BGP の設定

BGP

ボーダーゲートウェイ プロトコル（BGP）は、組織または自律システム間のループフリー ルーティングを実現する、インタードメイン ルーティング プロトコルです。Cisco NX-OS は BGP バージョン 4 をサポートします。BGP v4 に組み込まれているマルチプロトコル拡張機能を使用すると、IP マルチキャスト ルートおよび複数のレイヤ 3 プロトコル アドレス ファミリに関するルーティング情報を BGP に伝送させることができます。BGP では、他の BGP 対応デバイス（BGP ピア）との間で TCP セッションを確立するために、信頼できるトランスポート プロトコルとして TCP を使用します。外部組織に接続するときには、ルータが外部 BGP（eBGP）ピアリング セッションを作成します。ルーティング情報を交換するために同じ組織内の BGP ピアに接続するときは、ルータが内部 BGP（iBGP）ピアリング セッションを作成します。

BGP ではパスベクトル ルーティング アルゴリズムを使用して、BGP 対応ネットワーク デバイスまたは BGP スピーカーの間でルーティング情報を交換します。各 BGP スピーカはこの情報を使用して、宛先までのパスを判別し、なおかつルーティング ループを伴うパスを検出して回避します。ルーティング情報には、宛先のプレフィックス、宛先に対する自律システムのパス、およびその他のパス属性が含まれます。

BGP はデフォルトで、宛先ホストまたはネットワークへのベストパスとして、1 つだけパスを選択します。各パスは、BGP ベストパス分析で使用される well-known mandatory 属性、well-known discretionary 属性および optional transitive 属性を含みます。BGP ポリシーを構成し、これらの属性の一部を変更することによって、BGP パス選択を制御できます。詳細については、「ルート ポリシーおよび BGP セッションのリセット」を参照してください。BGP は、ロード バランシングまたは等コスト マルチパス（ECMP）もサポートします。詳細については、「ロード シェアリングとマルチパス」を参照してください。

Cisco NX-OS リリース 10.5(1)F から「基本 BGP を構成」の章と「高度な BGP を構成」の章は、まとめられて「BGP を構成」の章になりました。

BGP 自律システム

自律システム（AS）とは、単一の管理エンティティにより制御されるネットワークです。自律システムは 1 つまたは複数の IGP および整合性のある一連のルーティング ポリシーを使用して、ルーティング ドメインを形成します。BGP は 16 ビットおよび 32 ビットの自律システム番号をサポートします。詳細については、「自律システム」を参照してください。

個々の BGP 自律システムは外部 BGP（eBGP）ピアリング セッションを通じて、ルーティング情報をダイナミックに交換します。同じ自律システム内の BGP スピーカは、内部 BGP（iBGP）を通じて、ルーティング情報を交換できます。

4 バイトの AS 番号のサポート

BGP は、プレーン テキスト表記法または AS ドット付き表記法の 2 バイトの自律システム（AS）番号、もしくはプレーン テキスト表記法の 4 バイトの AS 番号をサポートします。

4 バイトの AS 番号を使用して BGP が設定されている場合は、**route-target auto** VXLAN コマンドを使用できません。これは、AS 番号とともに（すでに 3 バイト値である）VNI がルートターゲットの生成に使用されるためです。詳細については、『Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide』を参照してください。

アドミニストレーティブ ディスタンス

アドミニストレーティブディスタンスは、ルーティング情報源の信頼性を示す評価基準です。デフォルトで、BGP は表に示されたアドミニストレーティブ ディスタンスを使用します。

表 *1*: デフォルトの *BGP* アドミニストレーティブ ディスタンス

ディスタンス	デフォルト値	機能
外部	20	eBGP から学習したルートに適用されます。
内部	200	iBGP から学習したルートに適用されます。
ローカル	220	ルータを起点とするルートに適用されます。

（注） アドミニストレーティブ ディスタンスが BGP パス選択アルゴリズムに影響を与えることはありませんが、BGP で学習されたルートが IP ルーティング テーブルに組み込まれるかどうかを左右します。

詳細については、「アドミニストレーティブ ディスタンス」のセクションを参照してください。

BGP ピア

BGP スピーカーは他の BGP スピーカーを自動的に検出しません。ユーザ側で BGP スピーカ間の関係を設定する必要があります。BGP ピアは、別の BGP スピーカへのアクティブな TCP 接続を持つ BGP スピーカです。

BGP セッション

BGP は TCP ポート 179 を使用して、ピアとの TCP セッションを作成します。ピア間で TCP 接続が確立されると、各 BGP ピアは最初に相手と、それぞれのすべてのルートを交換し、BGP ルーティング テーブルを完成させます。初期交換以後、BGP ピアはネットワーク トポロジが変化したとき、またはルーティングポリシーが変更されたときに、差分アップデートだけを送信します。更新と更新の間の非アクティブ期間には、ピアは「キープアライブ」と呼ばれる特別なメッセージを交換します。ホールド タイムは、は、次の BGP アップデートまたはキープアライブ メッセージを受信するまでに経過することが許容される、最大時間限度です。

Cisco NX-OS は、次のピア設定オプションをサポートします。

- 個別の IPv4 または IPv6 アドレス：BGP は、リモート アドレスと AS 番号が一致する BGP スピーカとのセッションを確立します。
- 単一 AS 番号の IPv4 または IPv6 プレフィックス ピア：BGP は、プレフィックスおよび AS 番号が一致する BGP スピーカとのセッションを確立します。
- ダイナミック AS 番号プレフィックス ピア：BGP は、プレフィックスと、設定済み AS 番号のリストに載っている AS 番号と一致する BGP スピーカとのセッションを確立します。

プレフィックス ピアおよびインターフェイス ピアのダイナミック AS 番号

Cisco NX-OS では、BGP セッションを確立する AS 番号の範囲またはリストを受け入れます。たとえば IPv4 プレフィックス 192.0.2.0/8 および AS 番号 33、66、99 を使用するように BGP を設定する場合、BGP は 192.0.2.1 および AS 番号 66 を使用してセッションを確立しますが、192.0.2.2 および AS 番号 50 からのセッションは拒否します。

Cisco NX-OS リリース 9.3(6) 以降、ダイナミック AS 番号のサポートは、プレフィックス ピアに加えてインターフェイス ピアにも拡張されています。IPv4 および IPv6 アドレス ファミリ向け IPv6 リンク ローカル経由の BGP インターフェイス ピアリングの設定 （68 ページ）を参照してください。

Cisco NX-OS では、セッションが確立されるまで内部 BGP（iBGP）または外部 BGP（eBGP）セッションとして、プレフィックス ピアをダイナミック AS 番号と関連付けません。iBGP および eBGP の詳細については、「高度な *BGP* の構成」を参照してください。

（注） ダイナミック AS 番号プレフィックス ピア設定は、BGP テンプレートから継承した個々の AS 番号の設定よりも優先します。詳細については、高度な *BGP* の設定を参照してください。

BGP ルータ ID

ピア間で BGP セッションを確立するには、BGP セッションの確立時に、OPEN メッセージで BGP ピアに送信されるルータ ID を BGP に設定する必要があります。BGP ルータ ID は 32 ビット値であり、IPv4 アドレスで表すことがよくあります。ルータ ID はユーザ側で設定できます。ルータ ID はデフォルトで、Cisco NX-OS によってルータのループバック インターフェイスの IPv4 アドレスに設定されます。ルータ上でループバック インターフェイスが設定されていない場合は、ルータ上の物理インターフェイスに設定されている最大の IPv4 アドレスが BGP ルータ ID を表すものとして、ソフトウェアによって選択されます。BGP ルータ ID は、ネットワーク内の BGP ピアごとに一意である必要があります。

BGP にルータ ID が設定されていない場合、BGP ピアとのピアリング セッションを確立できません。

各ルーティング プロセスには、ルータ ID が関連付けられています。ルータ ID は、システムのあらゆるインターフェイスに設定できます。ルータ ID を構成しなかった場合、Cisco NX-OS は次の基準に基づいてルータ識別子を選択します。

- Cisco NX-OS は、他のあらゆるインターフェイスよりも loopback0 を優先します。loopback0 が存在しなかった場合、Cisco NX-OS は、他のあらゆるインターフェイス タイプよりも、最初のループバック インターフェイスを優先します。
- ループバック インターフェイスを構成しなかった場合、Cisco NX-OS はルータ識別子として構成ファイルの最初のインターフェイスを使用します。Cisco NX-OS がルータ識別子を選択した後にいずれかのループバック インターフェイスを設定した場合は、ループバック インターフェイスがルータ識別子となります。ループバック インターフェイスが loopback0 ではなく、loopback0 を IP アドレスで設定した場合は、ルータ ID が loopback0 の IP アドレスに変更されます。
- ルータ ID の元であるインターフェイスが変更されると、新しい IP アドレスがルータ ID となります。他のどのインターフェイスの IP アドレスが変更されても、ルータ ID はまったく変更されません。

BGP パス選択

BGP は、プレフィックスごとの複数パスの送受信と、このパスのアドバタイジングをサポートします。追加 BGP パスの構成については、「高度な *BGP* の構成」を参照してください。

所定のネットワークでパスが追加または削除されるたびに、ベストパス アルゴリズムが実行されます。ベストパス アルゴリズムは、ユーザが BGP 設定を変更した場合にも実行されます。BGP は所定のネットワークで使用できる一連の有効パスの中から、最適なパスを選択します。

Cisco NX-OS は次の手順で、BGP ベストパス アルゴリズムを実行します。

1. 2 つのパスを比較し、どちらが適切かを判別します（「ステップ 1 - 「パスの比較ペア」セクションを参照）。
2. すべてのパスを探索し、全体として最適なパスを選択するためにパスを比較する順序を決定します（ステップ 2 - 「比較順序の決定」セクションを参照）。

3. 新しいベスト パスを使用するに足るだけの差が新旧のベスト パスにあるかどうかを判別します（「ステップ 3：「ベストパス変更抑制の決定」セクションを参照）。

（注） 重要なのは、パート 2 で決定される比較順序です。3 つのパス A、B、C があるとします。Cisco NX-OS が A と B を比較する場合、A を選択します。Cisco NX-OS が B と C を比較する場合、B を選択します。しかし、Cisco NX-OS がA と C を比較した場合、A を選択しません。これは一部の BGP メトリックが同じネイバー自律システムからのパスだけに適用され、すべてのパスにわたっては適用されないからです。

パス選択には、BGP AS パス属性が使用されます。AS パス属性には、アドバタイズされたパスでたどる自律システム番号（AS 番号）のリストが含まれます。BGP 自律システムを自律システムの集合または連合に細分化する場合は、AS パスにローカル定義の自律システムを指定した連合セグメントが含まれます。

（注） VXLAN の導入では、BGP パス選択プロセスが使用されます。このプロセスは、ローカルパスからリモートパスへの通常の選択とは異なります。EVPN アドレスファミリの場合、BGP は MAC モビリティ属性のシーケンス番号を比較し（存在する場合）、より高いシーケンス番号のパスを選択します。比較対象の両方のパスに属性があり、シーケンス番号が同じである場合、BGP はローカルで生成されたパスよりもリモートピアから学習したパスを優先します。詳細については、『Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide』を参照してください。

BGP パス選択：パスびペアの比較

BGP ベストパス アルゴリズムの最初のステップでは、より適切なパスを判別するために 2 つのパスを比較します。次に、Cisco NX-OS が 2 つのパスを比較して、より適切なパスを判別する基本的なステップについて説明します。

1. Cisco NX-OS は、比較のために有効なパスを選択します（たとえば、到達不能なネクスト ホップがあるパスは無効です）。
2. Cisco NX-OS は、重みが最大のパスを選択します。
3. Cisco NX-OS は、ローカル プリファレンスが最大のパスを選択します。
4. パスの一方がローカル起点の場合、Cisco NX-OS はそのパスを選択します。
5. Cisco NX-OS は、AS パスが短い方のパスを選択します。

（注） AS パス長を計算するときに、Cisco NX-OS は連合セグメントを無視し、AS セットを 1 として数えます。詳細については、「*AS* 連合」の項を参照してください。

6. Cisco NX-OS は、起点が低い方のパスを選択します。IGP は EGP よりも低いと見なされます。

7. Cisco NX-OS は、Multi-Exit 識別子（MED）が小さい方のパスを選択します。

   パスのピア自律システムに関係なく、ベストパス アルゴリズムの MED 比較が必ず実行されるように、Cisco NX-OS を設定することもできます。詳細については、「最適パス アルゴリズムの調整」を参照してください。この設定を行わなかった場合、MED 比較が実行されるかどうかは、次のように比較する 2 つのパスの AS パス属性によって決まります。

   パスのピア自律システムに関係なく、ベストパス アルゴリズムの MED 比較が必ず実行されるように、Cisco NX-OS を設定することもできます。この設定を行わなかった場合、Cisco NX-OS によって MED 比較が実行されるかどうかは、次のように比較する 2 つのパスの AS パス属性によって決まります。

   1. パスに AS パスまたは AS_SET から始まる AS パスがない場合、パスは内部であり、Cisco NX-OS は他の内部パスに対して MED を比較します。
   2. AS パスが AS_SEQUENCE から始まる場合、ピア自律システムがシーケンスで最初の AS 番号になり、Cisco NX-OS は同じピア自律システムを持つ他のパスに対して MED を比較します。
   3. AS-path パス に連合セグメントだけが含まれている場合、または連合セグメントで始まり、AS_SET が続いている場合、パスは内部であり、Cisco NX-OS は他の内部パスに対して MED を比較します。
   4. AS パスが連合セグメントで始まり、AS_SEQUENCE が続いている場合、ピア自律システムが AS_SEQUENCE で最初の AS 番号になり、Cisco NX-OS は同じピア自律システムを持つ他のパスに対して MED を比較します。

（注） Cisco NX-OS がパスの指定された MED 属性を受信しなかった場合、Cisco NX-OS は欠落 MED が使用可能な最大値になるように、ユーザがベストパス アルゴリズムを設定していない限り、MED を 0 と見なします。詳細については、「最適パス アルゴリズムの調整」を参照してください。

   5. 非決定性の MED 比較機能がイネーブルの場合、ベスト-パス アルゴリズムでは Cisco IOS スタイルの MED 比較が使用されます。

8. 一方のパスが内部ピアから、他方のパスが外部ピアからの場合、Cisco NX-OS は外部ピアからのパスを選択します。

9. ネクスト ホップ アドレスへの IGP メトリックが異なるパスの場合、Cisco NX-OS は IGP メトリックが小さい方のパスを選択します。

10. Cisco NX-OS は、最後に実行したベストパス アルゴリズムによって選択されたパスを使用します。

ステップ 1 ～ 9 のすべてのパス パラメータが同じ場合、最適パス アルゴリズムを構成し、「ルータ ID の比較」を構成して、両方のパスが eBGP であるときに、ルータ ID の比較を適用できます。その他のすべての場合、ルータ ID の比較はデフォルトで実行されます。

詳細については、「最適パス アルゴリズムの調整」を参照してください。パスに発信元属性が含まれている場合、Cisco NX-OS はその属性をルータ ID として使用して比較します。発信もと属性が含まれていない場合、Cisco NX-OS はパスを送信したピアのルータ ID を使用します。パス間でルータ ID が異なる場合、Cisco NX-OS はルータ ID が小さい方のパスを選択します。

（注） 属性の送信元をルータ ID として使用する場合は、2 つのパスに同じルータ ID を設定することができます。また、同じピア ルータとの 2 つの BGP セッションが可能です。したがって、同じルータ ID で 2 つのパスを受信できます。

11. Cisco NX-OS は、クラスタ長が短いほうのパスを選択します。クラスタ リスト属性の指定されたパスを受け取らなかった場合、クラスタ長は 0 です。
12. Cisco NX-OS は、IP アドレスが小さい方のピアから受信したパスを選択します。ローカル発生のパス（再配布のパスなど）は、ピア IP アドレスが 0 になります。

（注） ステップ 9 以降が同じパスは、マルチパスを設定している場合、マルチパスに使用できます。詳細については、「ロード シェアリングとマルチパス」の項を参照してください。

BGPパス選択：比較の順序の決定

BGP ベストパス アルゴリズム実装の 2 番目のステップでは、Cisco NX-OS がパスを比較する順序を決定します。

1. Cisco NX-OS は、パスをグループに分けます。各グループ内で、Cisco NX-OS はすべてのパス間で MED を比較します。Cisco NX-OSは、「ステップ 1：パス ペアの比較」と同じルールを使用して、2 つのパス間で MED を比較できるかどうかを判断します。この比較では通常、ネイバー自律システムごとに 1 つずつグループが選択されます。**bgp bestpath med always** コマンドを設定すると、Cisco NX-OS はすべてのパスが含まれた 1 グループだけを選択します。
2. Cisco NX-OS は、常に最適な方を維持しながら、グループのすべてのパスを反復することによって、各グループのベスト パスを決定します。Cisco NX-OS は、各パスをそれまでの一時的なベスト パスと比較します。それまでのベスト パスよりも適切な場合は、そのパスが新しく一時的なベスト パスになり、Cisco NX-OS はグループの次のパスと比較します。

3. Cisco NX-OS は、ステップ 2 の各グループで選択されたベスト パスからなる、パス セットを形成します。Cisco NX-OS は、このパス セットでもステップ 2 と同様にそれぞれの比較を繰り返すことによって、全体としてのベスト パスを選択します。

BGP パス選択：最適パス変更抑制の決定

実装の次のパートでは、Cisco NX-OS が新しい最適パスを使用するのか抑制するのかを決定します。新しいベスト パスが古いパスとまったく同じ場合、ルータは引き続き既存のベスト パスを使用できます（ルータ ID が同じ場合）。Cisco NX-OS では引き続き既存のベスト パスを使用することによって、ネットワークにおけるルート変更を回避できます。

抑制機能をオフにするには、ルータ ID を比較するようにベスト パス アルゴリズムを設定します。詳細については、「最適パス アルゴリズムの調整」を参照してください。この機能を設定すると、新しいベスト パスが常にに既存のベスト パスよりも優先されます。

BGP およびユニキャスト RIB

BGP はユニキャスト RIB（ルーティング情報ベース）と通信して、ユニキャスト ルーティング テーブルに IPv4 ルートを格納します。ベスト パスの選択後、ベスト パスの変更をルーティング テーブルに反映させる必要があると BGP が判別した場合、BGP はユニキャスト RIB にルート アップデートを送信します。

BGP はユニキャスト RIB における BGP ルートの変更に関して、ルート通知を受け取ります。さらに、再配布をサポートする他のプロトコル ルートに関するルート通知を受け取ります。

BGP はネクストホップの変更に関する通知も、ユニキャスト RIB から受け取ります。BGP はこれらの通知を使用して、ネクストホップ アドレスへの到達可能性および IGP メトリックを追跡します。

ユニキャスト RIB でネクストホップ到達可能性または IGP メトリックが変更されるたびに、BGP は影響を受けるルートについて、ベスト パス再計算を開始させます。

BGP は IPv6 ユニキャスト RIB と通信し、IPv6 ルートについて、これらの動作を実行します。

BGP プレフィックス独立コンバージェンス

BGP プレフィックス独立コンバージェンス（PIC）エッジ機能は、リンク障害が発生した場合に、BGP バックアップ パスへの BGP IP ルートのコンバージェンスを高速化します。

BGP PIC エッジ機能により、ネットワーク障害後の BGP コンバージェンスが向上します。このコンバージェンスは、IP ネットワークのエッジ障害に適用されます。この機能は、ルーティング情報ベース（RIB）と転送情報ベース（FIB）にバックアップ パスを作成して保存します。これによって、プライマリ パスの障害が発生した場合に、ただちにバックアップ パスが引き継ぐことができ、フォワーディング プレーンの迅速なフェールオーバーが可能になります。BGP PIC エッジは、IPv4 アドレス ファミリのみをサポートします。

BGP PIC エッジが設定されている場合、BGP は、プライマリ ベスト パスに加えて、2 番目のベスト パス（バックアップ パス）も計算します。BGP は、PIC サポートを持つプレフィクス

のベスト パスとバックアップ パスの両方を BGP RIB にインストールします。また BGP は、API を介してリモートの次のホップとともにバックアップ パスを URIB にダウンロードし、その後バックアップとしてマークされたネクスト ホップで FIB を更新します。バックアップ パスにより、単一のネットワーク障害に対処する高速再ルーティング機能が提供されます。

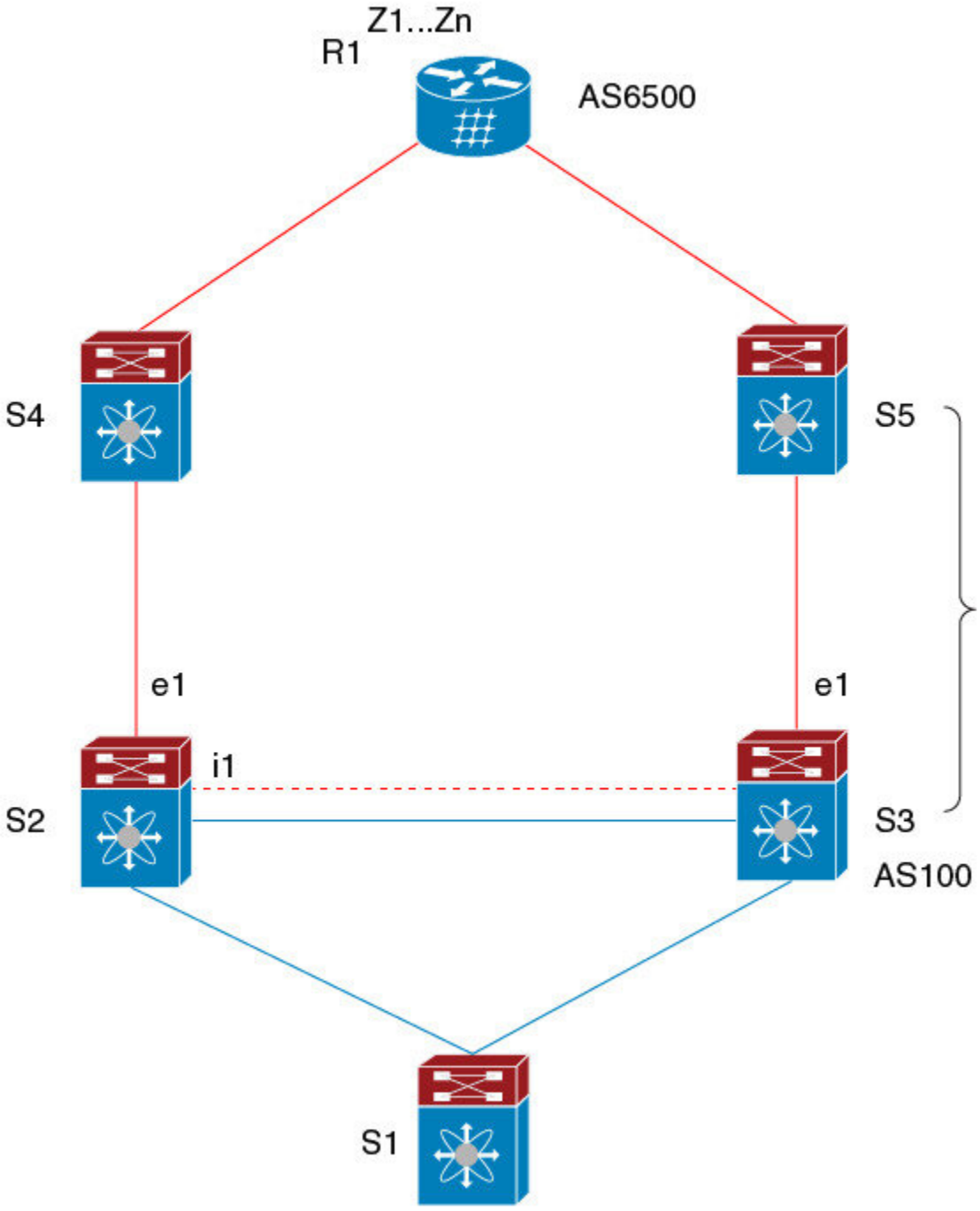
この機能は、ローカル インターフェイスとリモート インターフェイス/リンクの両方の障害を検出して、バックアップ パスが使用されるようにします。

BGP PIC エッジは、ユニパスとマルチパスの両方をサポートします。

BGP PIC エッジ ユニパス

次の図に、BGP PIC エッジ ユニパスのトポロジを示します。

図 *1 : BGP PIC* エッジ ユニパス



この図では次のようになっています。

- S2-S4とS3-S5の間はeBGPセッションです。
- S2-S3 の間は iBGP セッションです。

- S1 からのトラフィックは S2 を使用し、また e1 インターフェイスを使用して Z1..Zn プレフィックスに到達します。
- S2には、Z1…Znに到達するための 2 つのパスがあります。
  - S4 を経由するプライマリ パス
  - S5 を経由するバックアップ パス

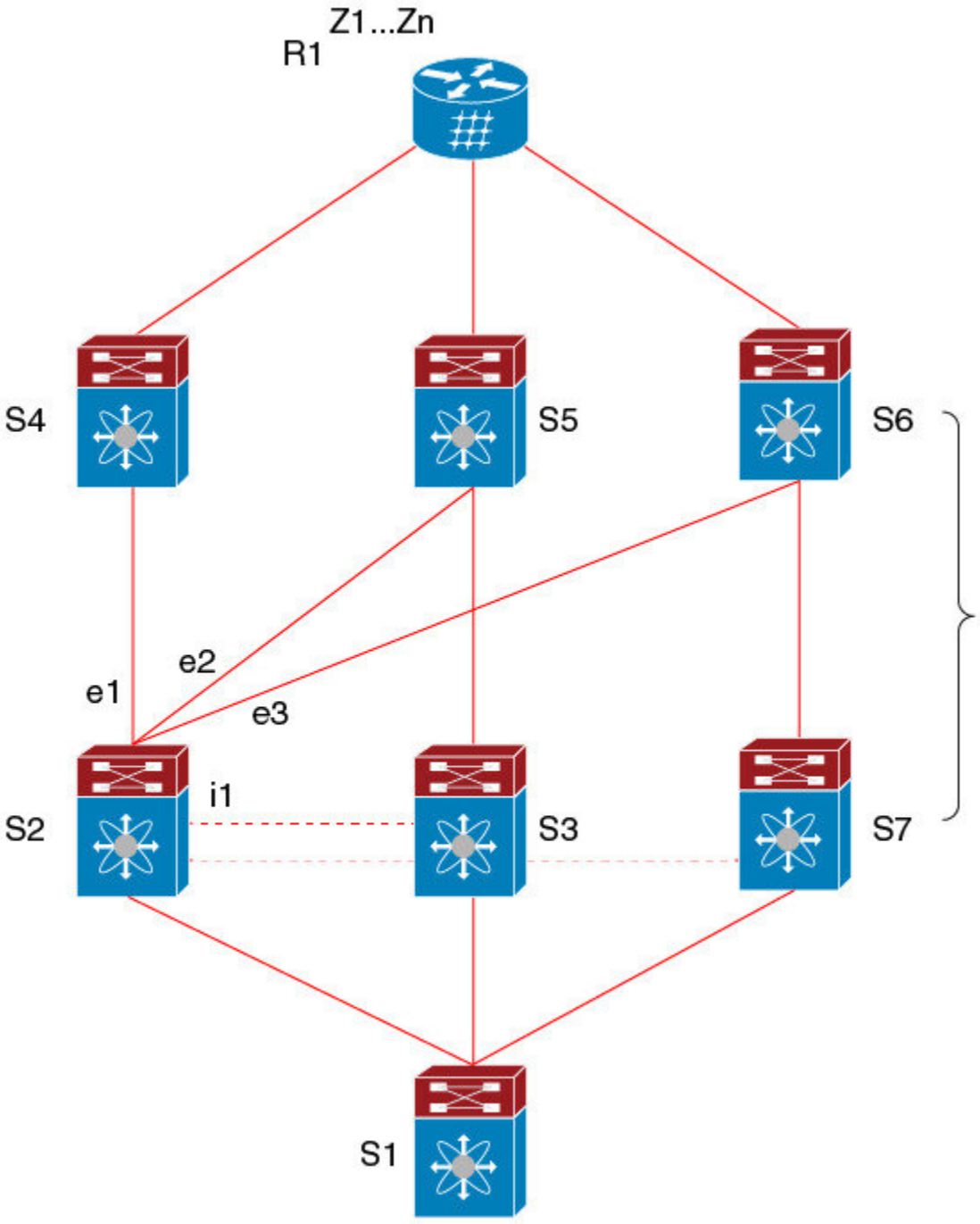
この例では、S3 が S2 に対し、到達すべきプレフィクス Z1…Zn をアドバタイズします（それ自身をネクスト ホップとして）。BGP PIC エッジが有効になっている場合、S2 の BGP は、AS6500へのベストパス（S4 経由）とバックアップパス（S3 または S5 を経由）の両方を RIB にインストールします。その後、RIBは両方のルートを FIBにダウンロードします。

S2-S4 のリンクがダウンすると、S2 上の FIB がリンク障害を検出します。その場合、自動的にプライマリパスからバックアップに切り替えられ、新しいネクスト ホップ S3 がポイントされます。トラフィックは、FIB 内のローカルの高速再コンバージェンスにより迅速に再ルーティングされます。リンク障害イベントを学習した後、S2 上の BGP はベストパス（以前のバックアップ パス）を再計算し、RIB からネクスト ホップ S4 を削除し、S3 をプライマリ ネクスト ホップとして RIB に再インストールします。また、新しいバックアップあればそれも計算し、RIB に通知します。BGP PIC エッジ機能のサポートにより、FIB はプライマリ ルートでのリンク障害の検出時に、BGP が新しいベストパスを選択してコンバージェンスするまで待機することなく、使用可能なバックアップルートに瞬時に切り替えます。こうして、高速な再ルーティングを実現しています。

