		BGP プロセスおよび関連する設定を削除するには、このコマンドで no オプションを使用します。

ステップ 6	**neighbor** *ip-address*
	例：
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		BGP ネイバー テーブルまたはマルチプロトコル BGP ネイバー テーブルにエントリを追加します。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。

ステップ 7	**address-family vpnv4/vpnv6 unicast**
	例：
	`switch(config-router-neighbor)# address-family vpnv4/vpnv6 unicast`
	`switch(config-router-neighbor-af)#`
		vpnv4/vpnv6 アドレス ファミリ タイプのルータ アドレスファミリ構成モードを開始します。

ステップ 8	**route-map** *map-name* **out**
	例：
	`switch(config-router-neighbor-af)# route-map ABC out`
	`switch(config-router-neighbor-af)#`
		発信ルートに設定された BGP ポリシーを適用します。
		マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

出力ノードでのネットワーク/再配布コマンドの拡張コミュニティカラー構成

プレフィックスが出力ノードによって通知されるときに、出力ノードで network/redistribute コマンドの拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**route-map** *map-name*
	例：
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ コンフィギュレーション モードを開始します。

ステップ 3	**set extcommunity color** *color-num* 例：	カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。

	コマンドまたはアクション	目的
	`switch(config-route-map)# set extcommunity color 20` `switch(config-route-map)#`	

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ設定モードを終了します。

ステップ 5	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1;`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
		BGP プロセスおよび関連する設定を削除するには、このコマンドで no オプションを使用します。

ステップ 6	**vrf** <*customer_name*>	VRF を設定します。

ステップ 7	**address-family ipv4 unicast**
	例：
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		VRF インスタンスの IPv4 アドレス ファミリを指定し、アドレス ファミリ構成モードを開始します。

ステップ 8	**redistribute static route-map** *map-name* **out**
	例：
	`switch(config-router-vrf-af)# redistribute static route-map ABC`
	`switch(config-router-af)#`
		スタティック ルートを BGP に再配布します。マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

ステップ 9	**network** *ip-prefix* [**route-map** *map-name*]
	例：
	`switch(config-router-vrf-af)# network 1.1.1.1/32 route-map ABC`
	`switch(config-router-af-network)#`
		ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。

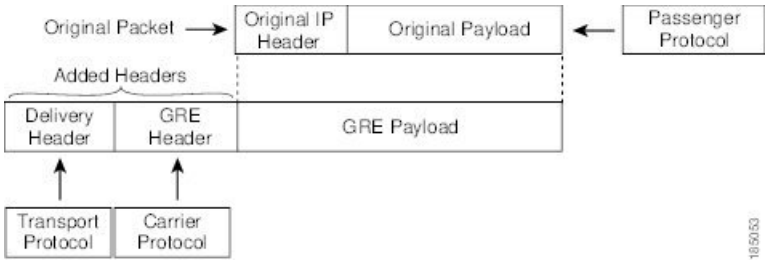
セグメント ルーティング MPLS および GRE トンネルの設定

GRE トンネル

Generic Routing Encapsulation（GRE）をさまざまなパッセンジャ プロトコルのキャリア プロトコルとして使用できます。

この次図は、GRE トンネルの IP トンネルのコンポーネントを示しています。オリジナルのパッセンジャ プロトコル パケットは GRE ペイロードとなり、デバイスはパケットに GRE ヘッダーを追加します。次にデバイスはトランスポート プロトコル ヘッダーをパケットに追加して送信します。

図 *5 : GRE PDU*



セグメント ルーティング MPLS および GRE

Cisco NX-OS リリース 9.3(1) 以降、Cisco Nexus デバイスではセグメント ルーティング MPLS とジェネリック ルーティング カプセル化(GRE)の両方を設定できます。これらのテクノロジーは両方ともシームレスに動作します。MPLS トンネルの終了後には、すべての MPLS トラフィックを GRE トンネルに転送できます。同様に、GRE の終了後には、GRE トンネルからのすべてのトラフィックを MPLS クラウドに転送できます。

すべての PE ルータは、別の GRE クラウドとの間で GRE トラフィックを開始、転送、または終了できます。同様に、すべてのトンネル通過ノードまたはトンネル エンド ノードは、MPLS トンネル カプセル化を設定できます。

Cisco Nexus 9000 スイッチでトンネルとセグメント ルーティングの両方が有効になっている場合、それぞれのフローの TTL 動作は次のとおりです。

- 着信 IP トラフィック、GRE ヘッダー付きの出力では、GRE ヘッダーの TTL 値は、着信 IP パケットの TTL 値より 1 少ない値です。
- 着信 IP トラフィック、MPLS ヘッダー付きの出力では、MPLS ヘッダーの TTL 値は、着信 IP パケットの TTL 値より 1 少ない値です。

- 着信 GRE トラフィック、MPLS ヘッダー付きの出力、MPLS ヘッダーの TTL 値はデフォルト (255) です。
- 着信 MPLS トラフィック、GRE ヘッダー付きの出力、GRE ヘッダーの TTL 値はデフォルト (255) です。

セグメント ルーティング MPLS および GRE の注意事項と制限事項

セグメント ルーティング MPLS および GRE には、次の注意事項と制限事項があります。

- トンネル パケットの入力統計はサポートされていません。
- default または template-mpls-heavy モードでのみサポートされます。
- MPLS セグメント ルーティングは、トンネル インターフェイスではサポートされていません。
- モジュラ スイッチのハードウェア制限により、トンネルの宛先IPアドレスの出力インターフェイスが Cisco Nexus 9300-FX/FX2 プラットフォーム スイッチを越える場合、トンネル Tx トラフィックはサポートされません。
- 最大 4 つの GRE トンネルがサポートされます。
- Cisco NX-OS リリース 9.3(3) 以降、Cisco Nexus 9300-GX プラットフォーム スイッチ上ではセグメント ルーティング MPLS と GRE の両方を設定できます。
- セグメント ルーティング MPLS と GRE の両方が共存している場合、トンネル Rx パケット カウンタは機能しません。