マルチパスを持つ BGP PIC エッジ

次の図に、BGP PIC エッジ マルチパス トポロジを示します。

図 *2 : BGP PIC* エッジ マルチパス



上記のトポロジでは、次のように所定のプレフィックスに 6 つのパスがあります。

- eBGP パス：e1、e2、e3
- iBGPパス：i1、i2、i3

優先順位は、e1 > e2 > e3 > i1 > i2 > i3 です。

考えられるマルチパスの状況は次のとおりです。

- 設定されたマルチパスなし：
  - ベストパス = e1
  - マルチパス-セット = []
  - バックアップ パス = e2
  - PIC 挙動：e1 が失敗すると、e2 がアクティブになります。
- 双方向の eBGP マルチパスが設定されている
  - ベストパス = e1
  - マルチパス-セット = [e1, e2]
  - バックアップ パス = e3
  - PIC 挙動：アクティブなマルチパスが相互にバックアップされます。すべてのマルチパスが失敗すると、e3 がアクティブになります。
- 3 方向の eBGP マルチパスが設定されている
  - ベストパス = e1
  - マルチパス-セット = [e1, e2, e3]
  - バックアップ パス = i1
  - PIC 挙動：アクティブなマルチパスが相互にバックアップされます。すべてのマルチパスが失敗すると、i1 がアクティブになります。
- 4 方向の eiBGP マルチパスが設定されている
  - – ベストパス = e1
  - – マルチパスセット = [e1, e2, e3, i1]
  - – バックアップパス = i2
  - – PIC 挙動：アクティブなマルチパスが相互にバックアップされます。すべてのマルチパスが失敗すると、i2 がアクティブになります。

等コストマルチパス（ECMP）がイネーブルになっている場合、どのマルチパスもバックアップ パスとして選択されません。

バックアップパスを使用するマルチパスのシナリオでは、すべてのアクティブなマルチパスで同時障害が発生しても、高速コンバージェンスは生じません。

BGP PIC コア

コアの BGP Prefix Independent Convergence（PIC）は、ネットワーク障害後の BGP コンバージェンスを向上させます。たとえば、プロバイダーエッジ（PE）でリンクに障害が発生した場合、ルーティング情報ベース（RIB）は新しいネクスト ホップで転送情報ベース（FIB）を更新します。FIB は、失敗したネクスト ホップを指しているすべての BGP プレフィックス、新しいネクストホップを指すように更新する必要があります。これは、時間とリソースを消費する可能性があります。BGP PIC コアを有効にすると、FIB 内でプレフィックスが階層的にプログラムされます。すべてのプレフィックスは、再帰ネクストホップではなく、ECMP グループを指します。同じ障害が発生した場合、FIBは、プレフィックスを更新せず、新しいネクストホップを指すよう ECMP グループを更新するだけで済みます。これにより、BGP は IGP コンバージェンスを即座に活用できます。

BGP PIC の機能サポート マトリクス

表 *2: BGP PIC* の機能サポート マトリクス

BGP PIC	IPv4 ユニキャスト	IPv6 ユニキャスト
エッジ ユニパス	○	非対応
マルチパスを持つエッジ（複数のアクティブ ECMP、バックアップ 1 つのみ）	○	非対応
コア	○	はい

BGP の仮想化

BGP は、仮想ルーティングおよび転送（VRF）インスタンスをサポートします。

前提条件

BGP を使用するには、次の前提条件を満たしている必要があります。

- BGP を有効にする必要があります（「*BGP* の有効化」の項を参照）。
- システムに有効なルータ ID を設定しておく必要があります。
- Regional Internet Registry（RIR）によって割り当てられたか、またはローカル管理の AS 番号を取得しておく必要があります。
- ネイバー関係を作成しようとするピアに到達可能でなければなりません（Interior Gateway Protocol（IGP）、スタティック ルート、直接接続など）。
- BGP セッションを確立するネイバー環境で、アドレス ファミリを明示的に設定する必要があります。

基本および拡張 BGP に関する注意事項と制限事項

BGP 設定時の注意事項および制約事項は、次のとおりです。

- 十分な規模（ピアあたり数百のピアや数千のルートなど）では、デフォルトの5分間の古いパス タイマーでは、BGP コンバージェンスが完了しないためにタイマーが期限切れになる可能性があるため、グレースフル リスタート メカニズムが失敗する可能性があります。次のコマンドを使用して、コンバージェンス プロセスにかかる実際の時間を確認します。

```
switch# show bgp vrf all all neighbors | in First|RIB
  Last End-of-RIB received 0.022810 after session start
  Last End-of-RIB sent 00:08:36 after session start
  First convergence 00:08:36 after session start with 398002 routes sent
```

- Cisco NX-OS 9.3(5) 以降では、vPC ピアへの TTL 値が 1 のパケットがハードウェア転送されます。

- レコード オプション（-Cr）を指定して SNMP バルク ウォークを使用する場合、大規模なルーティング テーブル（250 K以上）では、SNMP パフォーマンスの低下を避けるために 10 個を超えるレコードを使用しないでください。

- Cisco NX-OS リリース 9.3(5) 以降、コマンドの動作が変更された 3 つのシナリオがあります。

  - ```
    Router bgp 1
         Template peer abc
             Ttl-security hops 30
         Neighbor 1.2.3.4
             Inherit peer abc
    ```

    後で **ebgp-multihop 20** コマンドを入力すると、**ttl-security hops 30** コマンドが存在するため、構成はブロックされます。Cisco NX-OS リリース 9.3(5) 以降、設定はブロックされなくなりました。ただし、 **ttl-security hops** コマンドが優先され、有効な機能になります。

  - ```
    Router bgp 1
         Template peer abc
             Ebgp-multihops 20
         Neighbor 1.2.3.4
             Inherit peer abc
    ```

    後で **ttl-security hops 30** コマンドを入力すると、**ebgp-multihop 20** コマンドが存在するため、構成はブロックされます。Cisco NX-OS リリース 9.3(5) 以降、設定はブロックされなくなりました。ただし、ここでも **ttl-security hops** コマンドが優先され、有効な機能なります。

  - ```
    Router bgp 1
         Template peer abc
             Remote-as 1
         Neighbor 1.2.3.4
             Inherit peer abc
    ```

    後で **ttl-security hops 30** または **ebgp-multihop 20** コマンドを入力すると、ブロックされます。Cisco NX-OS リリース 9.3(5) 以降、設定はブロックされなくなりました。た
    ```

だし、ピアが iBGP ピアになる **remote-as** コマンドが優先されるため、これらの機能はオフになります。

- JSONペイロード を記述する場合は、 RFC 8259： javascript Object Notation (JSON) データ交換用フォーマットで定義されている標準規格のJSONシンタックスを使用します。
- プレフィックス ピアリングは、パッシブ TCP モードでのみ動作します。ピア アドレスがプレフィックス内にある場合、リモート ピアからの着信接続を受け入れます。
- Cisco NX-OS 9.3(5) 以降、vPC ピアへの TTL 値が 1 のパケットは、転送されるハードウェアです。
- **advertise-maps** コマンドを複数回構成することはサポートされていません。
- プレフィックス リスト内の名前は、大文字と小文字が区別されません。一意の名前を使用することを推奨します。大文字と小文字を変更して同じ名前を使用しないでください。たとえば、CTCPrimaryNetworks と CtcPrimaryNetworks は 2 つの異なるエントリではありません。
- ダイナミック AS 番号プレフィックス ピア設定は、BGP テンプレートから継承した個々の AS 番号の設定よりも優先します。
- AS 連合でプレフィックス ピアにダイナミック AS 番号を設定した場合、BGP はローカル連合の AS 番号のみでセッションを確立します。
- ダイナミック AS 番号プレフィックス ピアで作成された BGP セッションは、設定済みの eBGP マルチホップ存続可能時間（TTL）値や直接接続ピアに対するディセーブル済みのチェックを無視します。
- ルータ ID の自動変更およびセッション フラップを避けるために、BGP 用のルータ ID を設定します。
- ピアごとに最大プレフィックス設定オプションを使用し、受信するルート数および使用するシステム リソース数を制限してください。
- Cisco NX-OSリリース 10.6(2)F 以降、BGP の最大プレフィックス構成に新しいオプション discard が導入されました。このオプションを設定して、しきい値に達したときにネイバーから受信した過剰なプレフィックスを破棄できます。
  - この機能は ISSD をサポートしていません。
  - 既存の NX- OS BGP 最大プレフィックスの実装はポストインバウンド ポリシーです。したがって、soft-reconfig が有効になっている場合でも、BGP は受信したルートを保存します。ただし、破棄されたルートは BRIB にインストールされません。現在、インバウンド前の段階でこの構成を適用するオプションはありません。
  - BGP は、破棄されたプレフィックスが存在する場合のヒットレス グレースフル リスタートを保証できません。
- update-source を設定し、BGP/eBGP マルチホップ セッションでセッションを確立します。

- 再配布を設定する場合は、BGP ポリシーを指定します。
- VRF 内で BGP ルータ ID を定義します。
- IPv6ネイバーの場合は、VRFごとにルータIDを設定することを推奨します。VRFにIPv4インターフェイスがない場合、IPv6 BGPネイバーはルータIDがIPv4アドレスである必要があるため、アップしません。数値が最小のループバックIPv4アドレスがルータIDとして選択されます。ループバックアドレスが存在しない場合は、VRFインターフェイスから最も小さいIPアドレスが選択されます。これが存在しない場合、BGPネイバー関係は確立されません。
- キープアライブおよびホールド タイマーの値を小さくすると、BGP セッション フラップが発生する可能性があります。
- **advertisement-interval**コマンドを使用すると、BGP ルーティング アップデートを送信する最小ルート アドバタイズメント インターバル（MRAI）を構成できます。
- **show ip bgp** コマンドは BGP 構成の確認に使用できますが、代わりに **show bgp** コマンドを使用することを推奨します。
- BGP をIGP に再配布するとき、iBGP も再配布されます。この動作を無効にするには、ルート マップに追加 deny 文を挿入します。
- BGP へのエッジ サービス ゲートウェイ（ESG）ルートインジェクトは、重み 0 に割り当てられます。
- iBGP の単一ホップ ピアに対して BFD を有効にするには、物理インターフェイスの **update-source** オプションを構成する必要があります。
- Cisco NX-OS リリース 9.3(3) 以降では、BGP の BFD は BGP IPv4 と IPv6 のプレフィックス ピアでサポートされます。
- VLAN には、次の注意事項および制約事項が **remove-private-as** コマンドに適用されます。
  - これは、eBGP ピアにだけ適用されます。
  - これは、パブリック AS のみのルータのみに適用されます。この制約事項を回避するには、ネイバー単位で **neighbor local-as** コマンドを適用し、ローカル AS 番号をパブリック AS 番号として指定することです。
  - ネイバー コンフィギュレーション モードだけで設定可能となり、ネイバー アドレス ファミリ モードでは設定できません。
  - AS パスにプライベートとパブリック AS 番号を含める場合、プライベート AS 番号は削除されません。
  - AS パスに eBGP ネイバーの AS 番号が含まれている場合、プライベート AS 番号は削除されません。

- その AS パス内のすべての AS 番号がプライベート AS 番号範囲に属する場合のみ、プライベート AS 番号は削除されます。ピアの AS 番号または非プライベート AS 番号が AS パス セグメントに存在する場合、プライベート AS 番号は削除されません。

- BGP テーブル内のネクストホップの一致ルートに AS-Path がない場合、または BGP テーブル内のネクストホップに一致するルートがない場合、traceroute 出力に、ネクストホップとともに AS 番号が出力されることはありません。
- **aggregate-address** を使用する場合 コマンドを使用して集約アドレスを設定し、**suppress-fib-pending** コマンドを使用して BGP ルートを抑制するコマンドを使用する場合、集約のロスレス トラフィックを BGP またはシステム トリガーで保証できません。
- スイッチでFIB抑制をイネーブルにし、ルートプログラミングがハードウェアで失敗すると、BGPはハードウェアでローカルにプログラミングされていないルートをアドバタイズします。
- ネイバー、テンプレート ピア、テンプレート ピアセッション、またはテンプレート ピア ポリシー コンフィギュレーション モードでコマンドを無効にした場合（**inherit peer** または **inherit peer-session** コマンドが存在する場合）、**default** キーワードを使用してコマンドをデフォルトの状態に戻す必要があります。たとえば、実行コンフィギュレーションから **default update-source loopback 0** コマンドを無効にするには、**update-source loopback 0** コマンドを入力する必要があります。
- route-reflector クライアントに next-hop-self が設定されている場合、ルート リフレクタは自身をネクスト ホップとしてクライアントにルートをアドバタイズします。
- 重み付き ECMP に次の注意事項および制約事項が適用されます。
  - 重み付き ECMP 機能は、IPv4 アドレス ファミリでのみサポートされます。
  - BGPは、draft-ietf-idr-link-bandwidth-06.txtで定義されているリンク帯域幅EXTCOMMを使用して、重み付けECMP機能を実装します。
  - BGP は、eBGP ピアと iBGP ピアの両方からリンク帯域幅 EXTCOMM を受け入れることができます。
- IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカルを介した BGP インターフェイス ピアリングには、次の注意事項と制限事項が適用されます。
  - この機能は、複数のインターフェイス間で同じリンクローカルアドレスを設定することをサポートしていません。
  - IP6 リンクローカル静的 IPv6 アドレスを使用して BGP インターフェイス ピアリングを構成する場合は、BGP が機能するようにサブネットがピアと一致していることを確認します。
  - この機能は、論理インターフェイス（ループバック）ではサポートされていません。イーサネット インターフェイス、ポートチャネル インターフェイス、サブインターフェイス、およびブレークアウト インターフェイスのみがサポートされます。
  - Cisco NX-OS リリース 9.3(6) 以降では、VLAN インターフェイスがサポートされます。

- この機能は、リンクローカルアドレスを持つIPv6対応インターフェイスでのみサポートされます。
- この機能は、設定されたプレフィックス ピアとインターフェイスのリモート ピアが同じ場合はサポートされません。
- 次のコマンドはネイバーインターフェイス コンフィギュレーションモードではサポートされていません。
  - **disable-connected-check**
  - **maximum-peers**
  - **update-source**
  - **ebgp-multihop**
- BFD マルチホップおよび次のコマンドは、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカルを介したBGPインターフェイス ピアリングではサポートされません。
  - **bfd-multihop**
  - **bfd multihop interval**
  - **bfd multihop authentication**
- BGPでは、ルートアドバタイズメントのコンバージェンス時間が短縮されます。ルート アドバタイズメント（RA）リンクレベル プロトコルの検出を高速化するには、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカル経由 BGP インターフェイス ピアリングを使用する各IPv6対応インターフェイスで次のコマンドを入力します。

```
interface Ethernet
                        port/slot

ipv6 nd ra-interval 4 min 3
ipv6 nd ra-lifetime 10
```

- Cisco NX-OSリリース 9.3(1)F 以降、 REST API を使用して IPv4 ユニキャストキャスト ルートの BGP ルーティング テーブルを表示するときに、異なるパス タイプを持つ複数のエントリを区別するために、**rn** キーにパス タイプが導入されました。以前のリリースでは、出力は単一のエントリとして表示されます。
- ルート マップ削除機能は、BGP に関連付けられたルート マップ全体の削除をブロックするメカニズムを追加します。ルート マップの削除がブロックされても、ルート マップ ステートメントへの変更は引き続き許可されます。
- ルートマップに複数のシーケンスがある場合、少なくとも1つのシーケンスが使用可能になるまで、ユーザーはルート マップ シーケンスを削除できます。
- ユーザーは、クライアントからのルートマップの前方参照ケースを持つことができます。ただし、ルート マップが作成されて関連付けられると、ルート マップの削除はブロックされます。
- ブロック削除機能は、ノブを使用して動的に構成できます。

- ルート マップへの BGP アソシエーションを削除すること、および単一のトランザクション ペイロードでルート マップ自体を削除することは許可されています。
- ルート マップに BGP アソシエーションを追加することが許可されており、ルート マップの削除に対してエラーをスローする必要があります。
- 以下は、デュアル ステージに関連する動作のリストです。
  - ノブと削除が同時に発生した場合、デュアル ステージは検証し、コミットせずにエラーをスローする必要があります。
  - ノブはすでに存在し、ルートマップ削除がデュアル ステージで発生する場合、エラーをスローする必要があります。
  - ルート マップと CLI ノブが異なる順序のシングル コミットである場合、エラーをスローする必要があります。
  - ノブが有効になっておらず、ルート マップの削除がデュアル ステージで発生した場合は、正常に実行する必要があります。
  - 1回のベリファイで、「cliノブが無効かつルートマップの削除」が実行された場合、ルートマップの削除が許可されます。
- BGP テンプレートで使用されるルート マップがいずれの BGP ネイバーにも継承されない場合、ルート マップ全体の削除は引き続きブロックされます。
- BGP によって所有されているが、bgpInst の一部ではない、vrf コンテキストの下にいくつかのコマンドがあります。
- VPN アドレス ファミリ (L3VPN および EVPN) がサポートされていないため、同盟ピアから受信したルートは VPN アドレス ファミリでアドバタイズされません。
- Cloudscale IPv6 リンクローカル BGP のサポートには、512 を超える ing-sup TCAM リージョンを切り分ける必要があります (これを有効にするには、リロードが必要です)。
- VPN アドレス ファミリ (L3VPN および EVPN) がサポートされていないため、同盟ピアから受信したルートは VPN アドレス ファミリでアドバタイズされません。
- Cisco NX-OS リリース 10.3(1)F 以降、BGP は Cisco Nexus 9808 スイッチでサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、BGP は Cisco Nexus 9804 スイッチでサポートされます。
- Cisco NX-OS リリース 10.3(1)F 以降、VXLAN EVPN は、Cisco Nexus 9808 スイッチで、トランジットとしてのみサポートされます。
- Cisco NX-OS リリース 10.4(1)F 以降、VXLAN EVPN は、Cisco Nexus 9804 スイッチで、トランジットとしてのみサポートされます。
- 暗号化復号タイプ 6 は、BGP パスワードとキーチェーンではサポートされていません。

- Cisco NX-OS リリース 10.4(1)F 以降、BGP は、Cisco Nexus 9808 および 9804 スイッチを搭載した Cisco Nexus X98900CD-A および X9836DM-A ライン カードでサポートされます。
- Cisco NX-OS リリース 10.6(1)F 以降、BGP は Cisco Nexus N9336C-SE1 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 10.6(1)F 以降、ECMP は Cisco Nexus N9336C-SE1 プラットフォーム スイッチでサポートされます。

デフォルト設定

表 *3:* デフォルトの *BGP* パラメータ

パラメータ	デフォルト
BGP 機能	ディセーブル
キープアライブインターバル	60 秒
ホールド タイマー	180 秒
BGP PIC エッジ	ディセーブル
Auto-summary	常に無効
同期	常に無効

基本的 BGP の設定

CLI コンフィギュレーションモード

以下の項では、BGP に対応する各 CLI コンフィギュレーション モードの開始方法について説明します。現行のモードで **?** コマンドを入力すると、そのモードで使用可能なコマンドを表示できます。

グローバル コンフィギュレーション モード

グローバル コンフィギュレーション モードは、BGP プロセスを作成したり、AS 連合、ルート ダンプニングなどの拡張機能を設定したりする場合に使用します。詳細については、「高度な *BGP* の構成」を参照してください。

次に、ルータ コンフィギュレーション モードを開始する例を示します。

```
switch# configuration
switch(config)# router bgp 64496
switch(config-router)#
```

BGP は VRF をサポートしています。ネットワークで VRF を使用する場合は、適切な VRF 内でBGP を設定できます。設定の詳細については、「仮想化の設定」の項を参照してください。

次に、VRF コンフィギュレーション モードを開始する例を示します。

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)#
```

アドレス ファミリ設定モード

任意で、BGP がサポートするアドレス ファミリを設定できます。アドレス ファミリ用の機能を設定する場合は、ルータ 設定モードで address-family コマンドを使用します。ネイバーに対応する特定のアドレス ファミリを設定する場合は、ネイバー設定モードで address-family コマンドを使用します。

ルート再配布、アドレス集約、ロードバランシングなどの拡張機能を使用する場合は、アドレス ファミリを設定する必要があります。

次に、ルータ設定モードからアドレス ファミリ設定モードを開始する例を示します。

```
switch(config)# router bgp 64496
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)#
```

次に、VRF を使用している場合に、VRF アドレス ファミリ設定モードを開始する例を示します。

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# address-family ipv6 unicast
switch(config-router-vrf-af)#
```

ネイバー コンフィギュレーション モード

Cisco NX-OS には、BGP ピアを設定するためのネイバー コンフィギュレーション モードがあります。ネイバー コンフィギュレーション モードを使用して、ピアのあらゆるパラメータを設定できます。

次に、ネイバー コンフィギュレーション モードを開始する例を示します。

```
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1
switch(config-router-neighbor)#
```

次に、VRF ネイバー コンフィギュレーション モードを開始する例を示します。

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# neighbor 192.0.2.1
switch(config-router-vrf-neighbor)#
```

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

アドレス ファミリ固有のネイバー設定を入力し、ネイバーのアドレス ファミリをイネーブルにするには、ネイバー コンフィギュレーション サブモード内のアドレス ファミリ コンフィ

ギュレーションサブモードを使用できます。このモードは、所定のネイバーに認められるプレフィックス数の制限、eBGP のプライベート AS 番号の削除といった拡張機能に使用します。

RFC 5549 が導入されているため、IPv6 アドレスを持つネイバーに IPv4 アドレス ファミリを設定できます。

この例は、IPv4 アドレスでネイバーのための IPv4 ネイバー アドレス ファミリ設定モードを入力する方法を示します。

```
switch(config)# router bgp 64496
switch(config-router# neighbor 192.0.2.1
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

この例は、IPv6 アドレスでネイバーのための IPv4 ネイバー アドレス ファミリ設定モードを入力する方法を示します。

```
switch(config)# router bgp 64496
switch(config-router# neighbor 2001:db8::/64 eui64
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

この例は、IPv4 アドレスでネイバーのための VRF IPv4 ネイバー アドレス ファミリ設定モードを入力する方法を示します。

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# neighbor 209.165.201.1
switch(config-router-vrf-neighbor)# address-family ipv4 unicast
switch(config-router-vrf-neighbor-af)#
```

この例は、IPv6 アドレスでネイバーのための VRF IPv4 ネイバー アドレス ファミリ設定モードを入力する方法を示します。

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# neighbor 2001:db8::/64 eui64
switch(config-router-vrf-neighbor)# address-family ipv4 unicast
switch(config-router-vrf-neighbor-af)#
```

基本的 BGP の設定

ベーシック BGP を設定するには、BGP を有効にして、BGP ピアを設定する必要があります。ベーシック BGP ネットワークの設定は、いくつかの必須作業と多数の任意の作業からなります。BGP ルーティング プロセスおよび BGP ピアの設定は必須です。

（注） Cisco IOS の CLI に慣れている場合、この機能の Cisco NX-OS コマンドは従来の Cisco IOS コマンドと異なる点があるため注意が必要です。

BGPの有効化

BGP を設定するには、その前に BGP を有効にする必要があります。

手順の概要

1. **configure terminal**
2. **[no] feature bgp**
3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

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)#`
		設定モードに入ります。

ステップ 2	**[no] feature bgp** 例： `switch(config)# feature bgp`	BGP を有効にします。 この機能を無効化するには、このコマンドの **no** 形式を使用します。
ステップ 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

BGP インスタンスの作成

BGP インスタンスを作成し、BGP インスタンスにルータ ID を割り当てることができます。詳細については、「*BGP ルータ ID*」のセクションを参照してください。

始める前に

- BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。
- BGP はルータ ID（設定済みループバック アドレスなど）を取得できなければなりません。

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)#`
		コンフィギュレーション モードに入ります。

ステップ 2	[**no**] **router bgp** {*autonomous-system-number* | *auto*}
	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。
		auto オプションは、システム MAC アドレスに基づいて 4 バイトのプライベート自律システム番号を自動的に生成します。
		BGP プロセスおよび関連する構成を削除するには、このコマンドで **no** オプションを使用します。

ステップ 3	**router-id** {*ip-address* | *auto*} 例： `switch(config-router)# router-id 192.0.2.255`	（任意）BGP ルータ ID を設定します。この IP アドレスによって、この BGP スピーカを特定します。 「auto」オプションは、システム MAC アドレスに基づく BGP ルータ ID を有効にします。

ステップ 4	（任意） **address-family** {**ipv4**|**ipv6**} {**unicast**|**multicast**}
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		IPv4 または IPv6 アドレス ファミリに対してグローバル 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	（任意） **network** {*ip-address/length* | *ip-address* **mask** *mask*} [**route-map** *map-name*]
	例：
	`switch(config-router-af)# network 10.10.10.0/24`
	例：
	`switch(config-router-af)# network 10.10.10.0 mask 255.255.255.0`
		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。
		エクステリア プロトコルの場合、network コマンドでアドバタイズするネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を決定します。

ステップ 6	（任意） **show bgp all** 例： `switch(config-router-af)# show bgp all`	すべての BGP アドレス ファミリに関する情報を表示します。

	コマンドまたはアクション	目的
ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IPv4 ユニキャスト アドレス ファミリを指定して BGP をイネーブルに設定し、アドバタイズするネットワークを 1 つ追加する例を示します。

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# network 192.0.2.0
switch(config-router-af)# copy running-config startup-config
```

BGP インスタンスの再起動

BGP インスタンスを再起動し、そのインスタンスのすべてのピア セッションをクリアできます。

BGPインスタンスを再起動し、関連付けられたすべてのピアを削除するには、次のコマンドを使用します。

手順の概要

1. **restart bgp***instance-tag*

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**restart bgp***instance-tag* 例： `switch(config)# restart bgp 201`	BGPインスタンスを再起動し、すべてのピアリング セッションをリセットまたは再確立します。

BGP のシャットダウン

設定を維持しながら、BGP プロトコルをシャット ダウンして BGP を正常に無効にできます。

BGP をシャットダウンするには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **shutdown**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**shutdown** 例： `switch(config-router)# shutdown`	BGPインスタンスを再起動し、すべてのピアリングセッションをリセットまたは再確立します。

BGP ピア設定

BGP プロセス内で BGP ピアを設定できます。BGP ピアごとに、関連付けられたキープアライブ タイマーとホールド タイマーがあります。これらのタイマーは、グローバルに設定することも、BGP ピアごとに設定することもできます。ピア設定はグローバル設定を上書きします。

（注） ピアごとに、ネイバー コンフィギュレーション モードでアドレス ファミリを設定する必要があります。

始める前に

- BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** {*ip-address* | *ipv6-address*} **remote-as** {*as-number* | *external* | *internal*}
4. **remote-as** {*as-number* | *external* | *internal*}
5. （任意） **description** *text*
6. （任意） **timers***keepalive-time hold-time*
7. （任意） **shutdown**
8. **address-family**{**ipv4**|**ipv6**} {**unicast**|**multicast**}
9. （任意） **weight** *value*
10. （任意） **show bgp** {**ipv4**|**ipv6**} {**unicast**|**multicast**} **neighbors**
11. （任意） **copy running-config startup-config**

手順の詳細

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)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**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。

ステップ 3	**neighbor** {*ip-address* | *ipv6-address*} **remote-as** {*as-number* | *external* | *internal*}
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 64497`
	`switch(config-router)# neighbor`
		リモート BGP ピアの IPv4 アドレスまたは IPv6 アドレスおよび AS 番号を設定します。*The ip-address* 形式は x.x.x.x です。*ipv6-address* の形式は A:B::C:D です。
		remote-as 値を手動で指定することなく、external および internal オプションを使用すると、eBGP および iBGP セッションを確立できます。

ステップ 4	**remote-as** {*as-number* | *external* | *internal*} 例： `switch(config-router-neighbor)# remote-as 64497`	リモート外部 BGP ピアの AS 番号を構成します。 remote-as 値を手動で指定することなく、external および internal オプションを使用すると、eBGP および iBGP セッションを確立できます。

ステップ 5	（任意） **description** *text*
	例：
	`switch(config-router-neighbor)# description Peer Router B`
	`switch(config-router-neighbor)#`
		ネイバーの説明を追加します。最大 80 文字の英数字ストリングを使用できます。

ステップ 6	（任意） **timers***keepalive-time hold-time* 例： `switch(config-router-neighbor)# timers 30 90`	ネイバーのキープアライブおよびホールド タイムを表す BGP タイマー値を追加します。指定できる範囲は 0 ～ 3600 秒です。デフォルト値は、キープアライブ タイムで 60 秒、ホールド タイムで 180 秒です。 （注） ホールドタイマーが 10 秒以下の BGP セッションは、BGP セッションが 60 秒以上稼働するまで有効になりません。セッションが 60 秒間稼働すると、ホールドタイマーは構成どおりに動作します。

	コマンドまたはアクション	目的
ステップ 7	（任意） **shutdown** 例： `switch(config-router-neighbor)# shutdown`	この BGP ネイバーを管理目的でシャットダウンします。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。

ステップ 8	**address-family**{**ipv4**|**ipv6**} {**unicast**|**multicast**}
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		ユニキャスト IPv4 または IPv6 アドレス ファミリに対応するネイバー アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 9	（任意） **weight** *value* 例： `switch(config-router-neighbor-af)# weight 100`	このネイバーからのルートのデフォルトの重みを設定します。範囲は 0 ～ 65535 です。 このネイバーから学習したすべてのルートに、まず重みが割り当てられます。特定のネットワークへのルートが複数ある場合、最大の重みを持つルートが優先ルートとして選ばれます。**set weight route-map** コマンドで割り当てられた重みは、このコマンドで割り当てられた重みを上書きします。 BGP ピア ポリシー テンプレートを指定した場合、テンプレートのメンバーすべてが、このコマンドで設定された特性を継承します。
ステップ 10	（任意） **show bgp** {**ipv4**|**ipv6**} {**unicast**|**multicast**} **neighbors** 例： `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	BGP ピアに関する情報を表示します。
ステップ 11	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、BGP ピアの設定例を示します。

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1 remote-as 64497
switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor)# weight 100
switch(config-router-neighbor-af)# copy running-config startup-config
```