セグメント ルーティング MPLS および GRE の設定

静的 MPLS などの相互に排他的な MPLS 機能がイネーブルになっていない限り、MPLS セグメント ルーティングをイネーブルにできます。

始める前に

MPLS 機能セットは、**install feature-set mpls** および **feature-set mpls** コマンドを使用してインストールし、有効にする必要があります。

feature tunnel コマンドを使用して、トンネリング機能を有効にする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	グローバル設定モードを開始します。

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	
ステップ 2	[**no**] **feature segment-routing** 例： `switch(config)# feature segment-routing`	MPLS セグメント ルーティング機能を有効化します。このコマンドの **no** 形式は、MPLS セグメント ルーティング機能を無効化します。
ステップ 3	（任意） **show running-config | inc 'feature segment-routing'** 例： `switch(config)# show running-config | inc 'feature segment-routing'`	MPLS セグメント ルーティング機能のステータスを表示します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします

ステップ 5	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 6	**feature tunnel**
	例：
	`switch(config)# feature tunnel`
	`switch(config-if)#`
		新しいトンネルインターフェイスを作成できます。
		トンネルインターフェイス機能を無効にするには、このコマンドの **no** 形式を使用します。

ステップ 7	switch(config)# **interface tunnel** *number*	トンネル インターフェイス コンフィギュレーション モードを開始します。
ステップ 8	switch(config-if)# **tunnel mode** {**gre ip** }	このトンネル モードを GRE に設定します。 IP での GRE カプセル化の使用を指定するには、**gre** キーワードおよび **ip** キーワードを指定します。
ステップ 9	**tunnel source** {*ip-address* | *interface-name*} 例： `switch(config-if)# tunnel source ethernet 1/2`	この IP トンネルの送信元アドレスを設定します。送信元は、IP アドレスまたは論理インターフェイス名によって指定できます。

	コマンドまたはアクション	目的
ステップ 10	**tunnel destination** ip{address | hostname} 例： `switch(config-if)# tunnel destination 192.0.2.1`	このIPトンネルの宛先アドレスを設定します。宛先は、IPアドレスまたは論理ホスト名によって指定できます。
ステップ 11	**tunnel use-vrf** *vrf-name* 例： `switch(config-if)# tunnel use-vrf blue`	
ステップ 12	**ipv6 address** *IPv6* アドレス	`switch(config-if)# 10.1.1.1` IPv6 アドレス を設定します。 （注） トンネルの送信元アドレスと宛先アドレスは同じままです（IPv4アドレス）。
ステップ 13	（任意） switch(config-if)# **show interface tunnel** *number*	トンネルインターフェイスの統計情報を表示します。
ステップ 14	switch(config-if)# **mtu** *value*	インターフェイスで送信される IP パケットの Maximum Transmission Unit（MTU;最大伝送単位）を設定します。
ステップ 15	（任意） switch(config-if)# **copy running-config startup-config**	リブートおよびリスタート時に実行コンフィギュレーションをスタートアップコンフィギュレーションにコピーして、変更を継続的に保存します。

セグメント ルーティング MPLS および GRE の設定の確認

スタティック ルーティング MPLS および GRE の設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show segment-routing mpls	セグメント ルーティング MPLS 情報を表示します

レイヤ 3 EVPN の SR-TE の確認

ODN の検証は、L3VPN VRF プレフィックスに基づいています。

1. R1（ヘッドエンドと PCE サーバー）間の PCEP セッションが確立されていることを確認します。

```
R1# show srte pce ipv4 peer

PCC's peer database:
--------------------
Remote PCEP conn IPv4 addr: 58.8.8.8
Local PCEP conn IPv4 addr: 51.1.1.1
Precedence: 0
State: up
```

2. 次のコマンドを使用して、R1、R3、および R6 の BGP LS および BGP EVPN セッションを確認します。
   - Show bgp l2vpn evpn summary
   - Show bgp link-state summary

3. R1（ヘッドエンド）に、R6 ループバック アドレスへの可視性がないことを確認します。

```
R1# show ip route 56.6.6.6
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

56.6.6.6/32, ubest/mbest: 1/0
    *via Null0, [1/0], 1d02h, static
```

4. VRF プレフィックスが MP-BGP によって R1 VRF SR ルーティング テーブルにインジェクトされることを確認します。

```
R1# show ip route vrf sr
106.107.4.1/32, ubest/mbest: 1/0
    *via binding label 100534%default, [20/0], 1d01h, bgp-6503, external, tag 6500
(mpls-vpn)
```

5. SR-TE トンネルを確認します。

```
R1# show srte policy
Policy name: 51.1.1.1|1001
    Source: 51.1.1.1
    End-point: 56.6.6.6
    Created by: bgp
    State: UP
    Color: 1001
    Insert: FALSE
    Re-opt timer: 0
    Binding-sid Label: 100534
    Policy-Id: 2
    Flags:
    Path type = MPLS           Path options count: 1
     Path-option Preference:100 ECMP path count: 1
      1.      PCE         Weighted: No
        Delegated PCE: 58.8.8.8
                Index: 1                Label: 101104
                Index: 2                Label: 201102
                Index: 3                Label: 201103
```

セグメント ルーティングの設定の確認

スタティック ルーティングの設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show bgp ipv4 labeled-unicast *prefix*	指定された IPv4 プレフィックスのアドバタイズされたラベル インデックスおよび選択されたローカル ラベルを表示します。
show bgp paths	アドバタイズされたラベル インデックスを含む BGP パス情報を表示します。
show mpls label range	構成されたラベルの SRGB 範囲を表示します。
show route-map [*map-name*]	ラベル インデックスなど、ルート マップに関する情報を表示します。
show running-config rpm	ルート ポリシー マネージャ（RPM）についての情報を表示します。
show running-config | inc 'feature segment-routing'	MPLS セグメント ルーティング機能のステータスを表示します。
show ip ospf neighbors detail	OSPFv2 ネイバー、および割り当てられた隣接関係 SID のリストを、対応するフラグとともに表示します。
show ip ospf database opaque-area	隣接 SID の LSA を表示します。
show ip ospf segment-routing adj-sid-database	ローカルに割り当てられた隣接 SID をすべて表示します。
show running-config segment-routing	セグメント ルーティング機能のステータスを表示します。
show srte policy	許可されたポリシーのみを表示します。
show srte policy [**all**]	SR-TE で使用可能なすべてのポリシーのリストを表示します。
show srte policy [**detail**]	要求されたすべてのポリシーの詳細ビューを表示します。