プレフィックス ピアのダイナミック AS 番号の設定

BGP プロセス内で複数の BGP ピアを設定できます。BGP セッションの確立をルート マップの単一の AS 番号または複数の AS 番号に制限できます。

プレフィックス ピアのダイナミック AS 番号を介して設定された BGP セッションは、**ebgp-multihop** を無視します コマンドと **disable-connected-check** コマンドを使用する必要があります。

ルートマップの AS 番号のリストは変更できますが、ルートマップ名を変更するには no neighbor コマンドを使用する必要があります。設定されたルート マップの AS 番号に変更を加えた場合、新しいセッションのみに影響します。

始める前に

- BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *prefix* **remote-as route-map** *map-name*
4. **neighbor-as** *as-number*
5. （任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
6. （任意） **copy running-config startup-config**

手順の詳細

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)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**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。

ステップ 3	**neighbor** *prefix* **remote-as route-map** *map-name*
	例：
	`switch(config-router)# neighbor 192.0.2.0/8 remote-as routemap BGPPeers`
	`switch(config-router-neighbor)#`
		IPv4 プレフィックスまたは IPv6 プレフィックス、およびリモート BGP ピアの受け付けられた AS 番号のリストのルートマップを設定します。IPv4 の *prefix* 形式は、x.x.x.x/長さ長さの範囲は 1 ～ 32 です。IPv6 の場合、*prefix* の形式は「A:B::C:D/長さ」です。長さの範囲は 1 ～ 128 です。

	コマンドまたはアクション	目的
		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 4	**neighbor-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 5	（任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** 例： `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	BGP ピアに関する情報を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、プレフィックス ピアのダイナミック AS 番号を設定する例を示します。

```
switch# configure terminal
switch(config)# route-map BGPPeers
switch(config-route-map)# match as-number 64496, 64501-64510
switch(config-route-map)# match as-number as-path-list List1, List2
switch(config-route-map)# exit
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.0/8 remote-as route-map BGPPeers
switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-af)# end
switch# copy running-config startup-config
```

ルート マップについては、Route Policy Manager の設定を参照してください。

BGP PIC エッジの設定

BGP PIC エッジを設定するには、次の手順に従います。

（注） BGP PIC エッジ機能は、IPv4 アドレス ファミリのみをサポートします。

始める前に

BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **address-family ipv4 unicast**
4. **[no] additional-paths install backup**
5. （任意） **copy running-config startup-config**

手順の詳細

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)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**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。

ステップ 3	**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アドレス ファミリ構成モードを開始します。

ステップ 4	**[no] additional-paths install backup**
	例：
	`switch(config-router-af)#`
	`  [no] additional-paths install backup`
		ルーティング テーブルにバックアップ パスをインストールする BGP をイネーブルにします。

ステップ 5	（任意） **copy running-config startup-config**
	例：
	`switch(config-router-af)# end`
	`switch# copy running-config startup-config`
		この設定変更を保存します。

例

次の例は、IPv4 ネットワークで BGP PIC エッジをサポートするように、デバイスを設定する方法を示しています。

```
interface Ethernet2/2
 ip address 1.1.1.5/24
 no shutdown

interface Ethernet2/3
 ip address 2.2.2.5/24
 no shutdown

router bgp 100
 address-family ipv4 unicast
  additional-paths install backup
 neighbor 2.2.2.6
   remote-as 100
   address-family ipv4 unicast
```

BGP が 2 つのネイバー（1.1.1.6 と 2.2.2.6）から同じプレフィックス（99.0.0.0/24 など）を受信した場合、両方のパスが URIB にインストールされます。一方はプライマリ パスになり、もう一方はバックアップ パスになります。

BGP 出力：

```
switch(config)# show ip bgp 99.0.0.0/24
BGP routing table information for VRF default, address family IPv4 Unicast BGP routing
table entry
for 99.0.0.0/24, version 4
Paths: (2 available, best #2)
Flags: (0x00001a) on xmit-list, is in urib, is best urib route

Path type: internal, path is valid, not best reason: Internal path, backup path AS-Path:
 200 , path
sourced external to AS
2.2.2.6 (metric 0) from 2.2.2.6 (2.2.2.6)
Origin IGP, MED not set, localpref 100, weight 0

Advertised path-id 1
Path type: external, path is valid, is best path AS-Path: 200 , path sourced external
to AS
1.1.1.6 (metric 0) from 1.1.1.6 (99.0.0.1)
Origin IGP, MED not set, localpref 100, weight 0

Path-id 1 advertised to peers: 2.2.2.6
```

URIB 出力：

```
switch(config)# show ip route 99.0.0.0/24
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>
99.0.0.0/24, ubest/mbest: 1/0
*via 1.1.1.6, [20/0], 14:34:51, bgp-100, external, tag 200
via 2.2.2.6, [200/0], 14:34:51, bgp-100, internal, tag 200 (backup)
```

UFIB 出力：

```
switch# show forwarding route 123.1.1.0 detail module 8
Prefix 123.1.1.0/24, No of paths: 1, Update time: Wed Jul 11 19:00:12 2018
Vobj id: 141   orig_as: 65002   peer_as: 65100  rnh: 10.3.0.2
10.4.0.2           Ethernet8/4          DMAC: 0018.bad8.4dfd
```

```
packets: 2  bytes: 3484   Repair path  10.3.0.2   Ethernet8/3 DMAC: 0018.bad8.4dfd
packets: 0
bytes: 1
```

BGP PIC コアの設定

BGP PIC Core を設定するには、次のステップに従います。

手順の概要

1. **configure terminal**
2. **[no] system pic-core**
3. **copy running-config startup-config**
4. **reload**

手順の詳細

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] system pic-core** 例： `switch(config)# system pic-core`	PIC の有効化を管理します。
ステップ 3	**copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。
ステップ 4	**reload** 例： `switch(config)# reload`	デバイス全体をリブートします。

BGP 情報の消去

BGP 情報を消去するには、次のコマンドを使用します。

コマンド	目的
clear bgp all {*neighbor* | * | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	すべてのアドレス ファミリから 1 つ以上のネイバーをクリアします。*を指定すると、すべてのアドレス ファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 アドレス。 • *as-number*：自律システム番号。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。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • *prefix*：IPv4 または IPv6 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear bgp all dampening [**vrf** *vrf-name*]	すべてのアドレス ファミリのルート フラップ ダンプニング ネットワークをクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と-小文字は区別されます。
clear bgp all flap-statistics [**vrf** *vrf-name*]	すべてのアドレス ファミリのルート フラップ統計情報をクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と-小文字は区別されます。
clear bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} **dampening** [**vrf** *vrf-name*]	選択したアドレス ファミリのルート フラップ ダンプニング ネットワークをクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と小文字は区別されます。
clear bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} **flap-statistics** [**vrf** *vrf-name*]	選択したアドレス ファミリのルート フラップ統計情報をクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と-小文字は区別されます。

コマンド	目的
clear bgp {**ipv4** | **ipv6** } {*neighbor* |* | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	選択したアドレス ファミリから 1 つ以上のネイバーをクリアします。*を指定すると、そのアドレス ファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 アドレス。 • *as-number*：自律システム番号。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。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • *prefix*：IPv4 または IPv6 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

コマンド	目的
clear bgp {**ip** {**unicast** | **multicast**}} {*neighbor* |* |*as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	1 つ以上のネイバーをクリアします。* を指定すると、そのアドレス ファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 アドレス。 • *as-number*：自律システム番号。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。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • *prefix*：IPv4 または IPv6 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear bgp dampening [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ ダンプニングをクリアします。引数は次のとおりです。 • *ip-neighbor*：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4 そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

コマンド	目的
clear bgp flap-statistics [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ統計情報をクリアします。引数は次のとおりです。 • *ip-neighbor*：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4 そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear ip mbgp {**ip** {**unicast** | **multicast**}} {*neighbor* | * | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	1 つ以上のネイバーをクリアします。* を指定すると、そのアドレス ファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 アドレス。 • *as-number*：自律システム番号。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。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • *prefix*：IPv4 または IPv6 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

コマンド	目的
clear ip mbgp dampening [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ ダンプニングをクリアします。引数は次のとおりです。 • *ip-neighbor*：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4 そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear ip mbgp flap-statistics [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ統計情報をクリアします。引数は次のとおりです。 • *ip-neighbor*：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4 そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

ベーシック BGP の設定の確認

BGP 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 all [summary] [**vrf** *vrf-name*]	すべてのアドレス ファミリについて、BGP 情報を表示します。
show bgp convergence [**vrf** *vrf-name*]	すべてのアドレス ファミリについて、BGP 情報を表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **community** [**regexp** *expression* | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]} [**vrf** *vrf-name*]	BGP コミュニティと一致する BGP ルートを表示します。

コマンド	目的
show bgp [**vrf** *vrf-name*] {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community-list** *list-name* [**vrf** *vrf-name*]	BGP コミュニティ リストと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **extcommunity** [**regexp** *expression* | [**generic** [**non-transitive** | **transitive**] *aa4:nn* [**exact-match**]} [**vrf** *vrf-name*]	BGP 拡張コミュニティと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **extcommunity-list** *list-name* [**exact-match**]} [**vrf** *vrf-name*]	BGP 拡張コミュニティ リストと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* {**dampening** **dampened-paths** [**regexp** *expression*]} [**vrf** *vrf-name*]	BGP ルート ダンプニングの情報を表示します。ルート フラップ ダンプニング情報を消去するには、**clear bgp dampening** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **history-paths** [**regexp** *expression*] [**vrf** *vrf-name*]	BGP ルート ヒストリ パスを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **filter-list** *list-name* [**vrf** *vrf-name*]	BGP フィルタ リストの情報を表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*]	BGP ピアの情報を表示します。これらのネイバーを消去するには、**clear bgp neighbors** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} **unicast neighbors** [*ip-address* | *ipv6-prefix*] { [**advertised-routes** |**received-routes**] } [**detail**] [**vrf** *vrf-name*] **show bgp** {**ipv4** | **ipv6**} **unicast neighbors** [*ip-address* | *ipv6-prefix*] [**routes**] { [**advertised** |**received**] } [**detail**] [**vrf** *vrf-name*]	すべてのルートの詳細情報を表示します。 • インバウンド ルート マップを評価する前にピアから受信しました 。 • アウトバウンド ルート マップによって属性を更新する前に、ピアにアドバタイズされます。
show bgp {**ipv4** | **ipv6**} **unicast neighbors** [*ip-address* | *ipv6-prefix*] [**routes**] [**detail**] [**vrf** *vrf-name*]	インバウンド ルート マップの評価後に、このピアから受信したすべてのルートの詳細情報を表示します。
show bgp {**ipv4** | **ipv6**} **unicast neighbors** [*ip-address* | *ipv6-prefix*] [**advertised-routes** **processed**] [**vrf** *vrf-name*]	**処理された**オプションを使用してアウトバウンド ルート マップによってパス属性を更新した後、ピアにアドバタイズされたすべてのルートの要約情報を表示します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} **unicastneighbors** [*ip-address* | *ipv6-prefix*] [**advertised-routes processed**] [**detail**] [**vrf** *vrf-name*]	処理されたオプションを使用してアウトバウンドルートマップによってパス属性を更新した後、ピアにアドバタイズされたすべてのルートの詳細情報を表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] {**nexthop** | **nexthop-database**} [**vrf** *vrf-name*]	BGP ルートネクストホップの情報を表示します。
show bgp paths	BGP パス情報を表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **policy** *name* [**vrf** *vrf-name*]	BGP ポリシー情報を表示します。ポリシー情報を消去するには、**clear bgp polic** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **prefix-list** *list-name* [**vrf** *vrf-name*]	プレフィックスリストと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **received-paths** [**vrf** *vrf-name*]	ソフト再構成用に保管されている BGP パスを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **regexp** *expression* [**vrf** *vrf-name*]	AS_path 正規表現と一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **route-map** *map-name* [**vrf** *vrf-name*]	ルートマップと一致する BGP ルートを表示します。
show bgp peer-policy *name* [**vrf** *vrf-name*]	BGP ピア ポリシー情報を表示します。
show bgp peer-session *name* [**vrf** *vrf-name*] show bgp peer-session	BGP ピア セッション情報を表示します。
show bgp peer-template *name* [**vrf** *vrf-name*]	BGP ピア テンプレート情報を表示します。ピア テンプレートのすべてのネイバーを消去するには、**clear bgp peer-template** コマンドを使用します。
show bgp process	BGP プロセス情報を表示します。
show {**ipv** | **ipv6**} **bgp** [*options*]	BGP のステータスと構成情報を表示します。
show {**ipv** | **ipv6**} **mbgp** [*options*]	BGP のステータスと構成情報を表示します。

コマンド	目的
show running-configuration bgp	現在実行中の BGP コンフィギュレーションを表示します。

BGP 統計情報のモニタリング

BGP の統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **flap-statistics** [**vrf** *vrf-name*]	BGP ルートフラップの統計情報を表示します。これらの統計情報をクリアするには、**clear bgp flap-statistics command** を使用します。
show bgp sessions [**vrf** *vrf-name*]	すべてのピアの BGP セッションを表示します。これらの統計情報をクリアするには、**clear bgp sessions** コマンドを使用します。
show bgp statistics	BGP 統計情報を表示します。

ベーシック BGP の設定例

次に、ベーシック BGP 設定の例を示します。

```
switch(config)# feature bgp
switch(config)# router bgp 64496
switch(config-router)# neighbor 2001:ODB8:0:1::55 remote-as 64496
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# next-hop-self
```

関連項目

BGP の関連項目は、次のとおりです。

- 高度な *BGP* の構成
- *Route Policy Manager* の設定

次の作業

次の機能の詳細については、「高度な *BGP* の構成」を参照してください：

- ピア テンプレート
- ルートの再配布
- ルート マップ

その他の参考資料

BGP の実装に関連する詳細情報については、次の項を参照してください。

ベーシック BGP の MIB

MIB	MIB のリンク
BGP に関連する MIB	MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

高度な BGP の構成

拡張 BGP について

BGP は、組織または自律システム間のループフリー ルーティングを実現する、インタードメイン ルーティング プロトコルです。Cisco NX-OS は BGP バージョン 4 をサポートしています。BGP v4 に組み込まれているマルチプロトコル拡張機能を使用すると、IP マルチキャスト ルートおよび複数のレイヤ 3 プロトコル アドレス ファミリに関するルーティング情報を BGP に伝送させることができます。BGP では、他の BGP 対応デバイス（BGP ピア）との間で TCP セッションを確立するために、信頼できるトランスポート プロトコルとして TCP を使用します。外部組織に接続するときには、ルータが外部 BGP（eBGP）ピアリング セッションを作成します。同じ組織内の BGP ピアは、内部 BGP（iBGP）ピアリング セッションを通じて、ルーティング情報を交換します。

Cisco NX-OS リリース 10.5(1)F から「基本 BGP を構成」の章と「高度な BGP を構成」の章は、まとめられて「BGP を構成」の章になりました。

ピア テンプレート

BGP ピア テンプレートを使用すると、類似した BGP ピア間で再利用できる共通のコンフィギュレーションブロックを作成できます。各ブロックでは、ピアに継承させる一連の属性を定義できます。継承した属性の一部を上書きすることもできるので、非常に柔軟性のある方法で、繰り返しの多い BGP の設定を簡素化できます。

Cisco NX-OS は、3 種類のピア テンプレートを実装します。

- peer-session テンプレートでは、トランスポートの詳細、ピアのリモート自律システム番号、セッション タイマーなど、BGP セッション属性を定義します。peer-session テンプレートは、別の peer-session テンプレートから属性を継承することもできます（ローカル定義の属性によって、継承した peer-session 属性は上書きされます）。
- peer-policy テンプレートでは、着信ポリシー、発信ポリシー、フィルタ リスト、プレフィックス リストを含め、アドレス ファミリに依存する、ピアのポリシー要素を定義します。

peer-policy テンプレートは、一連の peer-policy テンプレートからの継承が可能です。Cisco NX-OS は、継承設定のプリファレンス値で指定された順序で、これらの peer-policy テンプレートを評価します。最小値が大きい値よりも優先されます。

- peer テンプレートは、peer-session および peer-policy テンプレートからの継承が可能であり、ピアの定義を簡素化できます。peer テンプレートの使用は必須ではありませんが、peer テンプレートによって再利用可能なコンフィギュレーション ブロックが得られるので、BGP の設定を簡素化できます。

認証

BGP ネイバー セッションに認証を設定できます。この認証方式によって、ネイバーに送られる各 TCP セグメントに MD5 認証ダイジェストが追加され、不正なメッセージや TCP セキュリティ アタックから BGP が保護されます。

（注） MD5 パスワードは、BGP ピア間で一致させる必要があります。

ルート ポリシーおよび BGP セッションのリセット

BGP ピアにルート ポリシーを関連付けることができます。ルート ポリシーではルート マップを使用して、BGP が認識するルートを制御または変更します。着信または発信ルート アップデートに関するルート ポリシーを設定できます。ルート ポリシーはプレフィックス、AS_path 属性など、さまざまな条件で一致が必要であり、ルートを選択して受け付けるかまたは拒否します。ルート ポリシーでパス属性を変更することもできます。

BGP ピアに適用するルート ポリシーを変更する場合は、そのピアの BGP セッションをリセットする必要があります。Cisco NX-OS は、BGP セッションをリセットするため、次の 3 つのメカニズムをサポートしています。

- ハード リセット：ハード リセットでは、指定されたピアリング セッションが TCP 接続を含めて切断され、指定のピアからのルートが削除されます。このオプションを使用すると、BGP ネットワーク上のパケット フローが中断します。ハード リセットは、デフォルトでディセーブルです。
- ソフト再構成着信：ソフト再構成着信によって、セッションをリセットすることなく、指定されたピアのルーティング アップデートが開始されます。このオプションを使用できるのは、着信ルート ポリシーを変更する場合です。ソフト再構成着信の場合、ピアから受け取ったすべてのルートのコピーを保存したあとで、着信ルート ポリシーを介してルートが処理されます。着信ルート ポリシーを変更する場合、Cisco NX-OS は変更された着信ルート ポリシーを介して保存ルートを渡し、既存のピアリング セッションを切断することなく、ルート テーブルをアップデートします。ソフト再構成着信の場合、まだフィルタリングされていない BGP ルートの保存に、大量のメモリ リソースを使用する可能性があります。ソフト再構成着信は、デフォルトでディセーブルです。
- ルート リフレッシュ：ルート リフレッシュでは、着信ルート ポリシーの変更時に、サポートするピアにルート リフレッシュ要求を送信することによって、着信ルーティング テー

ブルがダイナミックにアップデートされます。リモート BGP ピアは新しいルート コピーで応答し、ローカル BGP スピーカが変更されたルート ポリシーでそれを処理します。Cisco NX-OS は自動的に、プレフィックスのアウトバウンドルートの更新をピアに送信します。

- BGP ピアは、BGP ピア セッションの確立時に、BGP 機能ネゴシエーションの一部として、ルートリフレッシュ機能をアドバタイズします。ルートリフレッシュは優先オプションであり、デフォルトでイネーブルです。

（注） BGP はさらに、ルート再配布、ルート集約、ルート ダンプニングなどの機能にルート マップを使用します。ルート マップの詳細については、Route Policy Manager の設定を参照してください。

eBGP

eBGP を使用すると、異なる AS からの BGP ピアを接続し、ルーティング アップデートを交換できます。外部ネットワークへの接続によって、自分のネットワークから他のネットワークへ、またインターネットを介して、トラフィックを転送できます。

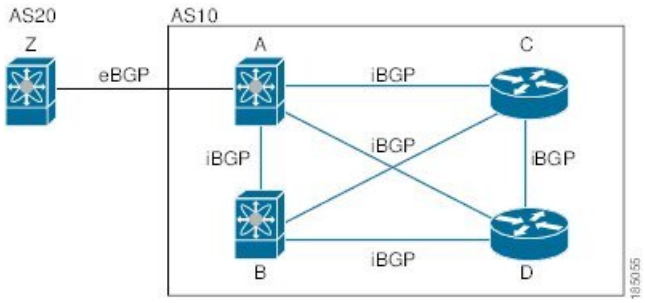
通常、eBGP ピアリングは、インターフェイスがダウンしたときにコンバージェンスが高速になるように、直接接続されたインターフェイス上で行う必要があります。

iBGP

iBGP を使用すると、同じ自律システム内の BGP ピアを接続できます。iBGP はマルチホーム BGP ネットワーク（同じ外部自律システムに対して複数の接続があるネットワーク）に使用できます。

図に、大きい BGP ネットワークの中の iBGP ネットワークを示します。

図 *3 : iBGP* ネットワーク



iBGP ネットワークはフルメッシュです。各 iBGP ピアは、ネットワーク ループを防止するために、他のすべての iBGP ピアに対して直接接続されています。

ネイバー コンフィギュレーション モードで update-source が設定された単一ホップ iBGP ピアでは、ピアは高速外部フェールオーバーをサポートします。

iBGP ピアリング セッションの確立には、ループバック インターフェイスを使用します。ループバック インターフェイスは、インターフェイス フラップが発生する可能性が小さいからです。インターフェイス フラップが発生するのは、障害またはメンテナンスが原因で、インターフェイスが管理上アップまたはダウンになったときです。マルチホップ、高速外部フォールオーバー、AS パス属性のサイズ制限については、eBGP の設定 （96 ページ） セクションを参照してください。

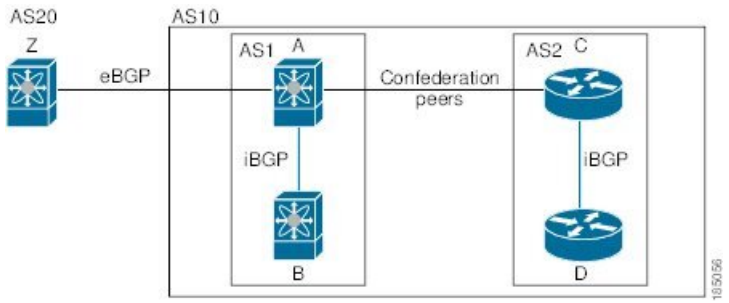
（注） iBGP ネットワークでは別個のインテリア ゲートウェイ プロトコルを設定する必要があります。

AS 連合

フルメッシュの iBGP ネットワークは、iBGP ピア数が増えるにしたがって複雑になります。自律システムを複数のサブ自律システムに分割し、それを 1 つの連合としてまとめることによって、iBGP メッシュを緩和できます。連合は、同じ自律システム番号を使用して外部ネットワークと通信する、iBGP ピアからなるグループです。各サブ AS はその中ではフルメッシュであり、同じ連合内の他のサブ AS に対する少数の接続があります。

図に BGP ネットワークが 2 つのサブ AS と 1 つの連合に分けられて表示されます。

図 *4: AS* 連合



この例では、AS10 が 2 つの AS（AS1 および AS2）に分割されています。各サブ AS はフルメッシュですが、サブ AS 間のリンクは 1 つだけです。AS 連合を使用することによって、 のフルメッシュ AS に比べて、リンク数を少なくできます。

ルート リフレクタ

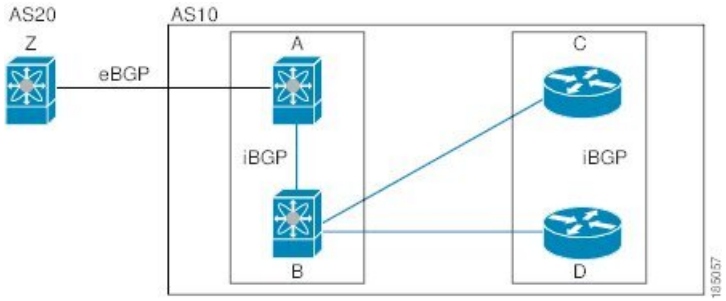
すべての iBGP ピアが完全に一致する必要がないように、ルート リフレクタが学習したルートをネイバーに渡すルート リフレクタ構成を使用することによって、iBGP メッシュを削減できます。

ある iBGP ピアをルート リフレクタとして設定すると、そのピアが iBGP で学習したルートを一連の iBGP ネイバーに渡す役割を担います。

図に、メッシュの iBGP スピーカを 4 つ（ルータ A、B、C、D）使用する、単純な iBGP 構成を示します。ルート リフレクタを使用しなかった場合、外部ネイバーからルートを受け取ったルータ A は、3 つの iBGP ネイバーのすべてにルートをアドバタイズします。

図では、ルータ B がルート リフレクタです。ルート リフレクタは、ルータ A からアドバタイズされたルートを受信すると、ルータ C と D へのルートをアドバタイズ（リフレクト）します。ルータ A は、ルータ C と D の両方にアドバタイズする必要がなくなります。

図 *5:* ルート リフレクタ



ルート リフレクタおよびそのクライアント ピアは、クラスタを形成します。ルート リフレクタのクライアント ピアとして動作するように、すべての iBGP ピアを設定する必要はありません。ただし、完全な BGP アップデートがすべてのピアに届くように、非クライアント ピアはフルメッシュとして設定する必要があります。

機能ネゴシエーション

BGP スピーカは機能ネゴシエーション機能を使用することによって、ピアでサポートされている BGP 拡張機能を学習できます。機能ネゴシエーションによって、リンクの両側の BGP ピアがサポートする機能セットだけを BGP に使用させることができます。

BGP ピアが機能ネゴシエーションをサポートしない場合で、なおかつアドレス ファミリが IPv4 として設定されている場合、Cisco NX-OS は機能ネゴシエーションを行わずに、ピアとの新規セッションを試みます。他のマルチプロトコル設定（IPv6 など）の場合は、機能ネゴシエーションが不可欠です。

ルート ダンプニング

ルート ダンプニングは、インターネットワーク上でのフラッピング ルートの伝搬を最小限に抑える BGP 機能です。ルート フラップが発生するのは、使用可能ステートと使用不能ステートが短時間で次々切り替わる場合です。

AS1、AS2、および AS3 という 3 つの BGP 自律システムからなるネットワークの場合について考えてみます。AS1 のルートがフラップした（使用不能になった）とします。ルート ダンプニングを使用しない場合、AS1 は AS2 に回収メッセージを送信します。AS2 は AS3 にその回収メッセージを伝達します。フラッピング ルートが再び発生すると、AS1 から AS2 にアドバタイズメント メッセージを送信し、AS2 は AS3 にそのアドバタイズメントを送信します。ルートの使用不能と使用可能が繰り返されると、AS1 は多数の回収メッセージおよびアドバタイズメント メッセージを送信することになり、それが他の自律システムに伝播します。

ルート ダンプニングによって、フラッピングを最小限に抑えることができます。ルート フラップが発生したとします。（ルート ダンプニングがイネーブルの）AS2 がルートにペナルティとして 1000 を割り当てます。AS2 は引き続き、ネイバーにルートの状態をアドバタイズします。ルート フラップが発生するたびに、AS2 がペナルティ値を追加します。ルート フラップ

が頻繁に発生して、ペナルティが設定可能な抑制限度を超えると、AS2 はフラップ回数に関係なく、ルートのアドバタイズを中止します。その結果、ルートが減衰（ダンプニング）します。

ルートに与えられたペナルティは、再使用限度に達するまで減衰します。その時点で、AS2 は再びルートをアドバタイズします。再使用限度が 50% になると、AS2 はそのルートのダンプニング情報を削除します。

（注） ルート ダンプニングがイネーブルの場合は、ピアのリセットによってルートが回収されても、リセット中の BGP にはペナルティは適用されません。

ロード シェアリングおよびマルチパス

BGP はルーティング テーブルに、同じ宛先プレフィックスに到達する複数の等コスト eBGP または iBGP パスを組み込むことができます。その場合、宛先プレフィックスへのトラフィックは、組み込まれたすべてのパス間で共有されます。

as-path multipath-relax コマンドを効果的に構成するには、BGP で VRF ごとに コマンドを構成します。また、複数のルータがカスタム VRF ルートターゲット（RT）にインストールされるように、カスタム VRF で as-path multipath-relax コマンドを構成します。

BGP ベストパス アルゴリズムでは、次の属性が同じ場合に、等コスト パスと見なされます。

- 重量
- ローカル プリファレンス
- AS_path
- オリジン コード
- Multi-Exit Discriminator（MED）
- BGP ネクスト ホップまでの IGP コスト

BGP はこれら複数のパスの中から、ベスト パスとして 1 つだけ選択し、そのパスを BGP ピアにアドバタイズします。詳細については、「BGP追加パス」の項を参照してください。

（注） 異なる AS 連合から受け取ったパスは、外部 AS_path 値およびその他の属性が同じ場合に、等コスト パスと見なされます。

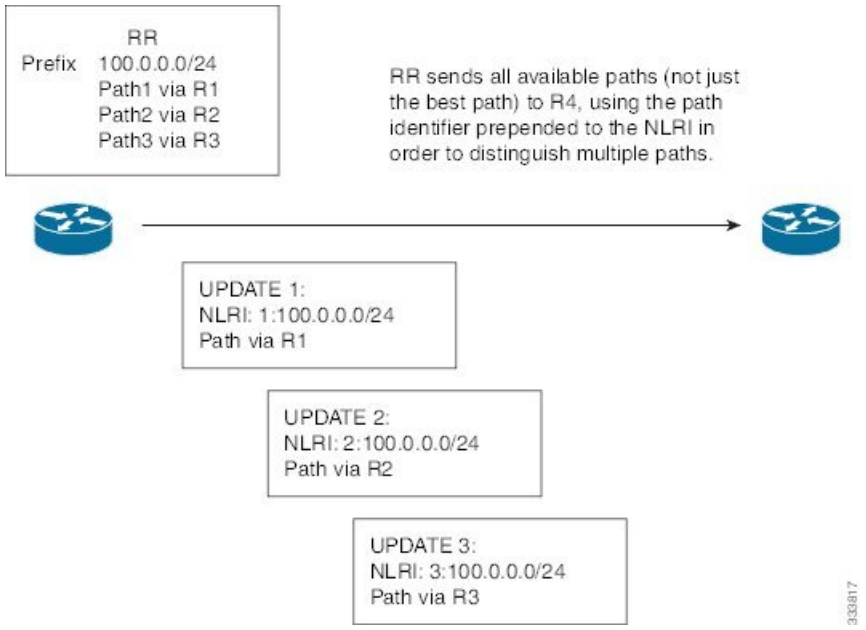
（注） iBGP マルチパスに関してルート リフレクタを設定すると、ルート リフレクタが、選択されたベスト パスをピアにアドバタイズします。そのパスのネクスト ホップは変更されません。

BGP の追加パス

1 つの BGP 最良パスだけがアドバタイズされ、BGP スピーカは特定ピアからの特定プレフィックスの 1 パスだけを受け入れます。BGP スピーカが同じセッション内で同じプレフィックスの複数のパスを受信した場合、最新のアドバタイズメントを使用します。

BGP は、以前のパスに代わる新しいパスなしで、BGP スピーカが同じプレフィックスに対して複数のパスを伝播し、受け入れることを可能にする追加のパス機能をサポートします。この機能は、BGP スピーカのピアが、プレフィックスごとの複数パスのアドバタイズおよび受信をサポートし、また、そのパスのアドバタイズをサポートするかどうかネゴシエートすることを可能にします。特別な 4 バイトのパス ID は、ピア セッションを介して送信される同じプレフィックスに対して複数のパスを区別するため、ネットワーク層到達可能性情報（NLRI）に追加されます。次の図に、追加の BGP パス機能を示します。

図 *6*：追加パスの機能を持つ *BGP* ルート アドバタイズメント



BGP 追加パス設定の詳細については、BGP 追加パスの設定 （92 ページ） の項を参照してください。

ルート集約

集約アドレスを設定できます。ルート集約を使用すると、固有性の強い一連のアドレスをすべての固有アドレスを代表する 1 つのアドレスに置き換えることによって、ルート テーブルを簡素化できます。たとえば、10.1.1.0/24、10.1.2.0/24、および 10.1.3.0/24 という固有性の強い 3 つのアドレスを 1 つの集約アドレス 10.1.0.0/16 に置き換えることができます。

アドバタイズされるルートが少なくなるように、BGP ルート テーブル内には集約プレフィックスが存在します。

（注） Cisco NX-OS は、自動ルート集約をサポートしません。

ルート集約はフォワーディング ループにつながる可能性があります。この問題を回避するために、集約アドレスのアドバタイズメントを生成するときに、BGP はローカル ルーティング テーブルに、その集約アドレスに対応するサマリー廃棄ルートを自動的に組み込みます。BGP はサマリー廃棄のアドミニストレーティブ ディスタンスを 220 に設定し、ルート タイプを廃棄に設定します。BGP はネクストホップ解決に廃棄ルートを使用しません。

ユーザが **aggregate-address** コマンドを発行すると、BGP テーブルにサマリー エントリが作成されますが、サマリーエントリは、集約のサブセットがテーブルで見つかるまでアドバタイズできません。

BGP 条件付きアドバタイズメント

BGP 条件付きアドバタイズメントを使用すると、プレフィックスが BGP テーブルに存在するかどうかに基づいてルートをアドバタイズまたは撤回するように BGP を設定できます。この機能は、たとえば、BGP でいずれかのプロバイダーにプレフィックスをアドバタイズするようなマルチホーム ネットワーク（他のプロバイダーからの情報が存在しない場合のみ）で便利です。

AS1、AS2、および AS3 という 3 つの BGP 自律システムからなるネットワークの例について考えてみます。この例で、AS1 と AS3 はインターネットと AS2 に接続しています。条件付きアドバタイズメントを使用しない場合、AS2 はすべてのルートを AS1 と AS3 の両方にプロパゲートします。条件付きアドバタイズメントを使用すれば、AS1 からのルートが存在しない場合のみ（たとえば AS1 へのリンクがダウンした場合）、特定のルートを AS3 にアドバタイズするように AS2 を設定できます。

BGP 条件付きアドバタイズメントでは、設定されたルート マップに一致する各ルートに、存在テストまたは非存在テストが追加されます。「BGP 条件付きアドバタイズメントの設定」を参照してください。

BGP ネクスト ホップ アドレス トラッキング

BGP は、インストールされているルートのネクスト ホップ アドレスをモニタして、ネクストホップの到達可能性の確認、および BGP ベストパスの選択、インストール、検証を行います。BGP ネクストホップ アドレスのトラッキングを行うと、ネクストホップの到達可能性に影響を及ぼす可能性のあるルート変更がルーティング情報ベース（RIB）で行われたときに確認プロセスをトリガーすることで、このようなネクストホップ到達可能性テストの速度が向上します。

ネクスト ホップ情報が変更されると、BGP は RIB から通知を受信します（イベント駆動型の通知）。BGP は、次のいずれかのイベントが発生したときに通知を受けます。

- ネクスト ホップが到達不能になった。
- ネクスト ホップが到達可能になった。

- ネクスト ホップへの完全再帰のインテリア ゲートウェイ プロトコル（IGP）メトリックが変更された。
- ファースト ホップの IP アドレスまたはファースト ホップのインターフェイスが変更された。
- ネクスト ホップが接続された。
- ネクスト ホップが接続解除された。
- ネクスト ホップがローカル アドレスになった。
- ネクスト ホップが非ローカル アドレスになった。

（注） 到達可能性および再帰メトリック イベントは、最適パスの再計算をトリガーします。

RIB からのイベント通知は、クリティカルおよび非クリティカルとして分類されます。クリティカルおよび非クリティカルイベントの通知は、別々のバッチで送信されます。ただし、非クリティカルイベントが保留中であり、クリティカルイベントを読み込む要求がある場合は、非クリティカル イベントがクリティカル イベントとともに送信されます。

- クリティカルなイベントとは、異なるパスに対してスイッチオーバーの原因となるネクスト ホップの消失など、ネクスト ホップの到達可能性に関連しています。異なるパスに対してスイッチオーバーの原因となるネクストホップの IGP メトリックの変更は、クリティカルなイベントと見なすことができます。
- 非クリティカルなイベントとは、最適パスに影響を与えたり、単一のネクスト ホップに IGP メトリックを変更したりせずに追加されるネクスト ホップに関連しています。

詳細については、「BGP ネクスト ホップ アドレス トラッキングの設定」を参照してください。

ルートの再配布

スタティック ルートまたは他のプロトコルからのルートを再配布するように、BGP を設定できます。再配布を指定したルート マップを設定して、どのルートが BGP に渡されるかを制御する必要があります。ルートマップを使用すると、宛先、送信元プロトコル、ルートタイプ、ルートタグなどの属性に基づいて、ルートをフィルタリングできます。詳細については、Route Policy Manager の設定を参照してください。

ルート マップを使用して両シナリオのデフォルト動作を無効にできますが、ルート マップの正しくない使用によってネットワーク ループが発生することがあるため、そうする場合は注意が必要です。次に、デフォルトの動作の変更にルート マップを使用する例を示します。

ルート マップの変更によって、シナリオ 1 のデフォルトの動作を次のように変更できます。

```
route-map foo permit 10
   match route-type internal
router ospf 1
   redistribute bgp 100 route-map foo
```

同様に、ルートマップの変更によって、シナリオ2のデフォルトの動作を次のように変更できます。

```
route-map foo deny 10
  match route-type internal
router ospf 1
   vrf bar
     redistribute bgp 100 route-map foo
```

ラベル付きユニキャスト ルートとラベルなしユニキャスト ルート

リリース7.0(3)I7(6) では、SAFI-1（ラベルなしユニキャスト）およびSAFI-4（ラベル付きユニキャスト ルーティング）が単一セッションの IPv4 BGP でサポートされるようになりました。詳細については、『*Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide*、*Release 7.x*』を参照してください。

BFD

この機能では、IPv4 および IPv6 用の双方向フォワーディング検出（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。

BGP の BFD は eBGP ピアおよび iBGP シングルホップ ピアでサポートされます。BFD を使用している iBGP シングル ホップ ピアのネイバー コンフィギュレーション モードで **update-source** オプションを設定します。

Cisco NX-OS リリース 9.3(3) 以降では、BGP の BFD は BGP IPv4 と IPv6 のプレフィックス ピアでもサポートされます。このサポートにより、BGP はマルチホップ BFD を使用できるようになり、BGP コンバージェンス時間が改善されます。プレフィックス ピアでは、シングルホップ BGP とマルチホップ BGP の両方がサポートされます。

Cisco NX-OS リリース 9.3(3) 以降、BFD は IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカルを介した BGP インターフェイス ピアリングをサポートします。ただし、BFD マルチホップはアンナンバード BGP ではサポートされません。

詳細については、『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』を参照してください。

BGP の調整

BGP タイマーによって、さらにベストパス アルゴリズムの調整によって、BGP のデフォルト動作を変更できます。

BGP タイマー

BGP では、ネイバー セッションおよびグローバル プロトコル イベントにさまざまなタイプのタイマーを使用します。確立されたセッションごとに、最低限2つのタイマーがあります。定期的にキープアライブ メッセージを送信するためのタイマー、さらに想定時間内にピアのキープアライブが届かなかった場合に、セッションをタイムアウトさせるためのタイマーです。ま

た、個々の機能を処理するための、その他のタイマーがあります。これらのタイマーは通常、秒単位で設定します。タイマーには、異なる BGP ピアで同じタイマーが異なるタイミングでスタートするように、ランダム アジャストメントが組み込まれています。

ベストパス アルゴリズムの調整

オプションの設定パラメータによって、ベストパス アルゴリズムのデフォルト動作を変更できます。たとえば、アルゴリズムでの Multi-Exit Discriminator（MED）属性およびルータ ID の扱い方を変更できます。

マルチプロトコル BGP

Cisco NX-OS の BGP は、複数のアドレス ファミリをサポートします。マルチプロトコル BGP（MP-BGP）は、アドレス ファミリに応じて異なるルート セットを伝送します。BGP ではたとえば、IPv4 ユニキャスト ルーティング用のルート セットを 1 つ、IPv4 マルチキャスト ルーティング用のルート セットを 1 つ、さらに IPv6 マルチキャスト ルーティング用のルート セットを 1 つ伝送できます。IP マルチキャスト ネットワークではリバース パス フォワーディング（RPF）のチェックに MP-BGP を使用できます。

（注） マルチキャスト BGP ではマルチキャスト状態情報をプロパゲートしないため、プロトコル独立マルチキャスト（PIM）などのマルチキャスト プロトコルが必要です。

マルチプロトコル BGP 設定をサポートするには、ルータ アドレスファミリおよびネイバー アドレス ファミリの各コンフィギュレーション モードを使用します。MP-BGP では、設定されたアドレス ファミリごとに別々の RIB が維持されます（ユニキャスト RIB と、BGP のマルチキャスト RIB など）。

マルチプロトコル BGP ネットワークは下位互換性がありますが、マルチプロトコル拡張機能をサポートしない BGP ピアは、アドレス ファミリ ID 情報など、マルチプロトコル拡張機能が伝送するルーティング情報を転送できません。

RFC 5549

BGP は RFC 5549 をサポートしており、IPv4 プレフィックスを IPv6 ネクスト ホップで伝送できます。BGP はすべてのホップで実行されるため、すべてのルータが IPv4 および IPv6 トラフィックを転送できます。したがって、ルータ間で IPv6 トンネルをサポートする必要はありません。BGP は、IPv6 ルートを介した IPv4 を Unicast Route Information Base（URIB）にインストールします。

Cisco NX-OS リリース9.2(2) 以降では、-R タイプのライン カードを搭載した Cisco Nexus 9500 プラットフォーム スイッチは、RFC 5549 をサポートします。

現在、NX-OS は IPv4 ルートの IPv6 再帰ネクストホップ（RNH）をサポートしていません。

RFC 6368

はじめに

このセクションでは、Cisco NX-OS のプロバイダー エッジ（PE）機能とカスタマー エッジ（CE）機能間で内部ボーダー ゲートウェイ プロトコル（iBGP）がどのように実装されているかについて説明します。

現在の展開で、プロバイダー/カスタマーエッジのルーティング プロトコルとしてBGPを使用すると、VPN プロバイダー自律システム（AS）とカスタマー ネットワーク自律システム間の外部ピアリングとしてピアリング セッションが設定されます。

RFC 6368では、これらのピアがiBGPピアとして設定されるようになりました。

Cisco NX-OSリリース10.1（2）以降では、EVPN-VxLANv4およびEVPN-VxLANv6のRFC 6368サポートが有効になっています。

フレームワーク

Cisco NX-OSリリース10.1（2）以降では、iBGP PE-CE機能を導入しています。

- as-override を使用した外部 Border Gateway Protocol（eBGP）を展開せずに、VRF の複数のサイトで単一の自律システム番号（ASN）を持つことができます。
- プロバイダー コアがまるで 1 つの透過ルート リフレクタ（RR）のように機能する、CE ルータへの内部ルート リフレクションを提供したいと考えます。

この機能を使用VRF サイトは、プロバイダー コアと同じ ASN を持つことができます。ただし、VRF サイトの ASN が プロバイダー コアの ASN と異なっている場合は、この機能のローカル自律システム（AS）を使用して、同じであるように表示できます。

iBGP PE-CE の実装

この機能を動作させるのは、次の 2 つの主要部分です。

- プロバイダー コアでVPN BGP 属性を透過的に伝送するために、新しい属性である `ATTR_SET` が BGP プロトコルに追加されました。
- PE ルータを、VRF 内の CE ルータへの iBGP セッションの RR にします。

新しい `ATTR_SET` 属性ではプロバイダーがカスタマーの BGP 属性すべてを透過的に伝送でき、プロバイダー属性や BGP ポリシーに干渉することがありません。こうした属性にはクラスタ リスト、ローカル設定などがあります。

BGP カスタマー ルート属性

`ATTR_SET` は、プロバイダー カスタマーの VPN BGP 属性を伝送するために使用される、新しい BGP 属性です。これは過渡的なオプション属性です。この属性では、Local Preference、Med、Origin、AS Path、Originator ID、Cluster list属性がプロバイダーネットワーク全体で伝送されます。`ATTR_SET` 属性の形式は次のとおりです。

```
+-----------------------------+
| Attr Flags□O | T□Code = 128  |
+-----------------------------+
| Attr. Length (1 or 2 octets) |
+-----------------------------+
| Origin AS (4 octets)        |
+-----------------------------+
|Path Attributes (variable)    |
+-----------------------------+
```

- 属性フラグは、通常の BGP 属性フラグです。
- 属性の長さは、この属性の長さが 1 オクテットであるか 2 オクテットであるかを示します。
- Origin AS フィールドある AS で発生するルートが、適切な `AS_PATH` 操作を行われずに、別の AS にリークされないようにします。
- 可変長-のパス属性フィールドには、プロバイダー コアで伝送されなければならない VPN BGP 属性が含まれます。

iBGP PE-CE の実装の詳細については、「iBGP PE-CE 機能の IOS 実装」を参照してください。

次に、iBGPカスタマーエッジデバイスのPEデバイスでのBGPネイバー設定の例を示します。

```
router bgp 200
vrf nxbgp3-leaf2-2
address-family ipv4 unicast
redistribute static route-map ALLOW-ALL
address-family ipv6 unicast
redistribute static route-map ALLOW-ALL
neighbor 101.101.101.101 remote-as 200
description ibgp sample config
internal-vpn-client (1)
address-family ipv4 unicast
route-reflector-client (2)
next-hop-self (3)
```

BGP モニタリング プロトコル

BGP モニタリング プロトコル（BMP）は、BGP アップデートとピア統計情報をモニタし、すべての Cisco Nexus 9000 シリーズ スイッチでサポートされます。

このプロトコルを使用して、BGP スピーカーは外部 BMP サーバに接続し、BGP イベントに関する情報を送信します。1つの BGP スピーカーに最大 2 つの BMP サーバを設定でき、各 BGP ピアは BMP サーバのすべてまたはサブセットによるモニタリング用に設定できます。BGP スピーカーは、BMP サーバからの情報を受け入れません。

グレースフル リスタートおよび高可用性

Cisco NX-OS は、BGP に対してノンストップ フォワーディングとグレースフル リスタートをサポートしています。

BGPルーティングプロトコル情報がフェールオーバー後に復元されている間に、転送情報ベース（FIB）内の既知のルートでデータ パケットを転送するように、BGPの無停止フォワーディング（NSF）を使用できます。NSF では、BGP ピアはルーティング フラップと無縁です。フェールオーバー時に、データ トラフィックはインテリジェント モジュール経由で転送され、スタンバイ スーパーバイザがアクティブになります。

Cisco NX-OSルータでコールドリブートが発生した場合、ネットワークはルータへのトラフィック転送を中止し、ネットワーク トポロジからルータを削除します。この状況では、BGP は非グレースフル リスタートになり、すべてのルートが削除されます。Cisco NX-OS がスタートアップ コンフィギュレーションを適用すると、BGPはピアリングセッションを再び確立して、ルートを再学習します。

Cisco NX-OSデュアル スーパーバイザ構成のルータでは、ステートフル スーパーバイザ スイッチオーバーが実行されます。スイッチオーバーの間、BGP は無停止フォワーディングを使用し、FIBの情報に基づいてトラフィックを転送します。システムがネットワーク トポロジから取り除かれることはありません。ネイバーが再起動しているルータは、「ヘルパー」と呼ばれます。スイッチオーバー後、グレースフル リスタート動作が開始されます。この処理が進行中の際、2つのルータはネイバー関係を再確立し、これらのBGPルートを交換します。それらネイバー関係が再起動したとしても、ヘルパーは再起動中のピアを指すプレフィックスを転送し続け、再起動中のルータはピアへトラフィックを転送し続けます。再起動中のルータがグレースフル リスタート可能なすべての BGP ピアを持つ場合、グレースフル リスタートが完了し、BGP は再び動作可能なネイバーを通知します。

BGP は、グレースフル リスタート タイマーが期限切れになる前に収束する必要があります。一時的なトラフィック損失を回避するために、ルート スケールの大きいネットワークでは、それに応じてBGP グレースフル リスタート タイマーを増やす必要があります。BGP 自体が他の BGP セッションを開くための到達可能性を提供する場合は、最初のアンダーレイ セッションがすでに収束した後にオーバーレイセッションを収束させるために必要な追加の時間に対応するため、stalepath-time も増やす必要があります。

グレースフル リスタート動作中であることがルータで検出されると、両方のルータがそれぞれのトポロジ テーブルを交換します。すべての BGP ピアからルート アップデートを受信したルータは、古いルートをすべて削除し、アップデートされたルートでベストパス アルゴリズムを実行します。

スイッチオーバーが完了すると、Cisco NX-OS は実行コンフィギュレーションを適用し、BGP は自身が再度使用可能になったことをネイバーに通知します。

ネイバー コンフィギュレーション モードで update-source が設定された単一ホップ iBGP ピアでは、ピアは高速外部フェールオーバーをサポートします。

Cisco NX-OS リリース 9.3(3) 以降、BGP プレフィックス ピアはグレースフル リスタートをサポートします。

追加 BGP パス機能により、特定のプレフィックスにアドバタイズされるパス数が再起動の前後で同じ場合、パス ID の選択は古いパスの最終状態および削除を保証します。いくつかのパスが指定されたプレフィックスにアドバタイズされる場合、古いパスがグレースフル リスタート ヘルパー ピアに発生する可能性があります。

メモリ不足の処理

BGP は、次の条件でメモリ不足に対処します。

- マイナー アラート：BGP は新しい eBGP ピアを確立しません。BGP は新しい iBGP ピアおよび連合ピアの確立は続行します。ピアは存続しますが、リセット ピアは再確立されません。
- 重大アラート：BGP は、メモリ アラートがマイナーになるまで、選択した確立済み eBGP ピアを 2 分おきにシャット ダウンします。eBGP ピアごとに、受信したパスの合計数と最適パスとして選択されたパスの数の比率が計算されます。比率が最高のピアが、メモリ使用状況を削減するためのシャットダウン対象として選択されます。オシレーションを回避するために、シャットダウンされた eBGP ピアを復帰する前にその eBGP ピアをクリアする必要があります。

（注） 重要な eBGP ピアをこの選択プロセスから除外できます。

- クリティカルアラート：BGP は確立されたすべてのピアを正常にシャットダウンします。シャットダウンされた eBGP ピアを復帰する前にその eBGP ピアをクリアする必要があります。

メモリ不足状態によるシャットダウンから BGP ピアを除外する方法の詳細については、「BGP のチューニング」を参照してください。

仮想化のサポート

1 個の BGP インスタンスを設定できます。BGP は、仮想ルーティングおよび転送（VRF）インスタンスをサポートします。

インターフェイスでの IP 転送の有効化

RFC 5549 を使用するには、少なくとも 1 つの IPv4 アドレスを設定する必要があります。IPv4 アドレスを設定しない場合は、RFC 5549 を使用するように IP転送機能を有効にする必要があります。

手順の概要

1. **configure terminal**
2. **interface** *type slot/port*
3. **ip forward**
4. （任意） **copy running-config startup-config**

手順の詳細

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 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

ステップ 3	**ip forward** 例： `switch(config-if)# ip forward`	インターフェイスにIPアドレスが設定されていない場合でも、そのインターフェイスでIPv4 トラフィックを許可します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

BGP セッション テンプレートの設定

BGP セッション テンプレートを使用すると、類似した設定が必要な複数の BGP ピアで、BGP の設定を簡素化できます。BGPテンプレートによって、共通のコンフィギュレーションブロックを再利用できます。先に BGP テンプレートを設定し、BGP ピアにテンプレートを適用します。

BGP セッション テンプレートでは、継承、パスワード、タイマー、セキュリティなどのセッション属性を設定できます。

peer-session テンプレートは、別の peer-session テンプレートからの継承が可能です。第 3 のテンプレートから継承するように第2テンプレートを設定できます。さらに最初のテンプレートもこの第3のテンプレートから継承させることができます。この間接継承を続けることができる peer-session テンプレートの数は、最大 7 つです。

ネイバーに設定した属性は、ネイバーが BGP テンプレートから継承した属性よりも優先されます。

始める前に

BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

（注）
- テンプレートを編集するときには、ピアまたはテンプレートのレベルで **no** 形式のコマンドを使用すると、テンプレートの設定を明示的に上書きできます。属性をデフォルトの状態にリセットするには、default 形式のコマンドを使用する必要があります。
- BGP ピア テンプレートを使用する場合、テンプレート内で使用されるコマンドをチェックして、そのコマンドが iBGP / eBGP ピアに適用されるかどうかを確認することはありません。たとえば、テンプレートを作成し、テンプレート内に「Remove-private-as」コマンドを追加し、このテンプレートを iBGP ピアに割り当てた場合、このコマンド「Remove-private-as」は適用されないというエラーは出力されません。 iBGP ピア。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-session** *template-name*
4. （任意） **password** *number password*
5. （任意） **timers** *keepalive hold*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **inherit peer-session** *template-name*
9. （任意） **description** *text*
10. （任意） **show bgp peer-session** *template-name*
11. （任意） **copy running-config startup-config**

手順の詳細

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** *autonomous-system-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**template peer-session** *template-name* 例：	peer-session テンプレート 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(config-router)# template peer-session BaseSession switch(config-router-stmp)#`	
ステップ 4	（任意） **password** *number password* 例： `switch(config-router-stmp)# password 0 test`	ネイバーにクリアテキストのパスワード「test」を追加します。パスワードは 3DES（タイプ 3 暗号形式）で保存および表示されます。
ステップ 5	（任意） **timers** *keepalive hold* 例： `switch(config-router-stmp)# timers 30 90`	peer-session テンプレートに BGP キープアライブおよびホールドタイマー値を追加します。 デフォルトのキープアライブ インターバルは 60 です。デフォルトのホールド タイムは 180 です。
ステップ 6	**exit** 例： `switch(config-router-stmp)# exit switch(config-router)#`	peer-session テンプレート コンフィギュレーション モードを終了します。
ステップ 7	**neighbor** *ip-address* **remote-as** *as-number* 例： `switch(config-router)# neighbor 192.168.1.2 remote-as 65535 switch(config-router-neighbor)#`	BGP ルーティング用のネイバー コンフィギュレーション モードを開始し、ネイバー IP アドレスを設定します。
ステップ 8	**inherit peer-session** *template-name* 例： `switch(config-router-neighbor)# inherit peer-session BaseSession switch(config-router-neighbor)#`	ピアに peer-session テンプレートを適用します。
ステップ 9	（任意） **description** *text* 例： `switch(config-router-neighbor)# description Peer Router A switch(config-router-neighbor)#`	ネイバーの説明を追加します。
ステップ 10	（任意） **show bgp peer-session** *template-name* 例： `switch(config-router-neighbor)# show bgp peer-session BaseSession`	peer-policy テンプレートを表示します。
ステップ 11	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。 **show bgp neighbor** コマンドを使用し、 コマンドを実行して、適用されたテンプレートを確認します。

例

BGP peer-session テンプレートを設定して、BGP ピアに適用する例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BaseSession
switch(config-router-stmp)# timers 30 90
switch(config-router-stmp)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# description Peer Router A
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

BGP peer-policy テンプレートの設定

peer-policyテンプレートを設定すると、特定のアドレス ファミリに対応する属性を定義できます。各peer-policyテンプレートにプリファレンスを割り当て、指定した順序でテンプレートが継承されるようにします。ネイバー アドレス ファミリでは最大 5 つの peer-policy テンプレートを使用できます。

Cisco NX-OS は、プリファレンス値を使用して、アドレス ファミリの複数のピア ポリシーを評価します。プリファレンス値が最小のものが最初に評価されます。ネイバーに設定した属性は、ネイバーが BGP テンプレートから継承した属性よりも優先されます。

peer-policyテンプレートでは、AS-path フィルタ リスト、プレフィックス リスト、ルート リフレクション、ソフト再構成など、アドレス ファミリ固有の属性を設定できます。

（注） **show bgp neighbor** コマンドを使用し、 コマンドを実行して、適用されたテンプレートを確認します。テンプレートで使用できる全コマンドの詳細については、『*Cisco Nexus 9000 Series NX-OS Unicast Routing Command Reference*』を参照してください。

始める前に

BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

（注） テンプレートを編集するときには、ピアまたはテンプレートのレベルで **no** 形式のコマンドを使用すると、テンプレートの設定を明示的に上書きできます。属性をデフォルトの状態にリセットするには、default 形式のコマンドを使用する必要があります。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*

3. **template peer-session** *template-name*
4. （任意） **advertise-active-only**
5. （任意） **maximum-prefix** *number*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family** {**ipv4** | **ipv6**} {**multicast** | **unicast**}
9. **inherit peer-policy** *template-name preference*
10. （任意） **show bgp peer-policy** *template-name*
11. （任意） **copy running-config startup-config**

手順の詳細

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`	コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**template peer-session** *template-name*
	例：
	`switch(config-router)# template peer-policy BasePolicy`
	`switch(config-router-ptmp)#`
		peer-policy テンプレートを作成します。

ステップ 4	（任意） **advertise-active-only** 例： `switch(config-router-ptmp)# advertise-active-only`	アクティブ ルートのみをピアにアドバタイズします。
ステップ 5	（任意） **maximum-prefix** *number* 例： `switch(config-router-ptmp)# maximum-prefix 20`	このピアに認めるプレフィックスの最大数を設定します。

ステップ 6	**exit**
	例：
	`switch(config-router-ptmp)# exit`
	`switch(config-router)#`
		peer-policy テンプレート 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 7	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65535`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー コンフィギュレーション モードを開始し、ネイバー IP アドレスを設定します。

ステップ 8	**address-family** {**ipv4** | **ipv6**} {**multicast** | **unicast**}
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		指定のアドレス ファミリに対しグローバル アドレス ファミリ設定モードを開始します。

ステップ 9	**inherit peer-policy** *template-name preference* 例： `switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1`	ピア アドレス ファミリ設定に peer-policy テンプレートを適用し、このピア ポリシーのプリファレンス値を割り当てます。
ステップ 10	（任意） **show bgp peer-policy** *template-name* 例： `switch(config-router-neighbor-af)# show bgp peer-policy BasePolicy`	peer-policy テンプレートを表示します。
ステップ 11	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。 **show bgp neighbor** コマンドを使用し、 コマンドを実行して、適用されたテンプレートを確認します。