コマンド	目的
show srte policy <*name*>	SR-TE ポリシーを名前でフィルタリングし、SR-TE でその名前で使用できるすべてのポリシーのリストを表示します。 （注） このコマンドには、ポリシー名のオートコンプリート機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。
show srte policy color <*color*> **endpoint** <*endpoint*>	カラーとエンドポイントの SR-TE ポリシーを表示します。 （注） このコマンドには、カラーとエンドポイントのオートコンプリート機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。
show srte policy fh	最初のホップのセットを表示します。
show segment-routing mpls clients	SR-APP に登録されているクライアントを表示します。
show segment-routing mpls details	詳細情報を表示します。
show segment-routing ipv4	IPv4 アドレス ファミリの BGP 情報を表示します。
show segment-routing mpls	セグメント ルーティング MPLS 情報を表示します
show segment-routing ipv4 connected-prefix-sid	SRGB の MPLS ラベル範囲を表示します。 （注） このコマンドは、Cisco NX-OS リリース 9.3(1) でのみ使用できます。
show ip ospf プロセス	OSPF モードを表示します。
show ip ospf プロセス **segment-routing sid-database**	セグメントルーティング データベースの詳細を表示します。
show ip ospf プロセス **segment-routing global block**	セグメントルーティング グローバルブロック情報を表示します。
show nve evi	EVI のステータスを表示します。

コマンド	目的
show nve peer mpls	セグメント ルーティング ピアのステータスを表示します。
show nve adjacency mpls	ピア隣接のステータスを表示します。

SRTE 明示パス エンドポイント置換の構成

この章には、SRTE 明示パス エンドポイント置換機能を構成する方法に関する情報が含まれています。

SRTE 明示パス エンドポイント置換

SRTE 明示パス エンドポイント置換機能を使用すると、ユーザーは明示パスを一連の MPLS ラベル（通常の明示パスと同様）として定義できますが、ポリシー エンドポイント ラベルを表す一連のプレースホルダーを追加できます。プレースホルダーは、**policy-endpoint** キーワードで表されます。ポリシーエンドポイントプレースホルダーが表示されるパス内の位置は、SRTE によって、ポリシーのエンドポイント IP アドレスのノード SID を表すセグメント ルーティング ラベルに内部的に解決されます。

これは、定義する必要があるポリシーの総数を減らすため、オンデマンドのカラーテンプレートと組み合わせて使用すると役立ちます。カラーとエンドポイントの組み合わせごとに個別のパスを定義する代わりに、ユーザーは、その色のすべてのエンドポイントのポリシーを定義するためのエンドポイント置換を含む明示的なパスを含むオンデマンド カラー テンプレートを定義できます。

SRTE 明示パス エンドポイント置換の注意事項と制限事項

SRTE 明示パス エンドポイントの置換には、次の注意事項と制限事項があります。

- Cisco NX-OS Release 10.1(1) 以降、SRTE 明示パス エンドポイント置換は、Cisco Nexus 9300-FX、9300-FX2、9300-FX3、および 9300-GX プラットフォーム スイッチでサポートされています。
- 部分パスが解決されたエンドポイント ラベルと同じラベルで終わる場合、余分な（重複した）トランスポート ラベルを追加しないでください。
- SRGB はすべてのノードで同じでなければなりません。そうでない場合、各中間ノードのセグメント構成によっては、機能が動作しない場合があります。
- セグメント リストには、ポリシー エンドポイント エントリを 1 つだけ含めることができます。

SRTE 明示的パス エンドポイント置換の構成

エンドポイント置換を使用するポリシーを作成するには、最初にセグメント リスト モードを使用してパスを定義します。次に、その名前を使用してパスをオンデマンドの色に関連付けます。

始める前に

MPLS セグメント ルーティング トラフィック エンジニアリング機能が有効になっていることを確認する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**segment-routing**
	例：
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		セグメントルーティング構成モードを開始します。

ステップ 3	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィック エンジニアリング モードに入ります。

ステップ 4	**segment-list name** *path*
	例：
	`switch(config-sr-te)# segment-list name path`
	`switch(config-sr-te-exp-seg-list)#`
		明示的なセグメントリストを構成します。

ステップ 5	**index 1 mpls label** *label-ID*
	例：
	`switch(config-sr-te-exp-seg-list)# index 1 mpls label 16201`
	`switch(config-sr-te-exp-seg-list)#`
		セグメント リストに MPLS ラベルを構成します。

ステップ 6	**index 2 policy-endpoint**
	例：
	`switch(config-sr-te-exp-seg-list)# index 2 policy-endpoint`
	`switch(config-sr-te-exp-seg-list)#`
		ポリシーのエンドポイント解決を構成します。

	コマンドまたはアクション	目的

ステップ 7	**exit**
	例：
	`switch(config-sr-te-exp-seg-list)# exit`
	`switch(config-sr-te)#`
		セグメント リスト モードを終了し、SRTE モードに戻ります。

ステップ 8	**on-demand color** *color_num*
	例：
	`switch(config-sr-te)# on-demand color 201`
	`switch(config-sr-te-color)#`
		オンデマンド色テンプレート モードを開始し、特定の色のオンデマンド色を構成します。

ステップ 9	**candidate-paths** 例： `switch(config-sr-te-color)# candidate-paths`	SR-TE カラー ポリシーの候補パスを指定します。
ステップ 10	**preference** *preference-number* 例： `switch(cfg-cndpath)# preference 100`	候補パスの優先順位を指定します。
ステップ 11	**explicit segment-list** *path* 例： `switch(cfg-pref)# explicit segment-list path`	明示セグメント リストを指定します。

SRTE 明示パス エンドポイントの置換構成例

この例は、SRTE 明示パス エンドポイントの置換構成を示しています。

```
switch(config)# segment-routing
switch(config-sr)# traffic-engineering
switch(config-sr-te)# segment-list name path
switch(config-sr-te-exp-seg-list)# index 1 mpls label 16201
switch(config-sr-te-exp-seg-list)# index 2 policy-endpoint
switch(config-sr-te-exp-seg-list)# exit
switch(config-sr-te)# on-demand color 201
switch(config-sr-te-color)# candidate-paths
switch(cfg-cndpath)# preference 100
switch(cfg-pref)# explicit segment-list path
```