例

BGP peer-policy テンプレートを設定して、BGP ピアに適用する例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BasePolicy
switch(config-router-ptmp)# maximum-prefix 20
switch(config-router-ptmp)# exit
switch(config-router)# neighbor 192.168.1.1 remote-as 65536
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy
switch(config-router-neighbor-af)# copy running-config startup-config
```

BGP peer テンプレートの設定

BGP peer テンプレートを設定すると、1 つの再利用可能なコンフィギュレーション ブロックで、セッション属性とポリシー属性を結合することができます。peer テンプレートも、peer-session または peer-policy テンプレートを継承できます。ネイバーに設定した属性は、ネイ

バーが BGP テンプレートから継承した属性よりも優先されます。ネイバーに設定できる peer テンプレートは 1 つだけですが、peer テンプレートは peer-session および peer-policy テンプレートを継承できます。

peer テンプレートは、eBGP マルチホップ TTL、最大プレフィックス数、ネクストホップ セルフ、タイマーなど、セッション属性およびアドレス ファミリ属性をサポートします。

始める前に

BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

（注） テンプレートを編集するときには、ピアまたはテンプレートのレベルで **no** 形式のコマンドを使用すると、テンプレートの設定を明示的に上書きできます。属性をデフォルトの状態にリセットするには、default 形式のコマンドを使用する必要があります。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer** *template-name*
4. （任意） **inherit peer-session** *template-name*
5. （任意） **address-family** {**ipv4**|**ipv6**} {**multicast**|**unicast**}
6. （任意） **inherit peer-policy** *template-name*
7. **exit**
8. （任意） **timers** *keepalive hold*
9. **exit**
10. **neighbor** *ip-address* **remote-as** *as-number*
11. **inherit peer** *template-name*
12. （任意） **timers** *keepalive hold*
13. （任意） **show bgp peer-template** *template-name*
14. （任意） **copy running-config startup-config**

手順の詳細

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	**router bgp** *autonomous-system-number* 例：	BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。

	コマンドまたはアクション	目的
	`switch(config)# router bgp 65535`	
ステップ 3	**template peer** *template-name* 例： `switch(config-router)# template peer BasePeer`	peer テンプレート コンフィギュレーション モードを開始します。
ステップ 4	（任意） **inherit peer-session** *template-name* 例： `switch(config-router-neighbor)# inherit peer-session BaseSession`	ピア テンプレートに peer-session テンプレートを適用します。

ステップ 5	（任意） **address-family** {**ipv4**|**ipv6**} {**multicast**|**unicast**}
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)`
		指定のアドレス ファミリに対しグローバル アドレス ファミリ コンフィギュレーション モードを設定します。

ステップ 6	（任意） **inherit peer-policy** *template-name* 例： `switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1`	ネイバー アドレス ファミリ設定に peer-policy テンプレートを適用します。
ステップ 7	**exit** 例： `switch(config-router-neighbor-af)# exit`	BGP ネイバー アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 8	（任意） **timers** *keepalive hold* 例： `switch(config-router-neighbor)# timers 45 100`	ピアに BGP タイマー値を追加します。 これらの値によって、peer-session テンプレート、BaseSession のタイマー値が上書きされます。
ステップ 9	**exit** 例： `switch(config-router-neighbor)# exit`	BGP 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	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65535`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー設定モードを開始し、ネイバー IP アドレスを設定します。

ステップ 11	**inherit peer** *template-name* 例：	peer テンプレートを継承します。

	コマンドまたはアクション	目的
	`switch(config-router-neighbor)# inherit peer BasePeer`	
ステップ 12	（任意） **timers** *keepalive hold* 例： `switch(config-router-neighbor)# timers 60 120`	このネイバーに BGP タイマー値を追加します。 これらの値によって、peer テンプレートおよび peer-session テンプレートのタイマー値が上書きされます。
ステップ 13	（任意） **show bgp peer-template** *template-name* 例： `switch(config-router-neighbor)# show bgp peer-template BasePeer`	peer テンプレートを表示します。
ステップ 14	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。 **show bgp neighbor** コマンドを使用し、 コマンドを実行して、適用されたテンプレートを確認します。

例

BGP peer テンプレートを設定して、BGP ピアに適用する例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer BasePeer
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer BasePeer
switch(config-router-neighbor)# copy running-config startup-config
```

プレフィックス ピアリングの設定

BGP では、IPv4 と IPv6 の両方のプレフィックスを使用してピア セットを定義できます。この機能を使用すると、各ネイバーを設定に追加する必要がありません。

プレフィックス ピアリングを定義する場合は、プレフィックスとともにリモート AS 番号を指定する必要があります。プレフィックス ピアリングが設定されている許容最大ピア数を超えない場合、BGP はプレフィックスおよび自律システムから接続するピアを受け付けます。

プレフィックス ピアリングに含まれている BGP ピアが切断されると、Cisco NX-OS は定義されているプレフィックス ピア タイムアウト値まで、ピア構造を維持します。この場合、そのプレフィックス ピアリングのすべてのスロットを他のピアが使い果たした結果、ブロックされるという危険性を伴わずに、確立されたピアのリセットまたは再接続が可能になります。

手順の概要

1. **timers prefix-peer-timeout** *value*
2. **maximum-peers** *value*

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**timers prefix-peer-timeout** *value* 例： `switch(config-router-neighbor)# timers prefix-peer-timeout 120`	ルータ コンフィギュレーション モードで BGP プレフィックス ピアリングのタイムアウト値を設定します。有効な範囲は 0 ～ 1200 秒です。デフォルト値は 30 です。 （注） プレフィックス ピアの場合は、プレフィックス ピア タイムアウトを、設定されたグレースフル リスタート タイマーよりも大きく設定します。プレフィックス ピア タイムアウトがグレースフル リスタート タイマーよりも大きければ、ピアのルートは再起動中に保持されます。プレフィックス ピア タイムアウトがグレースフル リスタート タイマーよりも小さいと、ピアのルートはプレフィックス ピア タイムアウトによって消去されます。これは、再起動が完了する前に発生する可能性があります。
ステップ 2	**maximum-peers** *value* 例： `switch(config-router-neighbor)# maximum-peers 120`	ネイバー設定モードのこのプレフィックス ピアリングの最大ピア数を設定します。範囲は 1 ～ 1000 です。

例

最大 10 のピアを受け付けるプレフィックス ピアリングの設定例を示します。

```
switch(config)# router bgp 65536
switch(config-router)# timers prefix-peer-timeout 120
switch(config-router)# neighbor 10.100.200.0/24 remote-as 65536
switch(config-router-neighbor)# maximum-peers 10
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

show bgp ipv4 unicast neighbors コマンドを使用し、すると、所定のプレフィックス ピアリングの設定の詳細とともに、現在受け付けられているインスタンスのリスト、アクティブ ピア数、最大同時ピア数、および受け付けたピアの合計数を表示できます。

IPv4 および IPv6 アドレス ファミリ向け IPv6 リンク ローカル経由の BGP インターフェイス ピアリングの設定

アンナンバード インターフェイスを使用した自動 BGP ネイバー探索のために、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカルを経由して、BGP インターフェイス ピアリングを設定できます。これにより、インターフェイス名を（インターフェイススコープのアドレスではなく）BGP ピアとして使用する BGP セッションを設定できます。この機能は、ICMPv6 ネイバー探索（ND）のルート アドバタイズメント（RA）を使用して自動ネイバー探索を行い、RFC 5549 を使用して IPv6 ネクスト ホップで IPv4 ルートを送信します。

始める前に

BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。

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`	コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム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。

ステップ 3	**neighbor** *interface-name* **remote-as** {*as-number* | **route-map** *map-name*}
	例：
	`switch(config-router)# neighbor Ethernet1/1 remote-as 65535`
	`switch(config-router-neighbor)#`
		BGP ルーティングのためにルータをネイバー設定モードにして、インターフェイスを BGP ピア用に設定します。
		（注）
		指定できるのは、イーサネット インターフェイス、ポートチャネル インターフェイス、サブインターフェイス、およびブレークアウト インターフェイスだけです。
		Cisco NX-OS リリース 9.3(6) 以降では、ルート マップを指定でき、AS リストを含められるルート マップを指定できます。ダイナミック AS 番号の使用の詳細については、「プレフィックス ピアおよびインターフェイス ピアのダイナミック *AS* 番号」を参照してください。

	コマンドまたはアクション	目的
		設定を複数のインターフェイスに適用する必要がある場合、*interface-name* は範囲にすることができます。
ステップ 4	**inherit peer** *template-name* 例： `switch(config-router-neighbor)# inherit peer PEER`	peer テンプレートを継承します。
ステップ 5	**address-family** {**ipv4** | **ipv6**} **unicast** 例： `switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#`	指定のアドレス ファミリに対しグローバル アドレス ファミリ設定モードを開始します。

ステップ 6	（任意） **show bgp** {**ipv4** | **ipv6**} **unicast neighbors** *interface*
	例：
	`switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors e1/25`
	例：
	`switch(config-router-neighbor-af)# show bgp ipv6 unicast neighbors 3FFE:700:20:1::11`
		BGP ピアに関する情報を表示します。

ステップ 7	（任意） **show ip bgp neighbors** *interface-name* 例： `switch(config-router-neighbor-af)# show ip bgp neighbors Ethernet1/1`	BGP ピアとして使用されるインターフェイスを表示します。
ステップ 8	（任意） **show ipv6 routers** [**interface** *interface*] 例： `switch(config-router-neighbor-af)# show ipv6 routers interface Ethernet1/1`	IPv6 ICMP ルータ アドバタイズメントによって学習されたリモート IPv6 ルータのリンク ローカル アドレスを表示します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカル経由で、BGP インターフェイス ピアリングを設定する例を示します。

リーフ 1 の iBGP インターフェイス ピアリング設定：

```
switch# configure terminal
switch(config)# router bgp 65000
switch(config-router)# neighbor Ethernet1/1 remote-as 65000
switch(config-router-neighbor)# inherit peer PEER
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor)# address-family ipv6 unicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

次に、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカル経由での、BGP インターフェイス ピアリングのサンプル出力例を示します。

```
switch(config-router-neighbor)# show bgp ipv4 unicast neighbors e1/15.1
BGP neighbor is fe80::2, remote AS 100, ibgp link, Peer index 4
Peer is an instance of interface peering Ethernet1/15.1
BGP version 4, remote router ID 5.5.5.5
Neighbor previous state = OpenConfirm
BGP state = Established, up for 2d16h
Neighbor vrf: default
Peer is directly attached, interface Ethernet1/15.1
Last read 00:00:54, hold time = 180, keepalive interval is 60 seconds
Last written 00:00:08, keepalive timer expiry due 00:00:51
Received 3869 messages, 0 notifications, 0 bytes in queue
Sent 3871 messages, 0 notifications, 0(0) bytes in queue
Enhanced error processing: On
0 discarded attributes
Connections established 2, dropped 1
Last reset by peer 2d16h, due to session closed
Last error length received: 0
Reset error value received 0
Reset error received major: 104 minor: 0
Notification data received:
Last reset by us never, due to No error
Last error length sent: 0
Reset error value sent: 0
Reset error sent major: 0 minor: 0
--More--
```

インターフェイス コンフィギュレーション：

次のいずれかのコマンドを使用して、対応するインターフェイスで IPv6 を有効にする必要があります。

- **ipv6 address** *ipv6-address*
- **ipv6 address use-link-local-only**
- **ipv6 link-local** *link-local-address*

```
switch# configure terminal
switch(config)# interface Ethernet1/1
switch(config-if)# ipv6 address use-link-local-only
```

（注） インターフェイスで IPv4 アドレスが設定されていない場合は、**ip forward** コマンドをインターフェイスで設定して IPv4 転送を有効にする必要があります。

（注） IPv6 ND タイマーを調整して、ネイバー探索を高速化し、BGP のルートコンバージェンスを高速化できます。

```
switch(config-if)# ipv6 nd ra-interval 4 min 3
switch(config-if)# ipv6 nd ra-lifetime 10
```

（注） Cisco NX-OS リリース 9.3(6) 以降で、パラレル リンクを使用するカスタマーの導入では、インターフェイス モードで次のコマンドを追加する必要があります。

```
switch(config-if)# ipv6 link-local use-bia
```

このコマンドは、異なるインターフェイス間での IPv6 LLA を一意にします。

BGP 認証の設定

MD5 ダイジェストを使用してピアからのルート更新を認証するように、BGP を設定できます。

または、Cisco NX-OS リリース 10.4(2)F 以降では、TCP 認証オプション（TCP AO）を使用してピアからのルート アップデートを認証するように BGP を構成できます。

Cisco NX-OS リリース 10.3(3)F 以降では、BGP パスワードのタイプ 6 暗号化が Cisco NX-OS スイッチでサポートされています。次の暗号化タイプがサポートされています。

- AES ベースの暗号化
- 秘密の暗号化と復号には、プライマリキーと呼ばれる構成可能な暗号キーが使用されます。

MD5 ダイジェストまたは TCP AO を使用するように BGP を構成するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

始める前に

- プライマリキーが Cisco NX-OS スイッチで **key config-key ascii** <*primary_key*> コマンドを使用して構成されていることを確認します。
- タイプ 6 暗号化を適切に機能させるには、Cisco NX-OS スイッチで **feature password encryption aes** が有効になっていることを確認します。
- BGP ネイバー セッション認証に TCP キーチェーン認証オプションを構成して使用するには、「TCP 認証オプションの構成」を参照してください。https://www.cisco.com/content/en/us/td/docs/dcn/nx-os/nexus9000/104x/configuration/security/cisco-nexus-9000-series-nx-os-security-configuration-guide-release-104x/chapter.html

手順の概要

1. **key config-key ascii** <*primary_key*>
2. **configure terminal**
3. **feature password encryption aes**
4. **router bgp** *AS* 番号
5. **template peer**テンプレート名
6. **password** {**0** | **3** | **7** | **6**} *string*
7. （任意） **encryption re-encrypt obfuscated**
8. （任意） **encryption delete type-6**
9. （任意） **ao** <*Keychain-name*> [**include-tcp-options**]

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**key config-key ascii** <*primary_key*> 例： `switch# key config-key ascii 0123456789012345`	プライマリキーを構成します。 （注） • このコマンドは、プライマリ キーが構成されていない場合にのみ入力します。 • プライマリ キーがすでに構成されている場合にこのコマンドを入力すると、実際には既存のプライマリ キー値が変更されます。新しい値に変更するには、プロンプトが表示されたら既存のプライマリ キー値を入力します。
ステップ 2	**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 3	**feature password encryption aes** 例： `switch(config)# feature password encryption aes`	AES パスワード暗号化を有効にします。
ステップ 4	**router bgp** *AS* 番号 例： `switch(config-router)# router bgp 1`	BGP ルータ モードを開始します。
ステップ 5	**template peer**テンプレート名 例：	BGP ネイバー 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-router-neighbor)# template peer abc`	
ステップ 6	**password** {**0** | **3** | **7** | **6**} *string* 例： `switch(config-router-neighbor)# password 6 JDYkLJd2Vh7laM/yC02kc3yTrQAc/T2lcalmKP4PHNljx/eR6wygnJWnHTpbwsQ40e1P90tnk8IIQUlsc3XGLT0A=`	MGP ネイバー セッションの MD5 パスワードを設定します。 （注） タイプ 0/タイプ 3/タイプ 7 を新しく構成する場合、プライマリ キーが構成されていて **feature password encryption aes** が有効になっている場合、タイプ 0/3/7 はタイプ 6 パスワードに自動的に暗号化されます。
ステップ 7	（任意） **encryption re-encrypt obfuscated** 例： `switch# encryption re-encrypt obfuscated`	既存のタイプ 0/タイプ 3/タイプ 7 パスワードをタイプ 6 パスワードに暗号化します。
ステップ 8	（任意） **encryption delete type-6** 例： `switch# encryption delete type-6`	タイプ 6 暗号化パスワードを削除します。
ステップ 9	（任意） **ao** <*Keychain-name*> [**include-tcp-options**]	パケットの MAC ダイジェストの計算中に TCP オプション ヘッダー（TCP AO オプション以外）を含めるかどうかを指定するオプションを構成します。

TCP 認証オプションの設定

本書では、Cisco NX-OS デバイスに TCP 認証オプションを設定する手順について説明します。

TCP 認証オプションについて

RFC 5925 で定義されている TCP 認証オプション（TCP-AO）を使用すると、より強力なメッセージ認証コード（MAC）を使用して、長期間の TCP 接続をリプレイから保護できます。

TCP-AO は、RFC 2385 で定義されている TCP MD5 の代替案です。TCP MD5 とは異なり、TCP-AO はコリジョン攻撃に対する耐性があり、アルゴリズム的俊敏性とキー管理のサポートを提供します。

TCP-AO には次のような顕著な特徴があります。

- TCP-AO は、長時間の TCP 接続のセキュリティを強化するために、より強力なメッセージ認証コード（MAC）の使用をサポートしています。
- TCP-AO は、長期的な TCP 接続のリプレイから保護し、より明示的なキー管理を提供することで、エンドポイント間のキー変更を調整します。

TCP-AO 機能により TCP MD5 は廃止されます。Cisco NX-OS デバイスは、レガシー BGP ピアの TCP-MD5 オプションを引き続きサポートします。ただし、一方の端がデバイスに TCP MD5 オプションが構成され、もう一方のピアリングに TCP-AO オプションが構成されている構成はサポートされていません。

TCP-AO キー チェーン

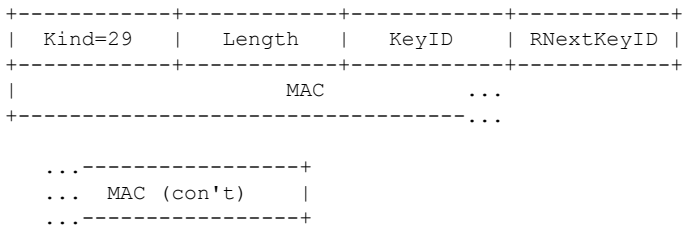
TCP-AO は、トラフィック キー、およびキーと MAC アルゴリズムを使用して生成されたメッセージ認証コード（MAC）に基づいています。トラフィック キーは、TCP-AO キー チェーンで設定できるマスター キーから導出されます。TCP-AO キー チェーンを作成し、チェーン内のキーを設定するには、グローバル コンフィギュレーション モードで **key chain** *key-chain-name* **tcp** コマンドを使用します。TCP 接続を介して通信する両方のピアで TCP-AO キー チェーンを設定する必要があります。

TCP-AO キー チェーンのキーには、次の設定可能なプロパティがあります。

設定可能なプロパティ	説明 （**Description**）
send-id	発信セグメントの TCP-AO オプションのキー識別子。 ルータで構成された送信識別子は、ピアで構成された受信識別子と一致する必要があります。
recv-id	認証時に着信セグメントの TCP-AO キー識別子と比較されるキー識別子。 ルータで構成された受信識別子は、ピアで構成された送信識別子と一致する必要があります。
cryptographic-algorithm	発信セグメントの MAC を作成するために使用される MAC アルゴリズム。アルゴリズムは次のいずれかになります。 • AES-128-CMAC 認証アルゴリズム • HMAC-SHA-1 認証アルゴリズム • HMAC-SHA-256 認証アルゴリズム

設定可能なプロパティ	説明 （Description）
include-tcp-options	このフラグは、MAC の計算に TCP-AO 以外の TCP オプションを使用するかどうかを示します。 このフラグを有効にすると、すべてのオプションの内容とゼロで埋められた認証オプションが MAC の計算に使用されます。 フラグを無効にすると、TCP-AO 以外のすべてのオプションが MAC 計算から除外されます。 このフラグはデフォルトでは無効になっています。 （注） このフラグの設定は、アプリケーション設定を使用可能にすると、アプリケーション設定によって上書きされます。
send-lifetime	この設定は、キーが有効であり、TCP セグメントの TCP-AO ベースの認証に使用できる時間を決定します。キーのライフタイムが経過し、キーが期限切れになると、ライフタイムが最も若い次のキーが選択されます。
key-string	キー文字列は、両方のピアで設定された事前共有マスターキーであり、トラフィック キーを導出するために使用されます。

TCP-AO 形式



TLV 形式のフィールドは、次のとおりです。

- Kind：TCP-AO を示す 29 という値。
- Length：TCP-AO シーケンスの長さを示します。
- KeyID：トラフィック キーの生成に使用されるマスター キー タプル（MKT）の送信識別子。
- RNextKeyID：受信したセグメントの認証に使用できる MKT の受信識別子。
- MAC：TCP セグメント データとプレフィックス付き疑似ヘッダーに対して計算された MAC。

マスター キー タプル

トラフィック キーは、個々の TCP セグメントのメッセージ認証コードを計算するために使用されるキー情報です。

マスター キー タプル（MKT）を使用すると、一意のトラフィック キーを導出し、それらのトラフィック キーの生成に必要なキー マテリアルを含めることができます。MKTは、トラフィック キーが設定されるパラメータを示します。パラメータには、TCP オプションが認証されているかどうか、そしてトラフィック キーの導出およびMAC 計算に使用されるアルゴリズムの指示子が含まれます。

各 MKT には、次の 2 つの識別子があります：

- **SendID**：**SendID** 識別子は、発信セグメントの TCP AO オプションの KeyID 識別子として挿入されます。
- **RecvID**： **RecvID は** 、着信セグメントの TCP AO キー ID と照合されます。

TCP-AO キー ロールオーバー

TCP-AO キーは、send-lifetime を使用して設定された定義済みの期間有効です。send-lifetime が設定されていない場合、キーは非アクティブと見なされます。キー ロールオーバーは、キーの送信ライフタイムに基づいて開始されます。

TCP-AO は、TCP-AO オプション フィールドの RNextKeyID および KeyID フィールドを使用して、新しいMKT の使用を調整します。ヒットレス キー ロールオーバーの場合、キーチェーン設定の新しいキーと古いキーには、少なくとも15分間のオーバーラップが必要です。これは、TCP-AO が新しい MKT の使用を調整するのに十分な時間を確保するために必要です。

キー ロールオーバーが開始されると、ピア ルータの1つ（たとえばルータ A）が、ロールオーバーが必要であることを示します。ロールオーバーが必要であることを示すために、ルータ A は使用する新しい MKT の受信識別子（recv-id）に RNextKeyID を設定します。TCP セグメントを受信すると、ピア ルータ（たとえばルータ B）は、データベースで送信識別子（send-id）を検索して、TCP-AO ペイロードの RNextKeyID によって示される MKT を見つけます。キーが使用可能で有効な場合、ルータ B は現在のキーを新しい MKT に設定します。ルータ B がロールオーバーした後、ルータ A も現在のキーを新しいプライマリ キー タプルに設定します。

送信ライフタイムと送信ライフタイムの有効期限が重複してキー ロールオーバーが開始されます。

現在のキーの有効期限が切れる前にアクティブ化できる新しいキーを設定しないと、キーがタイムアウトして期限切れになる可能性があります。このような期限が切れると、ピア ルータが期限切れのキーで認証されたセグメントを拒否し、再送信が発生することがあります。再送信タイムアウト（RTO）が原因で接続が失敗する可能性があります。新しい有効なキーが構成済みで使用可能な場合、接続を再確立することができます。

注意事項と制約事項

- キーチェーン内の各キーの send-id と recv-id は一意である必要があります。send-id とrecv-id は 0 ～ 255 の範囲から選択する必要があるため、TCP-AO キーチェーンに含められるのは最大 256 個のキーです。
- アプリケーション接続に関連付けられるキーチェーンは 1 つだけです。ロールオーバーは、常にこのキーチェーンのキー内で実行されます。
- 使用中のキーが期限切れになった場合は、有効なライフタイムを持つ新しいキーがそれぞれの側で設定されます。キーがロールオーバーするまで、セグメントの損失が予想されます。
- TCP-AO キーチェーン キーをアクティブと見なすには、send-id、recv-id、key-string、send-lifetime、および cryptographic-algorithm のすべての設定を行う必要があります。
- キー チェーン ソフトウェア プロセスでは、送信ライフタイム構成に基づいて最新のキーが使用されます。または、同じキー キーチェーンの 2 つの異なるキーに同じ send-lifetime が設定されている場合は、最後に設定されたキーを選択します。同じ送信ライフタイムを持つ 2 つのキーを設定することは、ベスト プラクティスではなく、推奨されません。
- ユーザーは、重複する 2 つのキー間の重複時間を 15 分以上に設定する必要があります。
- key-string、send-id、recv-id、cryptographic-algorithm、send-lifetime など使用中のキーの設定を変更すると、TCP 接続フラップが発生します。
- キーチェーンの設定タイプは、クライアントプロトコル内でリンクされているタイプと一致している必要があります。これらのタイプの不一致がある状態で試行されると、ユーザーに通知するための syslog メッセージが生成されます。たとえば、keychain_abc という名前のキーチェーンが Macsec キーチェーンとして設定されていても、BGP で TCP キーチェーンとして関連付けられている場合はサポートされません。同様に、キーチェーンが最初にクライアントに関連付けられ（前方参照と呼ばれるプロセス）、別のキーチェーンタイプとして設定される場合もサポートされません。

TCP キーチェーンおよびキーの設定

始める前に

- キー文字列、送信ライフタイム、暗号化アルゴリズム、およびキーの ID が両方のピアで一致することを確認します。
- ルータの送信 ID がピアルータの受信 ID と一致していることを確認します。個別のキースペースを使用する必要がある場合を除き、両方のパラメータに同じ ID を使用することをお勧めします。
- キーの送信 ID と受信 ID を同じキーチェーン内の別のキーに再利用することはできません。

- AESパスワード暗号化機能が有効になっており、プライマリ キーが構成されている場合、キーストリングは暗号化され、タイプ6形式で保存されます。それ以外の場合、パスワードはタイプ 7 暗号化形式で保存されます。
- 詳細については、「プライマリ キーの設定および AES パスワード暗号化機能のイネーブル化」を参照してください。

手順の概要

1. **configure terminal**
2. **key chain** *name* **tcp**
3. **key** *key-ID*
4. **send-id** *send-ID*
5. **recv-id** *recv-ID*
6. **key-string** *[encryption-type] text-string*
7. **[no] cryptographic-algorithm {HMAC-SHA-1 | HMAC-SHA-256 | AES-128-CMAC }**
8. **send-lifetime [local]** *start-time* **duration** [duration-value | infinite | end-time]
9. （任意） **include-tcp-options**

手順の詳細

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`	グローバル構成モードを開始します。
ステップ 2	**key chain** *name* **tcp** 例： `switch(config)# key chain bgp-keys tcp`	指定したキーチェーンのキーチェーン 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	**key** *key-ID* 例： `switch(config-tcpkeychain)# key 13`	指定したキーのキー コンフィギュレーション モードを開始します。*key-ID* 引数は、0 ～ 65535 の整数で指定する必要があります。
ステップ 4	**send-id** *send-ID* 例： `switch(config-tcpkeychain-tcpkey)# send-id 2`	キーの送信 ID を指定します。send-ID は、0 ～ 255 の範囲内で、キーチェーンごとに一意の値である必要があります。
ステップ 5	**recv-id** *recv-ID* 例： `switch(config-tcpkeychain-tcpkey)# recv-id 2`	キーの受信 ID を指定します。recv-ID は、0 ～ 255 の範囲内で、キーチェーンごとに一意の値である必要があります。

	コマンドまたはアクション	目的
ステップ 6	**key-string** *[encryption-type] text-string* 例： `switch(config-tcpkeychain-tcpkey)# key-string 0 AS3cureStr1ng`	そのキーのテキスト ストリングを設定します。text-string引数は英数字で指定します。特殊文字も使用できます。大文字と小文字は区別されます。 Encryption-type 引数に、次のいずれかの値を指定します。 • 0：入力した text-string 引数は、暗号化されていないテキスト文字列です。これがデフォルトです。 • 6：Cisco NX-OS リリース 10.3(3)F 以降、Cisco Nexus 9000 シリーズ プラットフォーム スイッチでシスコ独自の（タイプ 6 暗号化）方式がサポートされています。 • 7：入力した text-string 引数は、暗号化されています。シスコ固有の暗号方式で暗号化されます。このオプションは、別の Cisco NX-OS デバイス上で実行した **show key chain** コマンドの暗号化出力に基づいて、テキスト文字列を入力する場合に役立ちます。 **key-string** コマンドには、*text-string* での次の特殊文字の使用に関する制限があります。

特殊文字	説明（Description）	注
|	縦棒またはパイプ	キー文字列の先頭ではサポートされていません
>	右辺と比較して大きい	キー文字列の先頭ではサポートされていません
\	バックスラッシュ	キー文字列の先頭または末尾ではサポートされていません
（	左丸かっこ	キー文字列の先頭ではサポートされていません
'	アポストロフィ	キー文字列の先頭ではサポートされていません
"	引用符	キー文字列の先頭ではサポートされていません