SRTE 明示パス エンドポイント置換の構成の確認

SRTE 明示パス エンドポイント置換構成に関する必要な詳細を表示するには、次のいずれかのタスクを実行します。

表 *3 : SRTE* 明示パス エンドポイントの置換構成の確認

コマンド	目的
show srte policy	許可されたポリシーのみを表示します。 （注） エンドポイント ラベルが解決され、最初のホップに到達できる場合、状態は UP と表示されます。エンドポイントラベルが解決されていない場合、または最初のホップに到達できない場合、状態はDOWN と表示されます。
show srte policy [**all**]	SR-TE で使用可能なすべてのポリシーのリストを表示します。 （注） エンドポイント ラベルが解決され、最初のホップに到達できる場合、状態は UP と表示されます。エンドポイントラベルが解決されていない場合、または最初のホップに到達できない場合、状態はDOWN と表示されます。
show srte policy [**detail**]	要求されたすべてのポリシーの詳細ビューを表示します。 （注） エンドポイント ラベルが解決され、最初のホップに到達できる場合、状態は UP と表示されます。エンドポイントラベルが解決されていない場合、または最初のホップに到達できない場合、状態はDOWN と表示されます。
show srte policy <*name*>	SR-TE ポリシーを名前でフィルタリングし、SR-TE でその名前で使用できるすべてのポリシーのリストを表示します。 （注） このコマンドには、ポリシー名のオートコンプリート機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。

コマンド	目的
show srte policy color <*color*> **endpoint** <*endpoint*>	カラーとエンドポイントの SR-TE ポリシーを表示します。 （注） このコマンドには、カラーとエンドポイントのオートコンプリート機能があります。この機能を使用するには、疑問符を追加するか、TAB キーを押します。
show srte policy fh	既存の最初のホップとポリシー エンドポイントの状態を表示します。

デフォルト VRF を介した SRTE の構成

デフォルト VRF を介した SRTE について

デフォルト VRF を介した SRTE 機能を使用すると、セグメント ルーティング トラフィック エンジニアリングを組み込んで、ネットワークでトラフィック ステアリングの利点を実現できます。SRTE は、大規模なデータセンター（DC）でのルーティングに BGP を使用しながら、スケーラビリティを向上させます。

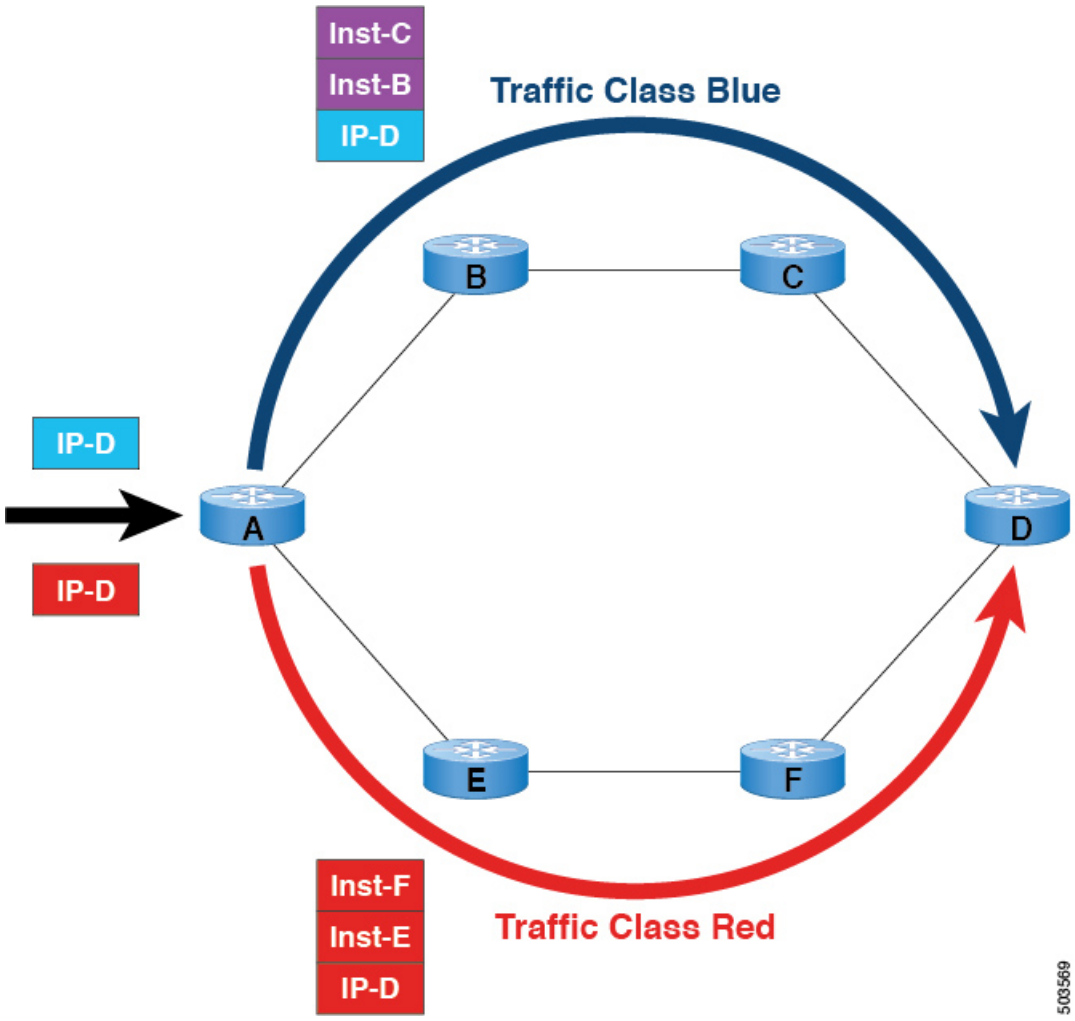
デフォルト VRF を介した SRTE 機能は、拡張コミュニティ属性として存在し、トラフィック ステアリングのベースとして番号で表されるルート カラーを使用します。カラーに基づいてプレーン分離が実現され、トラフィックを伝送するための SR ポリシーが作成されます。さらにカラーに基づいて、DCはさまざまなプレーンに分割されます。アプリケーションは、各プレーンを使用して特定のプレーンのみをルーティングし、トラフィックを適切な宛先に誘導するように構成されています。