<table>
<tr><th></th><th>コマンドまたはアクション</th><th>目的</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>特殊文字</th><th>説明（Description）</th><th>注</th></tr>
<tr><td>?</td><td>疑問符</td><td>サポート。ただし、疑問符（?）を入力する前に Ctrl+V を押します。</td></tr>
</table>
コマンドでの特殊文字の使用方法の詳細については、「コマンドラインインターフェイスについて」セクションを参照してください。
</td></tr>
<tr><td>ステップ 7</td><td>[no] cryptographic-algorithm {HMAC-SHA-1 | HMAC-SHA-256 | AES-128-CMAC }
例：
<code>switch(config-tcpkeychain-tcpkey)# cryptographic-algorithm HMAC-SHA-1</code></td><td>TCP セグメントの MAC の計算に使用するアルゴリズムを指定します。1 つのキー に設定できる暗号化アルゴリズムは 1 つだけです。</td></tr>
<tr><td>ステップ 8</td><td>send-lifetime [local] start-time duration [duration-value | infinite | end-time]
例：
<code>switch(config-tcpkeychain-tcpkey)# send-lifetime local 01:01:01 Jan 01 2023 01:01:01 Jan 10 2023</code></td><td>キーの送信ライフタイムを設定します。デフォルトでは、デバイスは start-time および end-time 引数を UTC として扱います。local キーワードを指定すると、デバイスはこれらの時間を現地時間として扱います。
start-time 引数は、キーがアクティブになる日時です。
送信ライフタイムの終了時は次のいずれかのオプションで指定できます。
• duration duration-value：ライフタイムの長さ（秒）。最大値は 2147483646 秒（約 68 年）です。
• infinite：キーの送信ライフタイムは期限切れになりません。
• end-time：end-time 引数はキーがアクティブでなくなる日時です。</td></tr>
<tr><td>ステップ 9</td><td>（任意） include-tcp-options
例：
<code>switch(config-tcpkeychain-tcpkey)# include-tcp-options</code></td><td>パケットの「MAC」ダイジェストを計算中に TCP ヘッダー（TCP AO オプション以外）の一部の「TCP オプション」全体を含める必要があるかどうかを指定するためのオプションの構成です。</td></tr>
</table>

TCP キーチェーンの確認

コマンド	目的
show key chain *[name] [detail]*	デバイスに設定されているキーチェーンを表示します。

```
switch# show key chain
Key-Chain bgp_keys tcp
  Key 2 -- text 7 "070e234f"
    send-id 2
    recv-id 2
    cryptographic-algorithm AES_128_CMAC
    send lifetime UTC (08:17:00 May 29 2023)-(08:21:00 May 29 2023)
    include-tcp-options
  Key 3 -- text 7 "070c2058"
    send-id 3
    recv-id 4
    cryptographic-algorithm HMAC-SHA-1
    send lifetime UTC (08:20:00 May 29 2023)-(always valid) [active]
    include-tcp-options
  Key 12 -- text ""
    send lifetime UTC (08:20:00 May 29 2023)-(always valid)
```

（注） [active] は、キーが有効でアクティブであることを示します。それ以外の場合、キーは非アクティブです。上記の例では、キー 3 のみがアクティブで使用可能です。

show key chain detail コマンドは、アクティブなキーと非アクティブなキーを明示的に表示します。タイプ 6 暗号化の場合、show key chain detail コマンドを実行すると、タイプ 6 キー文字列が復号化可能かどうかも表示されます。また、クライアントがパケットを認証するために現在使用している最も新しいアクティブな送信キーも表示されます。

```
switch# show key chain detail
Key-Chain bgp_keys tcp
  Key 1 -- text 6 "JDYk9k4kmaciqaH6Eu2+9C0tmCRl9k7JAMYs/fXGbW1lmHP88PAA=="
    Type6 Decryptable: yes
    send-id 1
    recv-id 1
    cryptographic-algorithm HMAC-SHA-1
    send lifetime local (18:15:42 May 15 2023)-(always valid) [active]
    include-tcp-options
    accept-ao-mismatch
  Key 2 -- text 6 "JDYkB+Fs8u3ujRDpFSu4tH6H7iTS45JJA6sKeGsBD0L3HjGDeg9AA=="
    Type6 Decryptable: yes
    send-id 2
    recv-id 2
    cryptographic-algorithm AES_128_CMAC
    send lifetime local (17:10:47 May 15 2023)-(18:15:42 May 15 2023) [inactive]

  youngest active send key: 1
```

TCP キーチェーンの構成例

bgp_keys という名前の TCP キーチェーンを設定する例を示します。各キー テキスト ストリングは暗号化されています。キーのライフタイム設定は重複しています。

```
key chain bgp_keys tcp
```

```
key 1
  send-id 1
  recv-id 1
  key-string 7 070e234f
  send-lifetime 01:00:00 Oct 10 2023 01:00:00 Oct 11 2023
  cryptographic-algorithm AES-128-CMAC
key 2
  send-id 2
  recv-id 2
  key-string 7 075e731f
  send-lifetime 00:45:00 Oct 11 2023 01:00:00 Oct 12 2023
  cryptographic-algorithm HMAC-SHA-256
  include-tcp-options
```

Resource Public Key Infrastructure（RPKI）

RPKIは、BGP（インターネット）プレフィックスを認証済みの送信元AS番号にマッピングする情報を含む、グローバルに配布されたデータベースです。BGPパスの送信元ASを検証するために、BGPを実行しているルータは、RPKIに接続できます。

RPKI-Cache-to-Router接続は多対多にすることができ、1つのRPKIキャッシュは複数のルーターにorigin-AS検証データを提供でき、1つのルーターは複数のRPKIキャッシュに接続できます。ルーターはRPKIキャッシュに接続して情報をダウンロードし、BGPがインターネットルーティング テーブルの発信元AS番号を検証するために使用できる特別なRPKIデータベースを構築します。

RPKIデータベースは、BGPが接続するさまざまなRPKIキャッシュから集約されたRoute-Origin-Attestation（ROA）オブジェクトのセットです。ROAオブジェクトは、BGPプレフィックスブロックと、そのブロックの発信を許可されたAS番号との間のマッピングを提供します。

Cisco NX-OSリリース10.6(1)F以降、ユーザーは、ASのeBGPルータでorigin-AS検証機能を設定し、発信元検証状態拡張コミュニティを使用して、独自の管理下でeBGPピアに検証状態を通知できます。

RPKI構成

RPKI構成は次のように分類されます。

- RPKIキャッシュに接続するためのコマンド。
- 受信プレフィックスにRPKI検証状態をマークするためのコマンド。
- BGPベストパス計算でRPKI検証状態を使用するためのコマンド。
- route-mapを使用して特定の検証状態を持つプレフィックスを削除または操作するためのコマンド。

RPKIキャッシュに接続するためのコマンド

RPKIキャッシュ構成は、router-bgpサブモードの新しいrpki-cacheサブモードで行います。これは、デフォルトのVRFでのBGPピアの構成に似ています。サブモードに入るには、「rpki

cache <IP address>」コマンドを使用します。サブモードに入ると、RPKI キャッシュのさまざまなパラメータを構成できます。

```
router bgp 100
 rpki cache 147.28.0.11
    description        A description to identify the cache
    shutdown           Shutdown the cache
    transport tcp port Transport port on which cache is listening
    vrf                Vrf in which RPKI cache is reachable
    refresh-interval   Specify periodic wait time between cache poll attempts
    retry-interval     Specify wait time before retrying failed serial or reset query
    expiry-interval    Specify how long to use current data while unable to perform
successful query
```

（注） トランスポート TCP ポートが明示的に構成されていない限り、BGP は RPKI-RTR ポート 323 で RPKI キャッシュへ接続します。

明示的に設定されていない限り、すべての間隔は、データ PDU の末尾の RPKI キャッシュによって提案されたとおりに決定されます。

受信プレフィックスを RPKI 検証状態でマークするためのコマンド

RPKI プレフィックス検証処理の動作を制御するためのノブがあります。これらのノブは、アドレス ファミリ レベルで構成できます。

- **origin-as validate**：アドレス ファミリ レベルで構成すると、ROA データベースに対する eBGP パス検証が有効になります。デフォルトでは無効になっています。

（注） このコマンドは、iBGP パスには関係ありません。iBGP パスは、ROA データベースに対して検証されません。iBGP パスでパス検証状態をマークする唯一の方法は、BGP プレフィックス発信元検証状態拡張コミュニティを受信することであり、コマンドを構成せずにデフォルトで実行されます。

- **origin-as validate signal ibgp**：アドレス ファミリ レベルで構成すると、BGP プレフィックス発信元検証状態拡張コミュニティを介した検証状態の iBGP シグナリングが有効になります。
- **origin-as validate signal ebgp**：アドレス ファミリ レベルで構成すると、すべての eBGP ピアへの BGP プレフィックス発信元検証状態拡張コミュニティを介した検証状態の eBGP シグナリングが有効になります。

  ネイバー アドレス ファミリ レベルで構成すると、BGP プレフィックス発信元検証状態拡張コミュニティを介したその特定の eBGP ピアへの eBGP 検証状態のシグナリングを有効にします。

- **origin-as validate accept ebgp**：アドレス ファミリ レベルで構成すると、BGP プレフィックス発信元検証状態拡張コミュニティを介した eBGP ピアからの検証状態の承諾が有効になります。

BGP 最適パス計算で RPKI 検証状態を使用するためのコマンド

RPKI プレフィックス検証処理の動作を制御するためのコマンドがあります。これらのコマンドは、アドレス ファミリ レベルで構成できます。

- **bestpath origin-as use-validity**：アドレス ファミリ レベルで構成することで、BGP ベストパス処理でのパスのプリファレンスに影響する BGP パスの有効性状態を有効にします。デフォルトでは無効になっています。
- **bestpath origin-as allow invalid**：アドレス ファミリ レベルで構成することで、すべての「無効な」パスが BGP 最適パス計算のために考慮されるようにします（best-path origin-as 検証が設定されている場合、そのようなパスはどれも最適パス候補ではありません）。デフォルトでは無効になっています。

route-mapを使用して特定の検証状態を持つプレフィックスを削除または操作するためのコマンド

以下は、ルートマップを使用して特定の検証状態を持つプレフィックスを削除または操作するためのコマンドです。

```
route-map sample1 permit 10
  match rpki {not-found | invalid | valid}
```

match rpki コマンドのパラメータは次のとおりです。

- `not-found`：この origin-AS は RPKI データベースでは不明です。
- `invalid`：RPKI データベース内の無効な origin-AS です。

  `valid`：RPKI データベース内の有効な origin-AS です。

この match 句は、インバウンド ルートマップにのみ関連します。

iBGP で学習されたパスの場合、更新の入力 BGP プレフィックス発信元検証状態拡張コミュニティが、このルートマップ句と比較されます。

eBGP 学習パスの場合、ROA データベース ルックアップによって取得された検証状態が、このルートマップ句と比較されます。

検証状態が無効であるとマークされたプレフィックスは、BGP での最適パスの計算に考慮されないため、無効になりますが、管理者は、システム メモリを節約するために、そのようなプレフィックスを完全に削除するように決定する場合があります。この目的には、次のインバウンド ルート マップが推奨されます。

```
route-map sample deny 10
match rpki invalid
route-map sample permit 20
```

RPKI Show コマンド

RPKI 構成情報を表示するには、次のいずれかのタスクを行います。

コマンド	目的
show bgp rpki summary	RPKI キャッシュの数を含む RPKI 統計情報の概要を表示します。
show bgp rpki table {**ipv4** | **ipv6**} {**IP address**/**masklength**}	現在の RPKI ROA データベースに関する情報を表示します。オプションを指定しなかった場合、コマンドは IPv4 ROA データベースを表示します。IPv6 オプション（show bgp rpki table ipv6）を指定すると、このコマンドは IPv6 ROA データベースを表示します。（接続の問題などにより）一時的にダウンしているキャッシュから受信した ROA は（*）で表示されます。キャッシュセッションがそのキャッシュのパージ時間内に確立されない場合、これらの ROA は RPKI データベースから削除されます。 table show コマンドの後に ROA プレフィックスブロックが指定されている場合（たとえば、show bgp rpki table 67.21.36.0/24 max 24）、その特定の ROA エントリが詳細に表示されます（ROA が存在する場合）。 （注） 1 つの ROA（IP アドレス/最小-最大）は、複数のオリジン AS を持つことができ、複数のキャッシュからソースを取得できます。
show bgp rpki cache {**IP address**}	構成されているすべてのキャッシュとそのパラメータ（**show bgp summary** など）の要約リストを表示します。 前のコマンドでキャッシュ IP アドレスが指定されている場合、そのキャッシュの詳細情報が表示されます。
show bgp {**ipv4 unicast** | **ipv6 unicast**} **origin-as validity-state** {**valid** | **invalid** | **unknown**}	BGP ピアに関する情報を表示します。このコマンドには、パス（validation_state）に基づいて BGP テーブル出力をフィルタリングする新しいオプションがあります。このコマンドで有効性状態（有効、無効、または不明）を指定すると、BGP テーブルから関連情報がフィルタリングされ、その有効性状態に一致する BGP パスのみが表示されます。

RPKI Clear コマンド

以下は RPKI Clear コマンドです。

- **clear bgp rpki cache *** - このコマンドは、構成されているすべての RPKI キャッシュのトランスポート セッションをリセットし、すべてのキャッシュから受信したすべての IPv4 および IPv6 ROA の RPKI データベースを即座に消去します。

RPKI Debug および Event History コマンド

以下は、RPKI Debug および Event History コマンドです。

- **debug bgp rpki** - このコマンドは、プレフィックス検証を除くすべての RPKI 関連操作のデバッグをオンにします。 これには、RPKI キャッシュ接続、RPKI キャッシュのプロトコル ステート マシン、ROA の挿入や削除などの RPKI データベース イベントなどのデバッグ イベントが含まれます。
- **sh bgp event-history rpki** - このコマンドは、RPKI に関する高レベルの情報をダンプします。

BGP セッションのリセット

BGP のルート ポリシーを変更した場合は、関連付けられた BGP ピア セッションをリセットする必要があります。BGP ピアがルート リフレッシュをサポートしない場合は、着信ポリシー変更に関するソフト再構成を設定できます。Cisco NX-OS は自動的に、セッションのソフト リセットを試みます。

ソフト再構成着信を設定するには、ネイバー アドレス ファミリ設定モードで次のコマンドを使用します。

手順の概要

1. **soft-reconfiguration inbound**
2. **clear bgp ipv4** {**unicast** | **multicast** *ip-address* **soft** {**in** | **out**}
3. （任意） **clear bgp** {**ipv4** | **ipv6** } {**unicast** | **multicast** *ip-address* **soft** {**in** | **out**}
4. **clear bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} *ip-address* **soft** (**in** | **out**)

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**soft-reconfiguration inbound** 例： `switch(config-router-neighbor-af)# soft-reconfiguration inbound`	着信 BGP ルート アップデートを格納するために、ソフト再構成をイネーブルにします。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。

	コマンドまたはアクション	目的
ステップ 2	**clear bgp ipv4** {**unicast** | **multicast** *ip-address* **soft** {**in** | **out**} 例： `switch# clear bgp ip unicast 192.0.2.1 soft in`	このモードは任意のモードで、TCP セッションを切断しないで、BGP セッションをリセットします。
ステップ 3	（任意） **clear bgp** {**ipv4** | **ipv6** } {**unicast** | **multicast** *ip-address* **soft** {**in** | **out**} 例： `switch# clear bgp ip unicast 192.0.2.1 soft in`	TCP セッションを切断しないで、BGP セッションをリセットします。
ステップ 4	**clear bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} *ip-address* **soft** (**in** | **out**) 例： `switch# clear bgp ip unicast 192.0.2.1 soft in`	TCP セッションを切断しないで、BGP セッションをリセットします。

ネクストホップ アドレスの変更

次の方法で、ルートアドバタイズメントで使用するネクストホップアドレスを変更できます。

- ネクストホップ計算をディセーブルにして、ローカル BGP スピーカ アドレスをネクストホップ アドレスとして使用します。
- ネクストホップ アドレスをサードパーティ アドレスとして設定します。この機能は、元のネクスト ホップ アドレスがルートの送り先のピアと同じサブネット上にある場合に使用します。この機能を使用すると、フォワーディング時に余分なホップを節約できます。

ネクストホップ アドレス トラッキングを変更するには、アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **next-hop-self**
2. **next-hop-third-party**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**next-hop-self** 例：	ルート アップデートのネクストホップ アドレスとして、ローカル BGP スピーカ アドレスを使用します。このコマンドによって、BGP ネイバー セッショ

	コマンドまたはアクション	目的
	`switch(config-router-neighbor-af)# next-hop-self`	ンの自動ソフト クリアまたはリフレッシュが開始されます。
ステップ 2	**next-hop-third-party** 例： `switch(config-router-neighbor-af)# next-hop-third-party`	ネクストホップ アドレスをサードパーティ アドレスとして設定します。このコマンドは、**next-hop-self** が設定されていないシングル ホップの EBGP ピアに使用します。 configured.

BGP ネクスト ホップ アドレス トラッキングの設定

BGP ネクスト ホップ アドレス トラッキングはデフォルトで有効であり、無効にすることができません。

BGP ネクスト ホップ トラッキングのパフォーマンスを向上するために、RIB チェック間の遅延インターバルを変更できます。

BGP ネクスト ホップ アドレス トラッキングを変更するには、アドレス ファミリ設定モードで次のコマンドを使用します。

手順の概要

1. **nexthop trigger-delay** {**critical** | **non-critical**} *milliseconds*

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**nexthop trigger-delay** {**critical**	**non-critical**} *milliseconds* 例： `switch(config-router-af)# nexthop trigger-delay critical 5000`	クリティカルなネクスト ホップの到達可能性ルートおよび非クリティカルなルートについて、ネクスト ホップ アドレス トラッキングの遅延タイマーを指定します。指定できる範囲は 1 ～ 4294967295 ミリ秒です。クリティカル タイマーのデフォルトは 3000 です。非クリティカル タイマーのデフォルトは 10000 です。

ネクスト ホップ フィルタリングの設定

BGP ネクストホップ フィルタリングを使用すると、RIB でネクストホップ アドレスがチェックされるときにそのネクストホップ アドレスの基盤となるルートがルート マップを経由します。ルート マップでそのルートが拒否されると、ネクストホップ アドレスは到達不能として扱われます。

BGP は、ルート ポリシーによって拒否されたすべてのネクストホップを無効であるとマークし、無効なネクストホップ アドレスを使用するルートについてベスト パスを計算しません。

BGP ネクストホップ フィルタリングを設定するには、アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **nexthop route-map** *name*

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**nexthop route-map** *name* 例： `switch(config-router-af)# nexthop route-map nextHopLimits`	BGP ネクストホップ ルートが一致するルート マップを指定します。63 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

デフォルト ルートによるネクストホップ解決の設定

BGP ネクストホップ解決では、IP デフォルト ルートを BGP ネクストホップ解決に使用するかどうかを指定できます。

BGP ネクストホップ解決を設定するには、ルータ設定モードで次のコマンドを使用します。

手順の概要

1. [**no**] **nexthop suppress-default-resolution**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	[**no**] **nexthop suppress-default-resolution** 例： `switch(config-router)# nexthop suppress-default-resolution`	IP デフォルト ルートを介した BGP ネクストホップの解決を防止します。 このコマンドを有効にすると、以下のようになります。 • **show bgp process detail** コマンドの出力には、次の行が含まれます。 Use default route for nexthop Resolution：No

	コマンドまたはアクション	目的
		• **show routing clients bgp** コマンドの出力には、次の行が含まれます。 Owned rnh will never resolve to 0.0.0.0/0

ネクストホップセルフによるリフレクト ルートの制御

NX-OS では、**next-hop-self** [all] 引数を使用して特定のピアに送信する際の iBGP ルートを制御できます。これらの引数を使用すると、ルートのリフレクトが実施されている場合でも、ルートのネクストホップを選択的に変更できます。

コマンド	目的
next-hop-self [all] 例： `switch(config-router-af)# next-hop-self all`	ルートアップデートのネクストホップアドレスとして、ローカルBGPスピーカアドレスを使用します。 all キーワードはオプションです。all を指定すると、すべてのルートが next-hop-self を使用するピアに送信されます。all を指定しなかった場合、リフレクトしたルートのネクストホップは変更されません。

セッションがダウンした場合のネクストホップ グループの縮小

セッションがダウンしたときに迅速な方法で ECMP グループを縮小するように BGP を設定できます。

この機能は、次の BGP パス障害イベントに適用されます。

- 1 つまたは複数のレイヤ 3 リンクの障害
- ライン カード障害
- BGP ネイバーの BFD 障害検出
- BGP ネイバーの管理上のシャットダウン（shutdown コマンドを使用）

最初の 2 つのイベント（レイヤ 3 リンク障害とラインカード障害）の迅速な処理はデフォルトでイネーブルになっており、イネーブルにするための設定コマンドは必要ありません。

最後の 2 つのイベントの迅速な処理を設定するには、ルータ設定モードで次のコマンドを使用します。

手順の概要

1. **neighbor-down fib-accelerate**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**neighbor-down fib-accelerate** 例： `switch(config-router)# neighbor-down fib-accelerate`	BGP セッションがダウンするたびに、すべてのネクストホップ グループ（ECMP グループと単一のネクストホップ ルート）から対応する次のネクスト ホップを取り消します。 （注） このコマンドは、IPv4 ルートと IPv6 ルートの両方に適用されます。

機能ネゴシエーションの無効化

機能ネゴシエーションをディセーブルにすると、機能ネゴシエーションをサポートしない古い BGP ピアとの相互運用が可能です。

機能ネゴシエーションをディセーブルにするには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **dont-capability-negotiate**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**dont-capability-negotiate** 例： `switch(config-router-neighbor)# dont-capability-negotiate`	機能ネゴシエーションをディセーブルにします。このコマンドの設定後、BGP セッションを手動でリセットする必要があります。

ポリシーのバッチ処理の無効化

プレフィックスに一意の属性がある BGP 展開では、BGP は、同じ BGP アップデートメッセージでバンドルする類似の属性を持つルートを識別しようとします。この追加の BGP 処理のオーバーヘッドを回避するには、バッチ処理をディセーブルにします。

固有のネクスト ホップを持つ多数のルートがある BGP 展開では、ポリシーバッチ処理を無効にすることを推奨します。

ポリシー バッチ処理を無効にするには、ルータ設定モードで次のコマンドを使用します。

手順の概要

1. **disable-policy-batching**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**disable-policy-batching** 例： `switch(config-router)# disable-policy-batching`	すべてのピアへのプレフィックスアドバタイズメントのバッチ評価をディセーブルにします。

BGP 追加パスの設定

BGPは、プレフィックスごとの複数パスの送受信と、このパスのアドバタイジングをサポートします。

追加パスの送受信機能のアドバタイズ

BGP ピア間の追加パスの送受信機能をアドバタイズするように BGP を設定できます。これを行うには、ネイバー アドレス ファミリ設定モードで次のコマンドを使用します。

手順の概要

1. [**no**] **capability additional-paths send** [**disable**]
2. [**no**] **capability additional-paths receive** [**disable**]
3. **show bgp neighbor**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	[**no**] **capability additional-paths send** [**disable**] 例： `switch(config-router-neighbor-af)# capability addtional-paths send`	BGP ピアに追加パスを送信する機能をアドバタイズします。**disable** オプションは、追加パス送信機能のアドバタイズをディセーブルにします。 このコマンドの **no** 形式を使用すると、追加パスの送信機能がディセーブルになります。

	コマンドまたはアクション	目的
ステップ 2	[**no**] **capability additional-paths receive** [**disable**] 例： `switch(config-router-neighbor-af)# capability addtional-paths receive`	BGP ピアから追加パスを受信する機能をアドバタイズします。**disable** オプションは、追加パス受信機能のアドバタイズをディセーブルにします。 このコマンドの **no** 形式は、追加パスの受信機能をディセーブルにします。
ステップ 3	**show bgp neighbor** 例： `switch(config-router-neighbor-af)# show bgp neighbor`	ローカル ピアがリモート ピアへの追加パス送受信機能をアドバタイズしたかを表示します。

例

BGP ピアに追加のパスを送受信する機能をアドバタイズする BGP の設定例を示します。

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# neighbor 10.131.31.2 remote-as 100
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# capability additional-paths send
switch(config-router-neighbor-af)# capability additional-paths receive
```

追加パスの送受信の設定

BGP ピア間の追加パスの送受信機能を設定できます。これを行うには、アドレス ファミリ設定モードで次のコマンドを使用します。

手順の概要

1. [**no**] **additional-paths send**
2. [**no**] **additional-paths receive**
3. **show bgp neighbor**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	[**no**] **additional-paths send** 例： `switch(config-router-af)# additional-paths send`	機能が無効になっていないこのアドレス ファミリで、すべてのネイバーの追加パスの送信機能を有効にします。 このコマンドの **no** 形式を使用すると、送信機能が無効になります。

	コマンドまたはアクション	目的
ステップ 2	[**no**] **additional-paths receive** 例： `switch(config-router-af)# additional-paths receive`	機能が無効になっていないこのアドレス ファミリで、すべてのネイバーの追加パスの受信機能を有効にします。 このコマンドの **no** 形式を使用すると、受信機能が無効になります。
ステップ 3	**show bgp neighbor** 例： `switch(config-router-af)# show bgp neighbor`	ローカル ピアがリモート ピアへの追加パス送受信機能をアドバタイズしたものとして表示します。

例

機能が無効になっていない指定されたアドレス ファミリで、すべてのネイバーの追加パスの受信機能を有効にする例を示します。

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths send
switch(config-router-af)# additional-paths receive
```

アドバタイズされるパスの設定

BGP にアドバタイズされたパスを指定できます。これを行うには、ルートマップ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. [**no**] **set ip next-hop unchanged**
2. [**no**] **set path-selection { all | backup | best2 | multipaths}** | advertise
3. **show bgp** {**ipv4** | **ipv6**} **unicast** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*]

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	[**no**] **set ip next-hop unchanged** 例： `switch(config-route-map)# set ip next-hop unchanged`	不変のネクストホップ 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 2	[**no**] **set path-selection { all | backup | best2 | multipaths}** | advertise 例： `switch(config-route-map)# set path-selection all advertise`	すべてのパスが指定されたプレフィックスにアドバタイズされるように指定します。次のいずれかのオプションを選択できます。 • all：使用可能なすべての有効なパスをアドバタイズします。 • backup：バックアップ パスとしてマークされたパスをアドバタイズします。このオプションでは、additional-path install backup コマンドを使用してバックアップ パスを有効にする必要があります。 • best2：2 番目に最適なパスをアドバタイズします。これは、すでに計算されているベストパスを除き、残りの使用可能なパスのベストパスです。 • multipaths：すべてのマルチパスをアドバタイズします。このオプションでは、maximum-paths コマンドを使用してマルチパスを有効にする必要があります。 （注） マルチパスがない場合、backup オプションと best2 オプションは同じです。マルチパスがある場合、best2 はマルチパスのリストの最初のパスで、バックアップは計算されたベスト パスとマルチパスを除くすべての使用可能なパスのベストパスです。 このコマンドの **no** 形式は、最適パスだけがアドバタイズされるように指定します。
ステップ 3	**show bgp** {**ipv4** | **ipv6**} **unicast** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*] 例： `switch(config-route-map)# show bgp ipv4 unicast`	プレフィックスの追加パスのパス ID とこれらのパスのアドバタイズメント情報を表示します。

例

すべてのパスがプレフィックス リスト p1 にアドバタイズされるよう指定する例を示します。

```
switch# configure terminal
switch(config)# route-map PATH_SELECTION_RMAP
```

```
switch(config-route-map)# match ip address prefix-list p1
switch(config-route-map)# set path-selection all advertise
```

追加パス選択の設定

プレフィックスに追加のパスを選択する機能を設定できます。これを行うには、アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. [**no**] **additional-paths selection route-map** *map-name*
2. {| } [ip-address | ipv6-prefix] [vrf-name] **show bgpipv4ipv6unicastvrf**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	[**no**] **additional-paths selection route-map** *map-name* 例： `switch(config-router-af)# additional paths selection route-map map1`	プレフィックスに追加のパスを選択する機能を設定します。 このコマンドの **no** 形式は、追加パス選択機能をディセーブルにします。
ステップ 2	{| } [ip-address	ipv6-prefix] [vrf-name] **show bgpipv4ipv6unicastvrf** 例： `switch(config-route-af)# show bgp ipv4 unicast`	プレフィックスの追加パスのパス ID とこれらのパスのアドバタイズメント情報を表示します。

例

指定されたアドレス ファミリで追加パス選択を設定する例を示します。

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths selection route-map PATH_SELECTION_RMAP
```

eBGP の設定

eBGP シングルホップ チェックの無効化

シングルホップ eBGP ピアがローカル ルータに直接接続されているかどうかのチェック機能を無効にするように、eBGP を設定できます。このオプションは、直接接続されたスイッチ間のシングルホップ ループバック eBGP セッションの設定に使用します。

シングルホップ eBGP ピアが直接接続されているかどうかのチェックを無効にするには、ネイバー設定モードで次のコマンドを使用します。

手順の概要

1. **disable-connected-check**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**disable-connected-check** 例： `switch(config-router-neighbor)# disable-connected-check`	シングルホップ eBGP ピアが直接接続されているかどうかのチェックを無効にします。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。

TTL セキュリティ ホップの構成

IP パケット ヘッダーの TTL 値が BGP ネイバー セッション用に設定された TTL 値以上の場合のみ BGP がセッションを確立または維持できるようにするには、次の作業を実行します。

始める前に

TTL セキュリティ チェックに対する BGP サポート機能の効果を最大化するために、参加している各ルータでこの機能を設定することを推奨します。この機能を有効にすると、eBGP セッションが受信方向だけ保護され、送信IPパケットまたはリモートルータは影響を受けません。

（注）
- TTL セキュリティ チェックに対する BGP サポート機能がマルチホップ ネイバー セッション用に構成されている場合、**neighbor ebgp-multihop** コマンドは必要なく、この機能を構成する前にこのコマンドを無効にする必要があります。
- 大きい直径のマルチホップ ピアリングでは、TTL セキュリティ チェックに対する BGP サポート機能の効果は下がります。大きい直径のピアリング用に設定された BGP ルータに対する CPU 利用率に基づく攻撃の場合は、影響を受けたネイバー セッションをシャットダウンして、この攻撃に対処する必要がある場合があります。
- この機能は、ローカル ネットワークおよびリモート ネットワーク内部が損なわれているピアからの攻撃には効果的ではありません。この制約事項には、ローカル ネットワークとリモート ネットワークの間のネットワーク セグメント上のピアも含まれます。

手順の概要

1. **enable**

2. **trace** *[protocol] destination*
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **neighbor** *ip-address*
6. **ttl-security hops** *hop-count*
7. **end**
8. **show running-config**
9. **show ip bgp neighbors** *[ip-address]*

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `switch(config)# enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**trace** *[protocol] destination* 例： `switch(config)# trace ip 10.1.1.1`	パケットが宛先に移動中、実際に通過する指定されたプロトコルのルートを検出します。 trace コマンドを入力して、指定されたピアへのホップ カウントを決定します。
ステップ 3	**configure terminal** 例： `switch(config)# 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 4	**router bgp** *autonomous-system-number* 例： `switch(config)# router bgp 65000`	ルータ コンフィギュレーションモードを開始して、BGP ルーティング プロセスを作成します。
ステップ 5	**neighbor** *ip-address* 例： `switch(config)# neighbor 10.1.1.1`	ネイバー IP アドレスを構成します。
ステップ 6	**ttl-security hops** *hop-count* 例： `switch(config)# ttl-security hops 2`	2 つのピアを区切るホップの最大数を設定します。 hop-count 引数は、ローカル ピアとリモート ピアを区切るホップ カウントに設定されます。IP パケット ヘッダーの予想される TTL 値が 254 の場合、数値 1 を hop-count 引数に設定する必要があります。値の範囲は、1 ～ 254 の数番です。 TTL セキュリティ チェックに対する BGP サポート機能が有効な場合、BGP は、予想値以上の TTL 値

	コマンドまたはアクション	目的
		を持つ着信 IP パケットを受け入れます。受け入れられないパケットは廃棄されます。 この設定例では、予想される着信 TTL 値が 253（255 引く TTL 値の 2）以上に設定されます。これは、BGP ピアから予想される最小 TTL 値です。ローカル ルータは、10.1.1.1 ネイバーが 1 または 2 ホップ離れている場合だけ、このネイバーからのピアリング セッションを受け入れます。
ステップ 7	**end** 例： `switch(config)# end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 8	**show running-config** 例： `switch(config)# show running-config | begin bgp`	（任意）現在実行中のコンフィギュレーション ファイルの内容を表示します。 このコマンドの出力は、各ピアの neighbor ttl-security コマンドの構成を出力の BGP コンフィギュレーション セクションの下に表示します。そのセクションには、ネイバー アドレスおよび構成されたホップ カウントが含まれます。 （注） この例では、このタスクに適用可能な構文だけが使用されています。詳細については、『Cisco IOS IP Routing: BGP Command Reference』を参照してください。
ステップ 9	**show ip bgp neighbors** *[ip-address]* 例： `switch(config)# show ip bgp neighbors 10.4.9.5`	（任意）ネイバーへの TCP 接続および BGP 接続の情報を表示します。 このコマンドは、TTL セキュリティ チェックに対する BGP サポート機能が有効になっている場合、「External BGP neighbor may be up to number hops away」と表示します。この number 値は、ホップ カウントを表します。これは、1 ～ 254 の数値です。 （注） この例では、このタスクに適用可能な構文だけが使用されています。詳細については、『Cisco IOS IP Routing: BGP Command Reference』を参照してください。

eBGP マルチホップの設定

eBGP マルチホップをサポートする eBGP 存続可能時間（TTL）値を設定できます。eBGP ピアは状況によって、別の eBGP ピアに直接接続されず、リモート eBGP ピアに到達するために複数のホップを必要とします。ネイバーセッションに eBGP TTL 値を設定すると、このようなマルチホップ セッションが可能になります。

（注） この設定は、BGP インターフェイス ピアリングではサポートされません。

eBGP マルチホップを設定するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **ebgp-multihop** *ttl-value*

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**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。

高速外部フォールオーバーの無効化

Cisco Nexus 7000 シリーズ デバイスは、すべての VRF のネイバーおよびアドレス ファミリ（IPv4 または IPv6）の高速外部フォールオーバーをデフォルトでサポートします。

Cisco NX-OS デバイスは、すべての VRF のネイバーおよびアドレス ファミリ（IPv4 または IPv6）の高速外部フォールオーバーをデフォルトでサポートします。通常、BGP ルータと直接接続 eBGP ピア間の接続が失われると、ピアとの eBGP セッションをリセットすることによって、BGP が高速外部フォールオーバーを開始します。この高速外部フォールオーバーをディセーブルにすると、リンク フラップが原因の不安定さを制限できます。

高速外部フォールオーバーをディセーブルにするには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **no fast-external-fallover**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**no fast-external-fallover** 例： `switch(config-router)# no fast-external-fallover`	eBGP ピアの高速外部フォールオーバーをディセーブルにします。このコマンドは、デフォルトでイネーブルになっています。

AS パス属性の制限

AS パス属性で自律システム番号が高いルートを廃棄するように eBGP を設定できます。

AS パス属性で AS 番号の多いルートを廃棄するには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **maxas-limit** *number*

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**maxas-limit** *number* 例： `switch(config-router)# maxas-limit 50`	AS パス セグメントの番号が指定された上限を超えている eBGP ルートを廃棄します。範囲は 1 ～ 512 です。

ローカル AS サポートの設定

ローカル AS 機能では、ルータが実際の AS に加えて、2 番めの自律システム（AS）のメンバであるように見せることができます。ローカル AS を使用すると、ピアリングの調整を変更せずに 2 つの ISP をマージできます。マージされた ISP 内のルータは、新しい自律システムのメンバになりますが、使用者に対しては古い自律システム番号を使用し続けます。

この機能は、正しい eBGP ピアにしか使用できません。別のコンフェデレーションのサブ自律システムのメンバである 2 ピアに対しては、この機能は使用できません。

さらに、remote-asコマンドで設定されたリモートピアのASNは、local-asコマンドで設定されたローカルデバイスのASNと同一にすることはできません。

eBGP ローカル AS のサポートを設定するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **local-as** *number* [**no-prepend** [**replace-as** [**dual-as**]]]

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**local-as** *number* [**no-prepend** [**replace-as** [**dual-as**]]] 例： `switch(config-router-neighbor)# local-as 1.1`	AS_PATH 属性にローカル AS の *number* を付加するよう eBGP を設定します。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。

例

次に、VRF のローカル AS サポートを設定する例を示します。

```
switch# configure terminal
switch(config)# router bgp 1
switch(config-router)# vrf test
switch(config-router-vrf)# local-as 1
switch(config-router-vrf)# show running-config bgp
```

AS 連合の設定

AS連合を設定するには、連合識別情報を指定する必要があります。AS連合内の自律システムグループは、自律システム番号として連合 ID を持つ、1 つの自律システムとして外部で認識されます。

BGP 連合 ID を設定するには、ルータ設定モードで次のコマンドを使用します。

手順の概要

1. **confederation identifier** *as-number*
2. **bgp confederation peers** *as-number* [*as-number2...*]

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**confederation identifier** *as-number* 例：	ルータ設定モードで、このコマンドは BGP 連合 ID を設定します。

	コマンドまたはアクション	目的
	`switch(config-router)# confederation identifier 4000`	このコマンドによって、BGP ネイバーセッションの自動通知およびセッション リセットが開始されます。
ステップ 2	**bgp confederation peers** *as-number* [*as-number2...*] 例： `switch(config-router)# bgp confederation peers 5 33 44`	ルータ設定モードで、このコマンドは AS 連合に属する自律システムを設定します。 このコマンドは、連合に属する自律システムのリストを指定し、BGP ネイバーセッションの自動通知とセッションリセットをトリガーします。

ルート リフレクタの設定

ルート リフレクタとして動作するローカル BGP スピーカに対するルート リフレクタ クライアントとして、iBGP ピアを設定できます。ルート リフレクタとそのクライアントがともにクラスタを形成します。クライアントからなるクラスタには通常、ルート リフレクタが 1 つ存在します。このような状況では、ルート リフレクタのルータ ID でクラスタを識別します。ネットワークの冗長性を高め、シングル ポイント障害を回避するために、複数のルート リフレクタからなるクラスタを設定できます。クラスタ内のすべてのルート リフレクタは、同じ 4 バイト クラスタ ID で設定する必要があります。これは、ルート リフレクタが同じクラスタ内のルート リフレクタからのアップデートを認識できるようにするためです。

始める前に

BGPをイネーブルにする必要があります。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **cluster-id** *cluster-id*
4. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
5. （任意） **client-to-client reflection**
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
9. **route-reflector-client**
10. （任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
11. （任意） **copy running-config startup-config**

手順の詳細

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	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**cluster-id** *cluster-id* 例： `switch(config-router)# cluster-id 192.0.2.1`	クラスタに対応するルート リフレクタの 1 つとして、ローカル ルータを設定します。クラスタを識別するクラスタ ID を指定します。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。

ステップ 4	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		指定のアドレス 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	（任意） **client-to-client reflection** 例： `switch(config-router-af)# client-to-client reflection`	クライアント間のルート リフレクションを設定します。この機能は、デフォルトでイネーブルになっています。このコマンドによって、BGP ネイバー セッションの自動ソフトクリアまたはリフレッシュが開始されます。

ステップ 6	**exit**
	例：
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		ルータ アドレス コンフィギュレーション モードを終了します。

ステップ 7	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.0.2.10 remote-as 65535`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの IP アドレスおよび AS 番号を設定します。

	コマンドまたはアクション	目的

ステップ 8	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		ユニキャスト IPv4 アドレス ファミリに対応するネイバー アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 9	**route-reflector-client** 例： `switch(config-router-neighbor-af)# route-reflector-client`	BGP ルート リフレクタとしてデバイスを設定し、そのクライアントとしてネイバーを設定します。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。
ステップ 10	（任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** 例： `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	BGP ピアを表示します。
ステップ 11	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、ルート リフレクタとしてルータを設定し、クライアントとしてネイバーを 1 つ追加する例を示します。

```
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.10 remote-as 65536
switch(config-router-neighbor)# address-family ip unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# copy running-config startup-config
```

アウトバウンド ルート マップを使用した、反映されたルートのネクスト ホップの設定

アウトバウンド ルート マップを使用して、BGP ルート リフレクタの反映されたルートのネクスト ホップを変更できます。ネクスト ホップ アドレスとしてピアのローカル アドレスを指定するため、アウトバウンド ルート マップを設定できます。

（注） この項で説明している **next-hop-self** コマンドは、ルート リフレクタによってクライアントに反映されるルートに対してこの機能を有効にしません。この機能は、アウトバウンド ルート マップを使用した場合にだけ有効にできます。

始める前に

BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

正しいVDCを使用していることを確認します（または**switchto vdc**コマンドを使用します）。

set next-hop を入力する必要がありますコマンドを入力して、アドレス ファミリ固有のネクスト ホップ アドレスを設定する必要があります。たとえば、IPv6アドレスファミリの場合は、**set ipv6 next-hop peer-address** コマンドを入力する必要があります。

- ルートマップを使用してIPv4 ネクストホップを設定する場合： **set ip next-hop peer-address** がルートマップと一致する場合、ネクストホップはピアのローカルアドレスに設定されます。ネクスト ホップがルート マップで設定されていない場合、ネクスト ホップはパスに保存されているネクスト ホップに設定されます。
- ルートマップを使用してIPv6ネクストホップを設定する場合： **set ipv6 next-hop peer-address** がルートマップと一致する場合、ネクスト ホップは次のように設定されます。
  - IPv6 ピアでは、ネクスト ホップはピアのローカル IPv6 アドレスに設定されます。
  - IPv4 ピアの場合、 **update-source** が設定されている場合、ネクスト ホップは、該当する場合、発信元インターフェイスの IPv6 アドレスに設定されます。IPv6 アドレスが設定されていない場合、ネクスト ホップは設定されません。
  - IPv4 ピアの場合、 **update-source** が設定されていない場合、ネクスト ホップは、該当する場合、送信先インターフェイスの IPv6 アドレスに設定されます。IPv6 アドレスが設定されていない場合、ネクスト ホップは設定されません。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. （任意） **update-source** *interface number*
5. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
6. **route-reflector-client**
7. **route-map** *map-name* **out**
8. （任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} [**ip-address** | **ipv6-prefix**] **route-map** *map-name* [**vrf** *vrf-name*]
9. （任意） **copy running-config startup-config**

手順の詳細

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** *as-number*
	例：
	`switch(config)# router bgp 200`
	`switch(config-router)#`
		BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.0.2.12 remote-as 200`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの IP アドレスおよび AS 番号を設定します。

ステップ 4	（任意） **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 5	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		指定のアドレス 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 6	**route-reflector-client** 例： `switch(config-router-neighbor-af)# route-reflector-client`	BGP ルートリフレクタとしてデバイスを設定し、そのクライアントとしてネイバーを設定します。このコマンドによって、BGP ネイバーセッションの自動通知およびセッション リセットが開始されます。
ステップ 7	**route-map** *map-name* **out** 例： `switch(config-router-neighbor-af)# route-map setrrnh out`	発信ルートに設定された BGP ポリシーを適用します。
ステップ 8	（任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} [**ip-address** | **ipv6-prefix**] **route-map** *map-name* [**vrf** *vrf-name*] 例：	ルートマップと一致する BGP ルートを表示します。

	コマンドまたはアクション	目的
	`switch(config-router-neighbor-af)# show bgp ipv4 unicast route-map setrrnh`	
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

アウトバウンド ルート マップを使用して、BGP ルート リフレクタの反映されたルートのネクスト ホップを設定する例を示します。

```
switch(config)# interface loopback 300
switch(config-if)# ip address 192.0.2.11/32
switch(config-if)# ipv6 address 2001::a0c:1a65/64
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# exit
switch(config)# route-map setrrnh permit 10
switch(config-route-map)# set ip next-hop peer-address
switch(config-route-map)# exit
switch(config)# route-map setrrnhv6 permit 10
switch(config-route-map)# set ipv6 next-hop peer-address
switch(config-route-map)# exit
switch(config)# router bgp 200
switch(config-router)# neighbor 192.0.2.12 remote-as 200
switch(config-router-neighbor)# update-source loopback 300
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# route-map setrrnh out
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# route-map setrrnhv6 out
```

ルート ダンプニングの設定

iBGP ネットワーク上でのルート フラップの伝播を最小限に抑えるために、ルート ダンプニングを設定できます。

ルート ダンプニングを設定するには、アドレス ファミリまたは VRF アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **dampening** [{*half-life reuse-limit suppress-limit max-suppress-time* | **route-map** *map-name*}]

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**dampening** [{*half-life reuse-limit suppress-limit max-suppress-time* | **route-map** *map-name*}] 例： `switch(config-router-af)# dampening route-map bgpDamp`	機能ネゴシエーションをディセーブルにします。パラメータ値は次のとおりです。 • *half-life*：指定できる範囲は 1 ～ 45 です。 • *resuse-limit*指定できる範囲は 1 ～ 20000 です。 • *suppress-limit*：指定できる範囲は 1 ～ 20000 です。 • *max-suppress-time*：指定できる範囲は 1 ～ 20000 です。

ロード シェアリングおよび ECMP の設定

等コスト マルチパス ロード バランシング用に BGP がルート テーブルに追加するパスの最大数を設定できます（EXMP）。

パスの最大数を設定するには、ルータ アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **maximum-paths** [**ibgp**] *maxpaths*

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**maximum-paths** [**ibgp**] *maxpaths* 例： `switch(config-router-af)# maximum-paths 8`	ロード シェアリング用の等コスト パスの最大数を設定します。デフォルトは 1 です。

BGP 経由不等コストマルチパス（UCMP）

UCMP は加重 ECMP とも呼ばれます。これは、ネクスト ホップごとに異なる重みを持つ、同じ宛先への複数のルートを許可し、ルーティングされたトラフィックをそれらの複数のネクス

トホップにロードバランシングするメカニズムです。基本的な UCMP は、ほとんどの顧客の要件に対応します。負荷エントロピーは、リンク使用効率を最大化する最良の方法です。

多くの場合、ネットワーク内のアプリケーションの分散は不均衡になりがちです。新しいクラスタは、古いクラスタとは異なるオーバーサブスクリプション率でロールインします。新しいクラスタには、古いクラスタよりも強力なサーバーがあり、CPU ごとにより多くの負荷を処理できます。ネットワークは完全ではないため、ルーティング動作をある程度制御する必要があります。トラフィックの負荷を分散し、ルーティング動作の制御を管理するために、BGP 経由の加重 ECMP を構成できます。

（注） リンク帯域幅拡張コミュニティは、非推移的な属性として定義されていますが、eBGP セッション全体でアドバタイズする必要があります。

Next-hop-self は、アドバタイズから Link-Bandwidth Extended Community を取り除く必要があります。

UCMP over BGP の有効化

ユースケースでリソースの不均等な分散と最適ではないトラフィック分散が発生している場合の解決策は、BGP 上で重み付き ECMP を構成することです。各インスタンスの重みは、（ホストまたはコントローラーから）ルートを挿入して通知できます。その後、インフラストラクチャ全体の重みを集計し、アプリケーション展開の分布に比例するようにトラフィックを配信できます。

BGP 経由 UCMP の注意事項と制限事項

- BGPは、draft-ietf-idr-link-bandwidth-06.txtで定義されているリンク帯域幅拡張コミュニティを使用して、重み付けECMP機能を実装します。リンク帯域幅拡張コミュニティは、次ホップが変更されていない限り、非推移的な属性として定義されていますが、eBGP セッション全体でアドバタイズされます。
- iBGP ピアと eBGP ピアの両方からリンク帯域幅拡張コミュニティを受け入れることができます。
- 重み付けプログラミングの場合、リンク帯域幅拡張コミュニティには、RIB にダウンロードする前に 0 ～ 1000 の間で正規化された 4 バイトの浮動小数点整数としてバイト/秒でエンコードされたリンク帯域幅があります。
- ハードウェア ECMP 幅は 64 サイズに固定されています。

最大プレフィックス数の設定

BGP が BGP ピアから受け取ることのできるプレフィックスの最大数を設定できます。任意で、プレフィックス数がこの値を超えた場合に、BGP に警告メッセージを生成させる、またはピアとの BGP セッションを切断させることを設定できます。

BGP ピアに認めるプレフィックスの最大数を設定するには、ネイバー アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **maximum-prefix** *maximum* [*threshold*] [**restart** *time* | **warning-only** | **discard**]

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**maximum-prefix** *maximum* [*threshold*] [**restart** *time* | **warning-only** | **discard**] 例： `switch(config-router-neighbor-af)# maximum-prefix 12`	ピアからのプレフィックスの最大数を設定します。パラメータの範囲は次のとおりです。 • *maximum*：指定できる範囲は 1 ～ 300000 です。 • *threshold*：指定できる範囲は 1 ～ 100 % です。デフォルトは 75% です。 • *time*：指定できる範囲は 1 ～ 65535 分です。 • *discard*：NX-OS リリース 10.6(2)F で導入されました。しきい値に達した後、プレフィックスを破棄します。 このコマンドによって、プレフィックス限度を超えた場合に、BGP ネイバーセッションの自動通知およびセッション リセットが開始されます。

DSCP の設定

ネイバーの differentiated services code point（DSCP）を設定します。IPv4 または IPv6 のローカル発信パケットの DSCP 値を指定できます。

DSCP 値を設定するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **dscp** *dscp_value*

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**dscp** *dscp_value*
	例：
	`switch(config-router-neighbor)# dscp 63`
	次に、対応する **show** コマンドの例を示します。
	`show ipv6 bgp neighbors`
	`BGP neighbor is 10.1.1.1, remote AS 0, unknown link, Peer index 4`
	`  BGP version 4, remote router ID 0.0.0.0`
	`  BGP state = Idle, down for 00:13:34, retry in 0.000000`
	`  DSCP (DiffServ CodePoint): 0`
	`  Last read never, hold time = 180, keepalive interval is 60 seconds`
		ネイバーの Differentiated Services Code Point（DSCP）の値を設定します。DSCP 値には、0 ～ 63 の数字、または、**ef**、**af11**、**af12**、**af13**、**af21**、**af22**、**af23**、**af31**、**af32**、**af33**、**af41**、**af42**、**af43**、**cs1**、**cs2**、**cs3**、**cs4**、**cs5**、**cs6**、または **cs7** のいずれかのキーワードを指定できます。
		デフォルト値は cs6 です。

ダイナミック機能の設定

BGP ピアのダイナミック機能を設定できます。

ダイナミック機能を設定するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **dynamic-capability**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**dynamic-capability** 例： `switch(config-router-neighbor)# dynamic-capability`	ダイナミック機能をイネーブルにします。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。

集約アドレスの設定

BGP ルート テーブルの集約アドレス エントリを設定できます。

集約アドレスを設定するには、ルータ アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **aggregate-address** *ip-prefix/length* [**as-set**] [**summary-only**] [**advertise-map** *map-name*] [**attribute-map** *map-name*] [**suppress-map** *map-name*]

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**aggregate-address** *ip-prefix/length* [**as-set**] [**summary-only**] [**advertise-map** *map-name*] [**attribute-map** *map-name*] [**suppress-map** *map-name*] 例： `switch(config-router-af)# aggregate-address 192.0.2.0/8 as-set`	集約アドレスを作成します。このルートに関してアドバタイズされるパスは、集約されているすべてのパスに含まれるすべての要素からなる、自律システム セットです。 • **as-set** キーワードは、関係するパスから自律システム セット パス情報およびコミュニティ情報を生成します。 • **summary-only** キーワードは、アップデートから具体的なルートをすべてフィルタリングします。 • **advertise-map** キーワードおよび引数では、選択されたルートから属性情報を選択するためのルート マップを指定します。 • **attribute-map** キーワードおよび引数では、集約から属性情報を選択するためのルートマップを指定します。 • **suppress-map** キーワードおよび引数によって、固有性の強いルートを条件付きでフィルタリングします。BGP ルート集約の実行中に **suppress-map** オプションを指定すると、BGP ルート更新のコミュニティ属性を設定できます。このオプションを使用すると、より具体的なルートにコミュニティ属性を設定できます。 • **suppress-map** キーワードおよび引数によって、固有性の強いルートを条件付きでフィルタリングします。BGPルート集約の実行中に **suppress-map** オプションを指定すると、特定のより具体的なルートがピアにアドバタイズされないように抑制したり、suppress-map route-map

	コマンドまたはアクション	目的
		設定に応じて、いくつかのコミュニティ属性が設定されたより具体的なルートをアドバタイズしたりすることができます。match 句だけで設定されたルートマップは、一致基準を満たすより具体的なルートを抑制します。ただし、ルートマップが match および set 句で設定されている場合、一致基準を満たすルートは、ルートマップによって変更された適切な属性でアドバタイズされます。2 番目のオプションでは、より具体的なルートにコミュニティ属性を設定できます。

BGP ルートの抑制

新しく学習されたBGPルートが転送情報ベース（FIB）により確認され、ハードウェアでプログラミングされた後にのみ、これらのルートをアドバタイズするようにCisco NX-OSを設定できます。ルートがプログラミングされた後は、これらのルートに対する以降の変更にはこのハードウェア プログラミングのチェックは必要ありません。

BGP ルートを抑制するには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

手順の概要

1. **suppress-fib-pending**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**suppress-fib-pending** 例： `switch(config-router)# suppress-fib-pending`	新しく学習された BGP ルート（IPv4 または IPv6）がハードウェアでプログラミングされるまで、ダウンストリームのBGPネイバーにアドバタイズされることを抑制します。

BGP 条件付きアドバタイズメントの設定

BGP がプロパゲートするルートを制限するように BGP 条件付きアドバタイズメントを設定できます。次の 2 つのルート マップを定義します。

- アドバタイズ マップ：BGP が条件付きアドバタイズメントを考慮する前にルートが一致する必要のある条件を指定します。このルートマップには、適切なmatch文を含めることができます。
- 存在マップまたは非存在マップ：BGP がアドバタイズ マップに一致するルートをプロパゲートする前に BGP テーブルに存在する必要のあるプレフィックスを定義します。非存在マップは、BGP がアドバタイズ マップに一致するルートをプロパゲートする前に BGP テーブルに存在してはならないプレフィックスを定義します。BGP は、これらのルートマップでプレフィックス リストの match 文内にある permit 文のみを処理します。
- Nexus は、条件付きルート アドバタイズメントを使用した他の BGP 属性変更操作（AS パスを付加する例）をサポートしていません。存在/非存在マップの構成に基づいてアドバタイズされるルートを制御するために使用されます。

ルートが条件を渡さない場合、そのルートが BGP テーブルにあれば BGP によってルートが取り消されます。

始める前に

BGP を有効にする必要があります（「BGP の有効化」のセクションを参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
5. **advertise-map** *adv-map* {**exist-map** *exist-rmap*|**non-exist-map** *nonexist-rmap*}
6. （任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
7. （任意） **copy running-config startup-config**

手順の詳細

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)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP モードを開始し、ローカル BGP スピーカに自律システム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 3	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65534`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー設定モードを開始し、ネイバー IP アドレスを設定します。

ステップ 4	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
	例：
	`switch(config-router-neighbor)# address-family ipv4 multicast`
	`switch(config-router-neighbor-af)#`
		アドレス ファミリ設定モードを開始します。

ステップ 5	**advertise-map** *adv-map* {**exist-map** *exist-rmap*|**non-exist-map** *nonexist-rmap*} 例： `switch(config-router-neighbor-af)# advertise-map advertise exist-map exist`	2 つの設定済みルート マップに従い、ルートを条件付きでアドバタイズするように BGP を設定します。 • *adv-map*：BGP がルートを次のルート マップに渡す前に、そのルートが渡す必要のある **match** 文を含むルート マップを指定します。*adv-map* 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。 • *exist-rmap*：プレフィックス リストの match ステートメントを使用してルートマップを指定します。BGP テーブル内のプレフィックスは、BGP がルートをアドバタイズする前に、プレフィックスリスト内のプレフィックスと一致する必要があります。*exist-rmap* 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。 • *nonexist-rmap*：プレフィックス リストの match ステートメントを使用してルートマップを指定します。BGP テーブル内のプレフィックスは、BGP がルートをアドバタイズする前に、プレフィックスリスト内のプレフィックスと一致してはいけません。*nonexist-rmap* 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） BGP 条件付きアドバタイズメント機能の場合、exist マップまたは nonexist マップに関連付けられている場合、プレフィックスリストで「le」または「ge」ステートメントが使用されていないことを確認します。

	コマンドまたはアクション	目的
ステップ 6	（任意） **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** 例： `switch(config-router-neighbor-af)# show ip bgp neighbor`	BGP に関する情報、および設定した条件付きアドバタイズメントのルートマップに関する情報を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、BGP 条件付きアドバタイズメントを設定する例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.2 remote-as 65537
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# advertise-map advertise exist-map exist
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# exit
switch(config)# route-map advertise
switch(config-route-map)# match as-path pathList
switch(config-route-map)# exit
switch(config)# route-map exit
switch(config-route-map)# match ip address prefix-list plist
switch(config-route-map)# exit
switch(config)# ip prefix-list plist permit 209.165.201.0/27
```

ルートの再配布の設定

別のルーティング プロトコルからのルーティング情報を受け入れて、BGP ネットワークを通じてその情報を再配布するように、BGP を設定できます。任意で、再配布ルートのためのデフォルト ルートを割り当てることができます。

始める前に

BGP 機能が有効になっていることを確認します

- BGP を有効にする必要があります（「BGP の有効化」の項を参照）。
- 正しい VDC を使用していることを確認します（または **switchto vdc** コマンドを使用します）。

BGPを有効にする必要があります。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family ipv4** {**unicast** | **multicast**}
4. **address-family** {**ipv4** | **ipv6** } {**unicast** | **multicast**}
5. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
6. **redistribute** {**direct**| {**eigrp** |**ospf** | **ospfv3** |**rip**} *instance-tag* | **static**} **route-map** *map-name*
7. **redistribute** {**direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static** | **icmpv6**} **route-map** *map-name*
8. （任意） **default-metric** *value*
9. （任意） **copy running-config startup-config**