平面分離には次の利点があります。

- 1 つのフローが他のフローに影響を与えることはありません。
- 大小のフローは、異なる平面に分離されます。
- デバッグを容易にするための障害分離：1 つのプレーンの障害が他のプレーンに影響を与えることはありません。たとえば、1 つのプレーンでネットワーク障害が発生した場合、そのプレーンのアプリケーションのみが影響を受けますが、残りのプレーンのアプリケーションは影響を受けません。さらに、障害を分離し、分離してトラブルシューティングを行うことができます。

次の例では、図を使用してデフォルト VRF を介した SRTE 機能を説明しています。

図 *6* : デフォルト *VRF* を介した *SRTE* の例



- BGP の場合、ノード A は入力ルータであり、ノード D は出力ルータです。D はネクストホップでもあります。
- SRTE の場合、ノード A は SRTE ヘッドエンドであり、ノード D はポリシーのエンドポイントです。
- ルート プレフィックス 1 はブルー プレーンを使用するように構成され、ルート 2 はレッド プレーンを使用するように構成されています。

青のトラフィックには、ノード B とノード C を介してトラフィックを誘導する命令が追加され、赤のトラフィックには、ノード E とノード F を経由してトラフィックを誘導する命令が追加されます。要約すると、トラフィックはアドバタイズメントのカラーに基づいて処理されます。これは、以前にアドバタイズされたプレフィックスです。

デフォルト VRF 経由の SRTE を構成する場合の注意事項と制限事項

- Cisco NX-OS リリース 10.1(1) 以降、セグメント ルーティング トラフィック エンジニアリングは、Cisco Nexus 9300-FX3、N9K-C9316D-GX、N9K-C93180YC-FX、N9K-C93240YC-FX2、および N9K-C9364C プラットフォーム スイッチのデフォルト VRF でサポートされます。この SR-TE 機能の制限は次のとおりです。
  - アンダーレイ IPv6 はサポートされません。SRv6 は代替です。
  - BGP の専用ファブリックにおける PCE の欠点のため、BGP アンダーレイを使用した PCE はサポートされていません。
  - NXOS が BGP-LS で LSA をアドバタイズできないため、PCE を使用した OSPF-SRTE はサポートされていません。
  - 合計 1000 の SRTE ポリシー スケール、130K v4 の BGP デフォルト VRF（v4）、および 1000 のアンダーレイ SR プレフィックスをサポートします。
- Cisco NX-OS リリース 10.2(3)F 以降、カラー専用（CO）ビットのオプションがルートマップに追加されています。SRTE ポリシーを使用している特定のプレフィックスの CO ビットの値が変更された場合、BGP は古いポリシーを削除し、新しいポリシーを追加します。この機能は、Cisco Nexus 9300-EX、9300-FX、9300-FX2、9300-GX、および 9300-GX2 プラットフォーム スイッチでサポートされます。

構成プロセス：デフォルト VRF を介した SRTE

構成プロセスは次のとおりです。

1. ネクストホップを変更しない: ネクストホップは、入力ノードで SR ポリシーを計算するために使用されます。プレフィックスがアップストリームにアドバタイズされるため、プレフィックスの SR ドメインのネクストホップを保持する必要があります。したがって、ホップバイホップの ebgp の場合、すべての上流ルータでネクストホップが変更されていない必要があります。
2. 出力ノード、入力ノード、ネットワーク/再配布、またはデフォルト発信元で拡張コミュニティ カラーを設定します。
3. 入力ノードは、カラー拡張されたコミュニティを受信すると、それを SR ポリシーに一致させます。
4. SR ポリシーのエンドポイントは、カラー拡張コミュニティのプレフィックスとカラーのネクストホップから派生します。

このセクションには、デフォルト VRF での SRTE の構成に関する次のトピックが含まれています。

ネクストホップ変更なしの構成

デフォルト VRF オーバーレイの中間（スパイン）ノードでネクスト ホップを変更せずに構成し、ネクスト ホップが変更されないようにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**route-map** *map-name*
	例：
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ コンフィギュレーション モードを開始します。

ステップ 3	[**no**] **set ip next-hop unchanged**
	例：
	`switch(config-route-map)# set ip next-hop unchanged`
	`switch(config-route-map)#`
		ネクストホップを変更せずに設定します。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ設定モードを終了します。

ステップ 5	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
		BGP プロセスおよび関連する設定を削除するには、このコマンドで no オプションを使用します。

ステップ 6	**neighbor** *ip-address*
	例：
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		BGP ネイバー テーブルまたはマルチプロトコル BGP ネイバー テーブルにエントリを追加します。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。

	コマンドまたはアクション	目的

ステップ 7	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IPv4 アドレス ファミリ タイプのルータのアドレスファミリ構成モードを開始します。

ステップ 8	**route-map** *map-name* **out**
	例：
	`switch(config-router-neighbor-af)# route-map ABC out`
	`switch(config-router-neighbor-af)#`
		発信ルートに設定された BGP ポリシーを適用します。

拡張コミュニティ カラーの構成

このセクションは、次のトピックで構成されています。

出力ノードでの拡張コミュニティ カラーの構成

プレフィックスが出力ノードによって通知されるときに、出力ノードで拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**route-map** *map-name*
	例：
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ コンフィギュレーション モードを開始します。

ステップ 3	**set extcommunity color** *color-num* [**co-flag** *co-flag*]
	例：
	`switch(config-route-map)# set extcommunity color 20 [co-flag 00]`
	`switch(config-route-map)#`
		カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。
		co-flag：カラー専用フラグを使用して、正確なカラーとエンドポイントのポリシーが見つからない場合に、カラーのみに基づいてトラフィックを SR ポリシーに誘導できるかどうかを制御します。デフォルトは 00 です。
		（注）

	コマンドまたはアクション	目的
		co-flag 00 を選択して、カラーとネクストホップに基づきデフォルトの自動ステアリングを指定します。co-flag が 00 もしくはデフォルトに設定されている場合、リクエストされたカラーとエンドポイントを持つポリシーのバインド SID がルーティングに使用されます。 co-flag 01 を選択し、カラーにのみ基づいてトラフィックを誘導します。co-flag が 01 に設定され、リクエストされたカラーとエンドポイントを持つポリシーが存在する場合、ポリシーのバインド SID がルーティングに使用されます。ポリシーが存在しないが、同じカラーを持つ null エンドポイント ポリシーが存在する場合、null エンドポイント ポリシーのバインド SID がルーティングに使用されます。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ設定モードを終了します。

ステップ 5	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
		BGP プロセスおよび関連する設定を削除するには、このコマンドで no オプションを使用します。

ステップ 6	**neighbor** *ip-address*
	例：
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		BGP ネイバー テーブルまたはマルチプロトコル BGP ネイバー テーブルにエントリを追加します。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。

	コマンドまたはアクション	目的

ステップ 7	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IPv4 アドレス ファミリ タイプのルータのアドレスファミリ構成モードを開始します。

ステップ 8	**route-map** *map-name* **out**
	例：
	`switch(config-router-neighbor-af)# route-map ABC out`
	`switch(config-router-neighbor-af)#`
		発信ルートに設定された BGP ポリシーを適用します。
		マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

入力ノードにおける拡張コミュニティ カラーの構成

SRTE ポリシーがインスタンス化される入力ノードによってプレフィックスが通知されるときに、入力ノードで拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**route-map** *map-name*
	例：
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ コンフィギュレーション モードを開始します。

ステップ 3	**set extcommunity color** *color-num* [**co-flag** *co-flag*]
	例：
	`switch(config-route-map)# set extcommunity color 20 [co-flag 00]`
	`switch(config-route-map)#`
		カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。
		co-flag：カラー専用フラグを使用して、正確なカラーとエンドポイントのポリシーが見つからない場合に、カラーのみに基づいてトラフィックを SR ポリシーに誘導できるかどうかを制御します。デフォルトは 00 です。
		（注）
		co-flag 00 を選択して、カラーとネクストホップに基づきデフォルトの自動ステアリングを指定します。co-flag が 00 もしくはデフォルトに設定されている

	コマンドまたはアクション	目的
		場合、リクエストされたカラーとエンドポイントを持つポリシーのバインド SID がルーティングに使用されます。 co-flag 01 を選択し、カラーにのみ基づいてトラフィックを誘導します。co-flag が 01 に設定され、リクエストされたカラーとエンドポイントを持つポリシーが存在する場合、ポリシーのバインド SID がルーティングに使用されます。ポリシーが存在しないが、同じカラーを持つ null エンドポイント ポリシーが存在する場合、null エンドポイント ポリシーのバインド SID がルーティングに使用されます。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ設定モードを終了します。

ステップ 5	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
		BGP プロセスおよび関連する設定を削除するには、このコマンドで no オプションを使用します。

ステップ 6	**neighbor** *ip-address*
	例：
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		BGP ネイバー テーブルまたはマルチプロトコル BGP ネイバー テーブルにエントリを追加します。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。

ステップ 7	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IPv4 アドレス ファミリ タイプのルータのアドレスファミリ構成モードを開始します。

ステップ 8	**route-map** *map-name* **in** 例：	構成された BGP ポリシーを受信ルートに適用します。

	コマンドまたはアクション	目的
	`switch(config-router-neighbor-af)# route-map ABC in` `switch(config-router-neighbor-af)#`	マップ-名には最大63文字の英数字を使用できます。大文字と小文字は区別されます。

出力ノードでのネットワーク/再配布コマンドの拡張コミュニティカラー構成

プレフィックスが出力ノードによって通知されるときに、出力ノードで network/redistribute コマンドの拡張コミュニティ カラーを構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**route-map** *map-name*
	例：
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ コンフィギュレーション モードを開始します。

ステップ 3	**set extcommunity color** *color-num* [**co-flag** *co-flag*]
	例：
	`switch(config-route-map)# set extcommunity color 20 [co-flag 00]`
	`switch(config-route-map)#`
		カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。
		co-flag：カラー専用フラグを使用して、正確なカラーとエンドポイントのポリシーが見つからない場合に、カラーのみに基づいてトラフィックを SR ポリシーに誘導できるかどうかを制御します。デフォルトは 00 です。
		（注）
		co-flag 00 を選択して、カラーとネクストホップに基づきデフォルトの自動ステアリングを指定します。co-flag が 00 もしくはデフォルトに設定されている場合、リクエストされたカラーとエンドポイントを持つポリシーのバインド SID がルーティングに使用されます。
		co-flag 01 を選択し、カラーにのみ基づいてトラフィックを誘導します。co-flag が 01 に設定され、リクエストされたカラーとエンドポイントを持つポリシー

	コマンドまたはアクション	目的
		が存在する場合、ポリシーのバインド SID がルーティングに使用されます。ポリシーが存在しないが、同じカラーを持つ null エンドポイント ポリシーが存在する場合、null エンドポイント ポリシーのバインド SID がルーティングに使用されます。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ設定モードを終了します。

ステップ 5	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は16ビット整数または32ビット整数にできます。上位16ビット10進数と下位 16 ビット 10 進数による xx.xx という形式です。
		BGP プロセスおよび関連する設定を削除するには、このコマンドで no オプションを使用します。

ステップ 6	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		VRF インスタンスの IPv4 アドレス ファミリを指定し、アドレス ファミリ構成モードを開始します。

ステップ 7	**redistribute static route-map** *map-name* **out**
	例：
	`switch(config-router-af)# redistribute static route-map ABC`
	`switch(config-router-af)#`
		スタティック ルートを BGP に再配布します。マップ-名には最大63文字の英数字を使用できます。大文字と小文字は区別されます。

ステップ 8	**network** *ip-prefix* [**route-map** *map-name*]
	例：
	`switch(config-router-af)# network 1.1.1.1/32 route-map ABC`
	`switch(config-router-af-network)#`
		ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。

出力ノードで Default-Originate の拡張コミュニティ カラーの構成

デフォルトのプレフィックスが出力ノードによって通知されたときに、出力ノードで default-originate の拡張コミュニティ カラー構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**route-map** *map-name*
	例：
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		ルート マップを作成するか、または既存のルート マップに対応するルート マップ コンフィギュレーション モードを開始します。
		マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

ステップ 3	**set extcommunity color** *color-num* [**co-flag** *co-flag*] 例： `switch(config-route-map)# set extcommunity color 20 [co-flag 00]`	カラー拡張コミュニティの BGP 外部コミュニティ属性を設定します。 **co-flag**：カラー専用フラグを使用して、正確なカラーとエンドポイントのポリシーが見つからない場合に、カラーのみに基づいてトラフィックを SR ポリシーに誘導できるかどうかを制御します。デフォルトは 00 です。 （注） co-flag 00 を選択して、カラーとネクストホップに基づきデフォルトの自動ステアリングを指定します。co-flag が 00 もしくはデフォルトに設定されている場合、リクエストされたカラーとエンドポイントを持つポリシーのバインド SID がルーティングに使用されます。 co-flag 01 を選択し、カラーにのみ基づいてトラフィックを誘導します。co-flag が 01 に設定され、リクエストされたカラーとエンドポイントを持つポリシーが存在する場合、ポリシーのバインド SID がルーティングに使用されます。ポリシーが存在しないが、同じカラー

	コマンドまたはアクション	目的
		を持つ null エンドポイント ポリシーが存在する場合、null エンドポイント ポリシーのバインド SID がルーティングに使用されます。

ステップ 4	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルートマップ設定モードを終了します。

ステップ 5	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp1`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
		BGP プロセスおよび関連する設定を削除するには、このコマンドで no オプションを使用します。

ステップ 6	**neighbor** *ip-address*
	例：
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		BGP ネイバー テーブルまたはマルチプロトコル BGP ネイバー テーブルにエントリを追加します。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。

ステップ 7	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IPv4 アドレス ファミリ タイプのルータのアドレスファミリ構成モードを開始します。

ステップ 8	**default-originate** [**route-map** *map-name*]
	例：
	`switch(config-router-neighbor-af)# default-originate route-map ABC`
	`switch(config-router-neighbor-af)#`
		BGP ピアへのデフォルト ルートを作成します。
		マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

入力ピアの BGP の構成（SRTE ヘッドエンド）

入力ピア（SRTE ヘッドエンド）の BGP を構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

ステップ 2	[**no**] **feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)`
		BGP を有効にします。
		この no コマンド形式を使用して、この機能を無効にします。

ステップ 3	[**no**] **router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
		BGP プロセスおよび関連する設定を削除するには、このコマンドで no オプションを使用します。

ステップ 4	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		IPv4 アドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 5	**neighbor** *ip-address*
	例：
	`switch(config-router-af)# neighbor 209.165.201.1`
	`switch(config-router-af-neighbor)#`
		リモート BGP ピアの IPv4 アドレスを設定します。ip-address の形式は x.x.x.x です。

ステップ 6	**remote-as** *as-number* 例： `switch(config-router-af-neighbor)# remote-as 64497`	リモート BGP ピアの AS 番号を設定します。
ステップ 7	**update-source** *interface number* 例： `switch(config-router-af-neighbor)# update-source loopback 300`	BGP セッションの送信元を指定し、更新します。

	コマンドまたはアクション	目的
ステップ 8	**ebgp-multihop** *ttl-value* 例： `switch(config-router-af-neighbor)# ebgp-multihop 5`	eBGP マルチホップの eBGP TTL を設定します。有効な範囲は 2 ～ 255 です。このコマンドの使用後、BGP セッションを手動でリセットする必要があります。
ステップ 9	**exit** 例： `switch(config-router-af-neighbor)# exit`	ネイバー コンフィギュレーション モードを終了します。

ステップ 10	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		IPv4 アドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 11	**route-map** *map-name* **in** 例： `switch(config-router-af)# route-map color 401 in`	SRTE 入力ピアのルート マップを指定します。 マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。 （注） NLRI に適用できる拡張コミュニティ カラーは 1 つのみなので、適用されたルート ポリシー/ルート マップは、以前の拡張コミュニティ カラーが存在する場合は上書きします。

入力ピアの BGP 構成（SRTE エンドポイント）

出力ピア（SRTE エンドポイント）の BGP を構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定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 2	**[no] feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)`
		BGP を有効にします。
		この no コマンド形式を使用して、この機能を無効にします。

ステップ 3	**[no] router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
		BGP プロセスおよび関連する設定を削除するには、このコマンドで no オプションを使用します。

ステップ 4	**neighbor** *ip-address*
	例：
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの IPv4 アドレスを設定します。ip-address の形式は x.x.x.x です。

ステップ 5	**remote-as** *as-number* 例： `switch(config-router-neighbor)# remote-as 64497`	リモート BGP ピアの AS 番号を設定します。
ステップ 6	**update-source** *interface-number* 例： `switch(config-router-neighbor)# update-source loopback 300`	BGP セッションの送信元を指定し、更新します。
ステップ 7	**ebgp-multihop** *ttl-value* 例： `switch(config-router-neighbor)# ebgp-multihop 5`	eBGP マルチホップの eBGP TTL を設定します。有効な範囲は 2 ～ 255 です。このコマンドの使用後、BGP セッションを手動でリセットする必要があります。
ステップ 8	**exit** 例： `switch(config-router-af-neighbor)# exit`	ネイバー コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的

ステップ 9	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		IPv4 アドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 10	**send-community**
	例：
	`switch(config-router-af)# send-community`
	`switch(config-router-af)#`
		BGP コミュニティ属性を BGP ネイバーに送信する必要があることを指定します。

ステップ 11	**send-community extended**
	例：
	`switch(config-router-af)#send-community extended`
	`switch(config-router-af)#`
		拡張コミュニティ属性が BGP ネイバーに送信されるように指定します。

ステップ 12	**route-map** *map-name* **out**
	例：
	`switch(config-router-af)# route-map color 301 out`
	`switch(config-router-af)#`
		SRTE 出力ピアのルート マップを指定します。
		マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。
		（注）
		NLRI に適用できる拡張コミュニティ カラーは 1 つのみなので、適用されたルート ポリシー/ルート マップは、以前の拡張コミュニティ カラーが存在する場合は上書きします。

入力ピア用 SRTE の構成

入力ピア（SRTE ヘッドエンド）の SRTE を構成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

ステップ 2	[**no**] **feature mpls segment-routing traffic-engineering**	MPLS SRTE を有効にします。

	コマンドまたはアクション	目的

	例：
	`switch(config)# feature mpls segment-routing traffic-engineering`
	`switch(config)`
		この no コマンド形式を使用して、この機能を無効にします。

ステップ 3	**segment-routing**
	例：
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		セグメントルーティング構成モードを開始します。

ステップ 4	**traffic-engineering**
	例：
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		トラフィック エンジニアリング モードに入ります。

ステップ 5	**segment-list name** *path*
	例：
	`switch(config-sr-te)# segment-list name path`
	`switch(config-sr-te-exp-seg-list)#`
		明示的なセグメントリストを構成します。

ステップ 6	**index 1 mpls label** *label-ID*
	例：
	`switch(config-sr-te-exp-seg-list)# index 1 mpls label 16601`
	`switch(config-sr-te-exp-seg-list)#`
		セグメント リストに MPLS ラベルを作成します。

ステップ 7	**index 2 mpls label** *label-ID*
	例：
	`switch(config-sr-te-exp-seg-list)# index 2 mpls label 16501`
	`switch(config-sr-te-exp-seg-list)#`
		セグメント リストに MPLS ラベルを作成します。

ステップ 8	**policy** *policy-name*-**bgp**
	例：
	`switch(config-sr-te-exp-seg-list)# policy dci1-edge1-bgp`
	`switch(config-sr-te-exp-seg-list)#`
		SRTE ポリシー名を指定します。

ステップ 9	**color** *color-num* **endpoint** *endpoint ID* 例： `switch(config-sr-te)# color 13401 endpoint 1.0.3.1`	ポリシーのカラーとエンドポイントを指定します（SRTE 出力ノード ループバック）。
ステップ 10	**candidate-paths** 例： `switch(config-sr-te-color)# candidate-paths`	SRTE カラー ポリシーの候補パスを指定します。

	コマンドまたはアクション	目的
ステップ 11	**preference** *preference-number* 例： `switch(cfg-cndpath)# preference 100`	候補パスの優先順位を指定します。
ステップ 12	**explicit segment-list** *path* 例： `switch(cfg-pref)# explicit segment-list path`	明示セグメントリストを指定します。

デフォルト VRF 経由の SRTE 構成例

次の例は、デフォルトの VRF 構成を介した SRTE を示しています。

構成例：ネクストホップ変更なし

```
route-map ABC
  set ip next-hop unchanged

router bgp 1
  neighbor 1.2.3.4
    address-family ipv4 unicast
       route-map ABC out
```

構成例：拡張コミュニティ カラー

このセクションには、拡張コミュニティ カラーの次の構成例が含まれます。

構成例：出力ノード

```
ip prefix-list pfx1 seq 5 permit 7.7.7.7/32
ip prefix-list pfx2 seq 5 permit 5.0.0.0/24
route-map ABC
  match ip address prefix-list pfx1 pfx2
    set extcommunity color 20

router bgp 1
  neighbor 1.2.3.4
    address-family ipv4 unicast
       route-map ABC out
```

入力ノードの構成例

```
ip prefix-list pfx1 seq 5 permit 7.7.7.7/32
ip prefix-list pfx2 seq 5 permit 5.0.0.0/24
route-map ABC
  match ip address prefix-list pfx1 pfx2
    set extcommunity color 20

router bgp 1
  neighbor 1.2.3.4
    address-family ipv4 unicast
       route-map ABC in
```

出力ノードでネットワーク/再配布コマンドの構成例

```
route-map ABC
  set extcommunity color 20

router bgp 1
  address-family ipv4 unicast
    redistribute static route-map ABC
    network 1.1.1.1/32 route-map ABC
```

構成例：出力ノードでデフォルトの生成をする場合

```
route-map ABC
  set extcommunity color 20

router bgp 1
    neighbor 1.2.3.4
       address-family ipv4 unicast
           default-originate route-map ABC
```

構成例：入力ピアの BGP（SRTE ヘッドエンド）

```
DCI-1(config)# show running-config bgp
feature bgp
router bgp 100
  address-family ipv4 unicast
  neighbor 1.0.3.1
    remote-as 101
    update-source loopback0
    ebgp-multihop 255
    address-family ipv4 unicast
       route-map color-3401 in
```

構成例：出力ピアの BGP（SRTE エンドポイント）

この例は、SRTE 明示パス エンドポイントの置換構成を示しています。

```
Edge-1(config)# show running-config bgp
feature bgp
router bgp 101
neighbor 1.0.1.1
    remote-as 100
    update-source loopback0
    ebgp-multihop 255
    address-family ipv4 unicast
      send-community
      send-community extended
      route-map color-3401 out
```

構成例：SRTE の入力ピア（SRTE ヘッドエンド）

```
DCI-1# show running-config srte
feature mpls segment-routing traffic-engineering
segment-routing
  traffic-engineering
    segment-list name dci1-edge1
      index 1 mpls label 16601
      index 2 mpls label 16501
    policy dci1-edge1-bgp
      color 13401 endpoint 1.0.3.1
      candidate-paths
```

```
preference 30
  explicit segment-list dci1-edge1
```

デフォルト VRF を介した SRTE 構成の確認

デフォルトの VRF 構成を介した SRTE に関する適切な詳細を表示するには、次のいずれかのタスクを実行します。

表 *4 :* デフォルト *VRF* 構成を介した *SRTE* の確認

コマンド	目的
show running-config bgp	入力ピアまたは SRTE ヘッドエンドに関する情報を表示します。
show running-config bgp	出力ピアまたは SRTE エンドポイントに関する情報を表示します。
show running-config srte	入力ピアの SRTE ポリシーに関する情報を表示します。

その他の参考資料

関連資料

関連項目	マニュアル タイトル
BGP	*Cisco Nexus 9000* シリーズ ユニキャスト ルーティング設定ガイド

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。