手順の詳細

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** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**address-family ipv4** {**unicast**	**multicast**}
		例：
		`switch(config-router)# address-family ipv4 unicast`
		`switch(config-router-af)#`
			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	**address-family** {**ipv4**	**ipv6** } {**unicast**	**multicast**}
			例：
			`switch(config-router)# address-family vpnv4 unicast`
			`switch(config-router-af)#`
				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	**address-family** {**ipv4**	**ipv6**} {**unicast**	**multicast**}
			例：
			`switch(config-router)# address-family ipv4 unicast`
			`switch(config-router-af)#`
				アドレスファミリ コンフィギュレーション 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	**redistribute** {**direct**	{**eigrp**	**ospf**	**ospfv3**	**rip**} *instance-tag*	**static**} **route-map** *map-name* 例： `switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap`	他のプロトコルからのルートを BGP に再配布します。
ステップ 7	**redistribute** {**direct**	{**eigrp**	**isis**	**ospf**	**ospfv3**	**rip**} *instance-tag*	**static**	**icmpv6**} **route-map** *map-name* 例： `switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap`	他のプロトコルからのルートを BGP に再配布します。 Cisco NX-OS リリース 10.3(3)F 以降では、icmpv6 ルートを他のプロトコルから BGP に再配布するために **icmpv6** キーワードがサポートされています。
ステップ 8	（任意） **default-metric** *value* 例： `switch(config-router-af)# default-metric 33`	BGP へのデフォルト ルートを生成します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、EIGRP を BGP に再配布する例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap
switch(config-router-af)# copy running-config startup-config
```

DMZ リンク帯域幅

DMZ リンク帯域幅機能は、複数の自律システム出口リンクを介して到達可能な BGP 学習ルートへのトラフィック ロード バランシングを有効にするために使用されます。ロード バランシングは、これらのリンクの帯域幅に比例して行われます。

リンク帯域幅拡張コミュニティは、直接接続された 2 つの（シングルホップ）eBGP ピア間のリンクの帯域幅を伝送するために使用されます。dmz-link-bandwidth コマンドがネイバーのアドレスファミリ モードで構成されている場合、Nexus デバイスは、直接接続された eBGP ネイバーから受信した BGP ルートにこの拡張コミュニティを接続します。この拡張コミュニティ属性は、send-community extended または send-community both コマンドで拡張コミュニティ交換が有効になっているときに、iBGP ピアに伝達されます。この属性は、転送時に他のパスに関連するロード シェアリング値として使用されます。

さらに、BGP ピアから受信したルートのリンク帯域幅拡張通信を強制的に変更することもできます。また、ピアから受信したルートのサブセットのみにこの拡張コミュニティを構成することもできます。これを実現するには、ピアへのインバウンドルートマップを構成し、その下に「set extcommunity bandwidth <1-4000000>」を構成します。

注意事項と制約事項

BGP DMZ リンク帯域幅

リンク帯域幅機能を構成する前に、次の注意事項と制約事項を考慮してください。

- **dmz-link-bandwidth** コマンドは、BGP ネイバーの IPv4 ユニキャストおよび IPv6 ユニキャスト アドレス ファミリでのみ設定できます。
- リンク帯域幅拡張コミュニティは、直接接続された BGP ネイバーから受信したルートにのみ接続されます。BGP マルチホップ ネイバーでは行われません。
- グローバル モードと VRF モードの両方で構成できます。
- この機能を有効にするには、**maximum-paths** コマンドを使用して、アドレスファミリで BGP マルチパスロードバランシングを構成する必要があります。
- リンク帯域幅拡張コミュニティのアドバタイズ先の iBGP ネイバー間で、BGP 拡張コミュニティ交換がイネーブルになっている必要があります。
- リンク帯域幅拡張コミュニティは、ある VRF から別の VRF にリークされたルートにシームレスに伝送されます。

BGP DMZ リンク帯域幅の構成

Cisco NX-OS リリース 10.5(1)F 以降では、この機能を構成できます。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family [ipv4|ipv6] unicast**
4. **maximum-paths** *max-path*
5. **template peer** *peer-template-name*
6. **address-family [ipv4 | ipv6] unicast**
7. **dmz-link-bandwidth**
8. **neighbor** *neighbor*
9. **remote-as** *remote-as*
10. **address-family [ipv4 | ipv6] unicast**
11. **dmz-link-bandwidth**
12. **route-map** *name* **permit** *route*
13. **set extcommunity bandwidth <1-4000000>**
14. **neighbor** *neighbor* **address-family [ipv4 | ipv6] unicast route-map** *name* **in**

手順の詳細

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	**router bgp** *as-number* 例： `switch# configure terminal`	ルータ コンフィギュレーションモードを開始して、BGP ルーティング プロセスを作成または設定します。
ステップ 3	**address-family [ipv4|ipv6] unicast** 例： `switch(config)# address-family ipv4 unicast`	アドレス ファミリ IPv4 または IPv6 ユニキャストを構成します。
ステップ 4	**maximum-paths** *max-path* 例： `switch(config)# maximum-paths 10`	アドレスファミリで BGP マルチパスを有効にします。
ステップ 5	**template peer** *peer-template-name* 例： `switch(config)# template peer host_peer`	テンプレート モードを開始し、peer パラメータを構成します。
ステップ 6	**address-family [ipv4 | ipv6] unicast** 例： `switch(config)# address-family ipv4 unicast`	IPv4 または IPv6 のアドレス ファミリを構成します。
ステップ 7	**dmz-link-bandwidth** 例： `switch(config)# dmz-link-bandwidth`	リンク帯域幅拡張コミュニティを受信したルートに接続することで、直接接続されたピアへのトラフィックのロードバランシングを考慮するように BGP を構成します。

ステップ 8	**neighbor** *neighbor*
	例：
	`switch(config)# neighbor 1.1.1.1`
	`or`
	`switch(config)# neighbor 11::1`
		BGP ネイバーを構成します。

ステップ 9	**remote-as** *remote-as* 例： `switch(config)# remote-as 100`	ネイバーの自律システム番号を指定します。

	コマンドまたはアクション	目的
ステップ 10	**address-family [ipv4 | ipv6] unicast** 例： `switch(config)# address-family ipv4 unicast`	アドレス ファミリ IPv4 または IPv6 ユニキャストを構成します。
ステップ 11	**dmz-link-bandwidth** 例： `switch(config)# dmz-link-bandwidth`	リンク帯域幅拡張コミュニティを受信したルートに接続することで、直接接続されたピアへのトラフィックのロードバランシングを考慮するように BGP を設定します。
ステップ 12	**route-map** *name* **permit** *route* 例： `switch(config)# route-map change_link_bandwidth permit 10`	route-map を構成します。
ステップ 13	**set extcommunity bandwidth <1-4000000>** 例： `switch(config-route-map)# set extcommunity bandwidth 1000`	route-map を構成して、リンク帯域幅拡張コミュニティを設定しま

ステップ 14	**neighbor** *neighbor* **address-family [ipv4 | ipv6] unicast route-map** *name* **in**
	例：
	`switch(config-router)# neighbor 1.1.1.1`
	`switch (config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)# route-map change_link_bandwidth in`
		ネイバーを着信ルートマップに接続するように設定します。

BGP DMZ リンク帯域幅の構成例

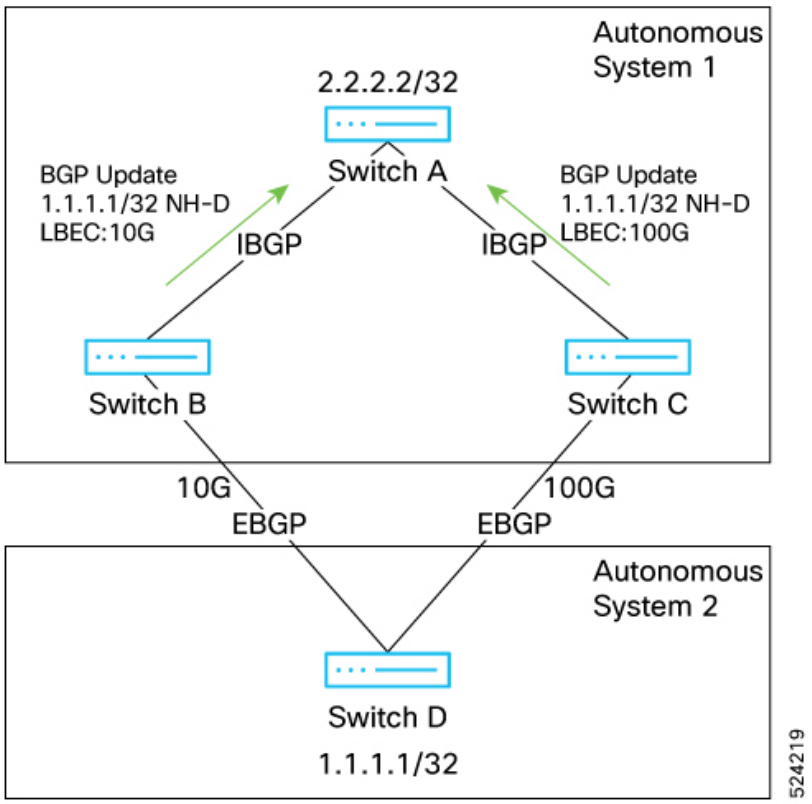
次の例では、AS 1 が 2 つの異なる帯域幅のリンクを介して AS 2 に接続されています。B-D リンクは 10G、C-D リンクは 100G です。B-D リンクは 10G、C-D リンクは 100G です。A-B と A-C は IBGP セッションによって接続されています。EBGP セッションによって接続された B-D と C-D。

トラフィックのロードバランシングが、これらのリンク帯域幅に合わせて比例して行われるようにする場合（つまり、A から C に対し、B に対するより 10 倍の発信トラフィックを送信する場合）は、B と C で、EBGP ネイバー D に向けて dmz-link-bandwidth コマンドを実行します。B は B-D の帯域幅をリンク帯域幅拡張コミュニティ（Link Bandwidth Extended Community、LBEC）にパッケージ化し、ルートエントリ 1.1.1.1/32 の BGP パスにアタッチします。同様に、C は C-D の帯域幅を LBEC にパッケージ化し、ルートエントリ 1.1.1.1/32 の BGP パスにアタッチします。

B と C は、iBGP ピア A に LBEC とともにルートをアドバタイズします。

A では、BGP がハッシュ 10/110 を NH-B に、100/110 を NH-C に指定して、転送をプログラムします。

図 *7 : DMZ* リンク帯域幅の構成



スイッチ B から D および B から A の構成

```
router bgp 1
 neighbor D
    address-family ipv4|v6 unicast
      dmz-link-bandwidth
  neighbor A
    address-family ipv4|v6 unicast
       send-community extended
```

スイッチ C から D および C から A の構成

```
router bgp 1
 neighbor D
    address-family ipv4|v6 unicast
      dmz-link-bandwidth
  neighbor A
    address-family ipv4|v6 unicast
       send-community extended
```

スイッチ A のコントロールプレーンとデータプレーンの状態

BGP テーブルの状態：

```
1.1.1.1/32
  NH-B    LBEC: 10G
  NH-C    LBEC: 100G
```

転送の状態：

```
1.1.1.1/32
  NH-B   hash 10:110
  NH-C   hash 100:110
```

リンク帯域幅拡張コミュニティを使用した不等コスト マルチパス（UCMP）の構成

始める前に

「BGP DMZ リンク帯域幅の構成」を参照してください。この機能を動作させるには、まずエッジ デバイスでその機能を構成する必要があります。つまり、エッジ デバイスでは、**dmz-link-bandwidth** コマンドを構成するか、 **set extcommunity link-bandwidth <1-4000000>** を使用してインバウンド ルートマップを構成することによって、直接接続された ebgp ピアから受信した BGP ルートにリンク帯域幅拡張コミュニティを接続する必要があります。 それが発生するまで、このセクションで説明されている機能はいずれも動作しません。

Cisco NX-OS リリース 10.5(1)F 以降では、ネイバーのアドレスファミリ モードで、「link-bandwidth cumulative」コマンドが構成されている場合に、BGPスピーカーは直接接続されているBGPネイバーにBGP学習ルートまで利用可能な累積帯域幅を伝えることができます。これは、リンク帯域幅拡張コミュニティを活用することで実現されます。この拡張コミュニティに挿入された値 n で BGP ルートをアドバタイズします。ここで、n = min（使用可能なすべてのマルチパスのリンク帯域幅拡張コミュニティから取得した帯域幅の合計、アドバタイズメントが送信されるネイバーへのリンクの帯域幅） 。

ガイドラインと制約事項

BGP UCMP 機能を構成する前に、次の注意事項と制約事項を考慮してください。

- **link-bandwidth cumulative** コマンドは、BGP ネイバーの IPv4 ユニキャストおよび IPv6 ユニキャスト アドレス ファミリでのみ構成できます。
- 直接接続された BGP ネイバーに対してのみ有効になります。
- このコマンドは、使用可能なすべてのマルチパスにリンク帯域幅拡張コミュニティがある場合にのみ有効です。
- グローバル モードと VRF モードの両方で構成できます。
- リンク帯域幅拡張コミュニティは、ある VRF から別の VRF にリークされたルートにシームレスに伝送されます。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *neighbor*
4. **remote-as** *remote-as*
5. **address-family [ipv4 | ipv6] unicast**
6. **send-community extended**
7. **link-bandwidth cumulative**

手順の詳細

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	**router bgp** *as-number* 例： `switch(config)# router bgp 120`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成または設定します。

ステップ 3	**neighbor** *neighbor*
	例：
	`switch(config)# neighbor 1.1.1.1`
	`or`
	`switch(config)# neighbor 11::1`
		BGP ネイバーを構成します。

ステップ 4	**remote-as** *remote-as* 例： `switch(config)# remote-as 100`	ネイバーの自律システム番号を指定します。
ステップ 5	**address-family [ipv4 | ipv6] unicast** 例： `switch(config)# address-family ipv4 unicast`	アドレス ファミリ IPv4 または IPv6 ユニキャストを構成します。
ステップ 6	**send-community extended** 例： `switch(config)# send-community extended`	このネイバーへの BGP 拡張コミュニティの送信を構成します。
ステップ 7	**link-bandwidth cumulative** 例：	累積リンク帯域幅をネイバーに送信します。この構成では、マルチパスのいずれかにリンク帯域幅拡張

	コマンドまたはアクション	目的
	`switch(config)# link-bandwidth cumulative`	コミュニティがない場合、累積リンク帯域幅はネイバーにアドバタイズされません。

設定例

設定例

次の図では、以下の点が描かれています：

- 4 つのクロスレイヤで直接接続された ebgp デバイス間のシングルホップ。
- すべてのリンクは 100G ですが、10G である B-D と B-E を除きます。これは、B から接続先 1.1.1.1 へのトラフィックが 20G のリンク帯域幅のためボトルネックになることを意味します。
- ユーザーがエンドツーエンドのトラフィックのロードバランスを望む場合、このリンク帯域幅の狭さを考慮する必要があります。
- レイヤ T1 の各デバイスで、レイヤ T0 のすべての BGPネイバーに対してコマンド **dmz-link-bandwidth** を実行します。
- レイヤ T1 の各デバイスで、レイヤ T2 のすべての BGP ピアに対してコマンド **link-bandwidth cumulative** を実行します。
- レイヤ T2 の各デバイスで、レイヤ T1 および T3 のすべての BGP ピアに対してコマンド **link-bandwidth cumulative** を実行します。
- レイヤ T3 の各デバイスで、レイヤ T2 のすべての BGP ピアに対してコマンド **link-bandwidth cumulative** を実行します。
- コマンド **dmz-link-bandwidth** により、スイッチ D はリンク D-F の帯域幅をリンク帯域幅拡張コミュニティ（LBEC）にパッケージ化し、ルートエントリ 1.1.1.1/32 の対応する BGP 経路にアタッチします。
- コマンド **dmz-link-bandwidth** により、スイッチ E はリンク E-F の帯域幅を LBEC にパッケージ化し、ルートエントリ 1.1.1.1/32 の対応する BGP 経路にアタッチします。
- コマンド **link-bandwidth cumulative** により、スイッチ A、B、C、D、および E は帯域幅 **n** を LBEC に挿入し、一方、BGP の更新をそれが構成されているピアにアドバタイズします。
- 値 **n** は、最小値を求める式（すべてのマルチパスの LBEC の合計、アップデートがアドバタイズされるピアへの帯域幅）を使用して計算されます。
- BGP アップデートでの LBEC の伝播により、すべてのデバイスでの転送が比例ハッシュを用いてプログラムされます。
- リンク障害が発生した場合、LBEC は動的に再計算されます。

図 *8* : 構成例 *1*

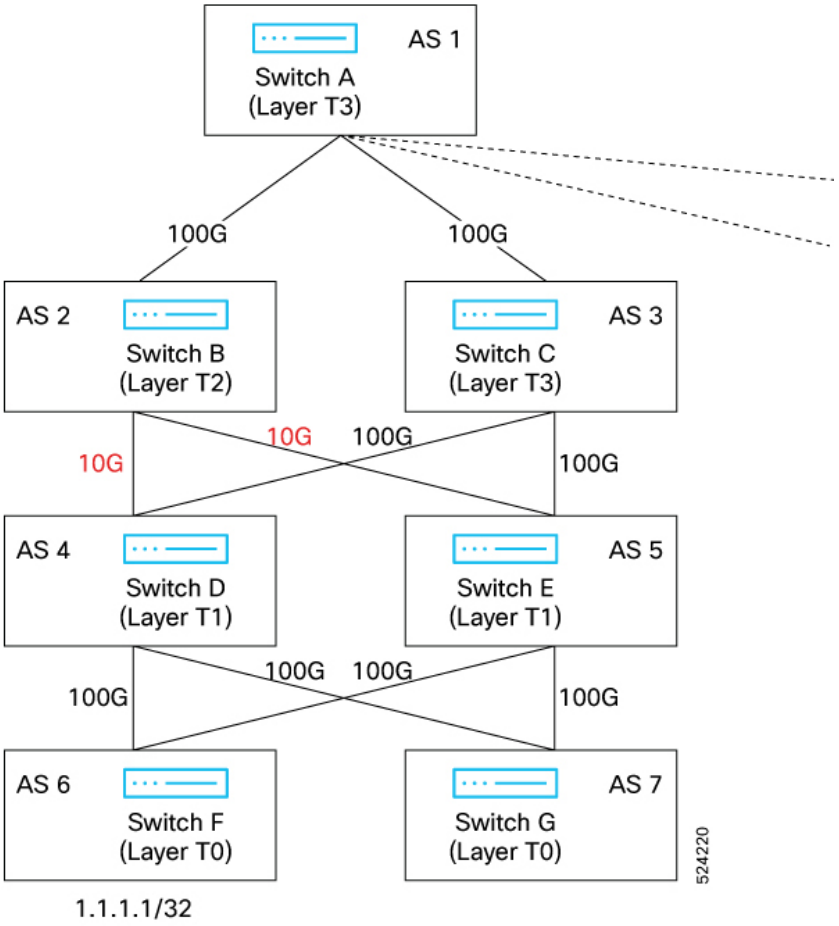
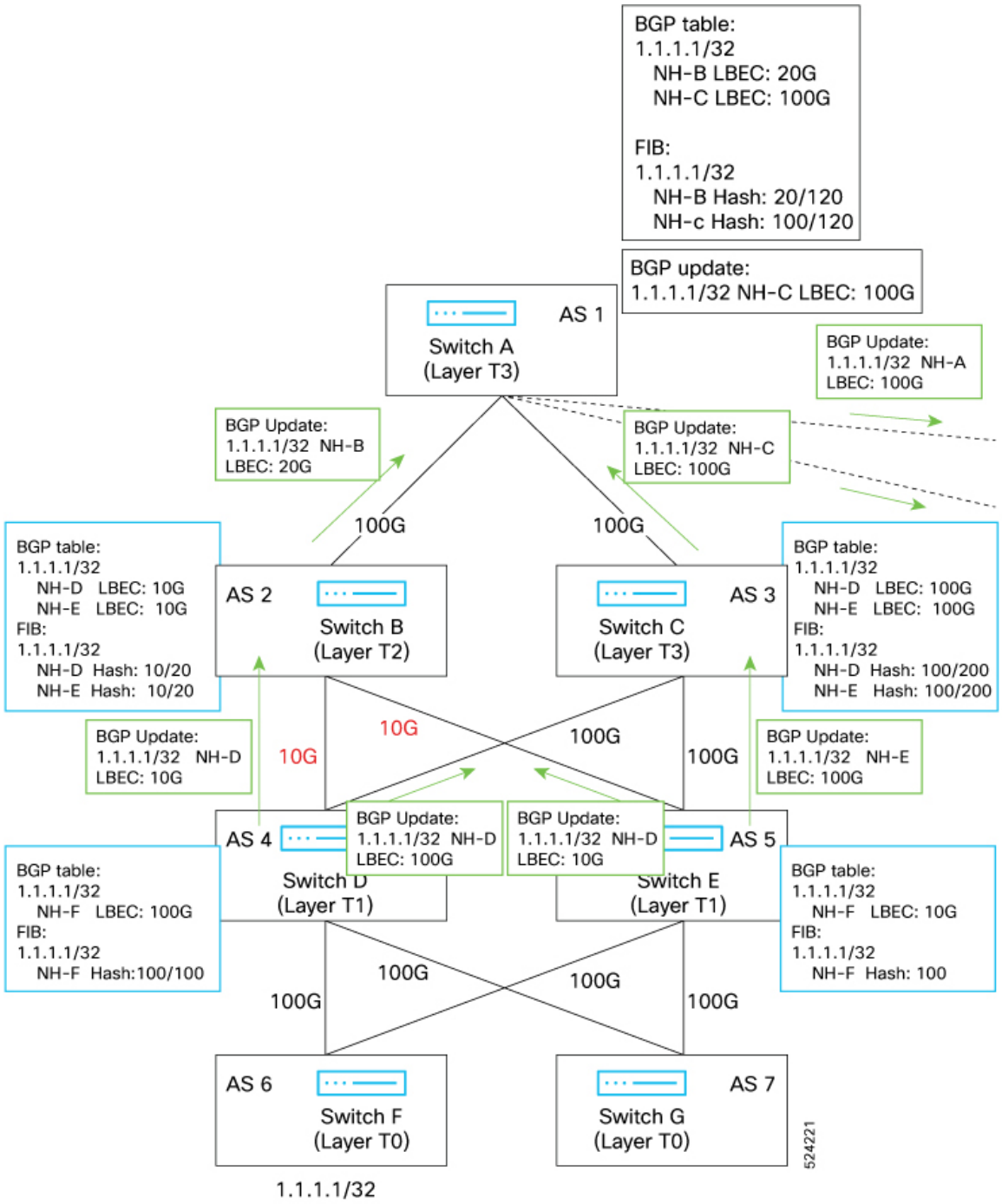

AS 1
Switch A
(Layer T3)
100G
100G
AS 2
Switch B
(Layer T2)
AS 3
Switch C
(Layer T3)
10G
10G
100G
100G
AS 4
Switch D
(Layer T1)
AS 5
Switch E
(Layer T1)
100G
100G
100G
100G
AS 6
Switch F
(Layer T0)
AS 7
Switch G
(Layer T0)
524220
1.1.1.1/32

図 *9*: 構成例 *2*



A から B および C への構成

```
router bgp 1
neighbor B
    address-family ipv4|v6 unicast
```

```
       link-bandwidth cumulative
       send-community extended
  neighbor C
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
  neighbor B'
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
  neighbor C'
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
```

B から D、A、および E への構成

```
router bgp 1
neighbor B
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
  neighbor C
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
  neighbor B'
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
  neighbor C'
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
```

C から A、D、および E への構成

```
router bgp 3
  neighbor A
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
  neighbor D
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
  neighbor E
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
```

D から F、B、および C への構成

```
router bgp 4
  neighbor F
    address-family ipv4|v6 unicast
      dmz-link-bandwidth
  neighbor B
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
neighbor C
```

```
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
```

E から F、B、および C への構成

```
router bgp 5
  neighbor F
    address-family ipv4|v6 unicast
      dmz-link-bandwidth
  neighbor B
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
neighbor C
    address-family ipv4|v6 unicast
       link-bandwidth cumulative
       send-community extended
```

設定の確認

構成を検証するには、次のコマンドを使用します：

- 次のコマンドを実行して、dmz-link-bandwidth コマンドがピア に対して有効になっているかどうかを確認します。

```
show bgp ipv4 unicast neighbors 192.168.11.2 | i i link
         dmz-link-bandwidth is enabled
```

- 次のコマンドを実行して、link-bandwidth cumulative コマンドがピアに対して有効になっているかどうかを確認します。

```
show bgp ipv4 unicast neighbors 10.1.1.2 | i i link
         link-bandwidth cumulative is enabled
```

- 次のコマンドを実行して、BGPパスにリンク帯域幅拡張コミュニティがあるかどうかを確認します。

```
show bgp ipv4 unicast 1.1.1.1/32
BGP routing table information for VRF default, address family IPv4 Unicast
BGP routing table entry for 1.1.1.1/32, version 403 Paths: (1 available, best #1)
Flags: (0x8000001a) (high32 0x002000) on xmit-list, is in urib, is best urib route,
 is in HW
Advertised path-id 1 Path type: external, path is valid, is best path, no labeled
nexthop, in rib

AS-Path: 10 33299 51178 47751 {27016} , path sourced external to AS 192.168.11.2
(metric 0) from 192.168.11.2 (192.168.11.2) Origin EGP, MED 2219, localpref 100,
weight 0 Community: 1:1 Extcommunity: LB:1:125000000

Path type: external, path is valid, is multi-path, no labeled nexthop, in rib
AS-Path: 10 33299 51178 47751 {27016} , path sourced external to AS 192.168.11.3
(metric 0) from 192.168.11.2 (192.168.11.2) Origin EGP, MED 2219, localpref 100,
weight 0 Community: 1:1 Extcommunity: LB:1:250000000
```

- 次のコマンドを実行して、回送テーブルにハッシュ比率があるかどうかを確認します。

```
 show ip route 1.1.1.1/32 detail

100.1.1.1/32, ubest/mbest: 1/0 *via 192.168.11.2, [20/2219], 00:14:22, bgp-1, bw:333,
 external, tag 10 client-specific data: 10 recursive next hop: 192.168.11.2/32
extended route information: BGP origin AS 0 BGP peer AS 10
```

```
100.1.1.2/32, ubest/mbest: 1/0 *via 192.168.11.3, [20/2219], 00:14:22, bgp-1, bw:666,
 external, tag 10 client-specific data: 10 recursive next hop: 192.168.11.3/32
extended route information: BGP origin AS 0 BGP peer AS 10
```

デフォルト ルートのアドバタイズ

デフォルトのルート（ネットワーク 0.0.0.0）をアドバタイズするように BGP を設定できます。

始める前に

BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。

手順の概要

1. **configure terminal**
2. **route-map allow permit**
3. **exit**
4. **ip route** *ip-address network-mask* **null** *null-interface-number*
5. **router bgp** *as-number*
6. **address-family** {**ipv4** | **ipv6**} **unicast**
7. **default-information originate**
8. **redistribute static route-map allow**
9. （任意） **copy running-config startup-config**

手順の詳細

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	**route-map allow permit**
	例：
	`switch(config)# route-map allow permit`
	`switch(config-route-map)#`
		ルータのマップ コンフィギュレーション モードを開始し、ルートを再配布する条件を定義します。。

ステップ 3	**exit**
	例：
	`switch(config-route-map)# exit`
	`switch(config)#`
		ルータのマップ設定モードを終了します。

ステップ 4	**ip route** *ip-address network-mask* **null** *null-interface-number*	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
	例： `switch(config)# ip route 192.0.2.1 255.255.255.0 null 0`	

ステップ 5	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP モードを開始し、AS 番号をローカルの BGP スピーカに割り当てます。

ステップ 6	**address-family** {**ipv4** | **ipv6**} **unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレスファミリ設定モードに入ります。

ステップ 7	**default-information originate** 例： `switch(config-router-af)# default-information originate`	デフォルトのルートをアドバタイズします。
ステップ 8	**redistribute static route-map allow** 例： `switch(config-router-af)# redistribute static route-map allow`	デフォルトのルートを再配布します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config-router-af)# copy running-config startup-config`	この設定変更を保存します。

BGP 属性フィルタリングの設定とエラー処理

Cisco NX-OS リリース 9.3(3) 以降では、BGP属性フィルタリングとエラー処理を設定して、セキュリティ レベルを向上させることができます。次の機能を利用でき、次の順序で実装されます。

- **パス属性 treat-as-withdraw:** アップデートに指定した属性タイプが含まれている場合に、指定したネイバーから受け取った BGP アップデートを treat-as-withdraw とすることを許可します。アップデートに含まれるプレフィックスは、ルーティングテーブルから削除されます。
- **パス属性 discard:**BGP アップデートの特定のパス属性を特定のネイバーから削除できます。
- **拡張属性エラー処理:** 形式が誤っているアップデートに起因するピア セッションのフラッピングを防止します。

属性タイプ 1、2、3、4、8、14、15、16 は、パス属性 treat-as-withdraw とパス属性 discard に対して設定できません。属性タイプ 9（Originator）、タイプ 10（Cluster-id）は、eBGP ネイバーでのみ設定できます。

BGP 更新メッセージからのパス属性の取り消しとしての処理

特定のパス属性を含むBGP更新を「扱うように」処理するには、ルータネイバーコンフィギュレーションモードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	[**no**] **path-attribute treat-as-withdraw** [*value* | **range** *start end*] **in**
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute treat-as-withdraw 100 in`
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute treat-as-withdraw range 21 255 in`
		指定されたパス属性またはパス属性の範囲を含む着信BGP更新メッセージをすべて取り消すものとして扱い、ルーティングテーブルが最新であることを確認するために着信ルートリフレッシュをトリガーします。treat-as-withdraw である BGP 更新のプレフィックスは、BGPルーティングテーブルから削除されます。
		このコマンドは、BGPテンプレートピアおよびBGPテンプレート ピア セッションでもサポートされます。

BGP 更新メッセージからのパス属性の破棄

特定のパス属性を含む BGP アップデートを廃棄するには、ルータ ネイバー コンフィギュレーション モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	[**no**] **path-attribute discard** [*value* | **range** *start end*] **in**
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute discard 100 in`
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute discard range 100 255 in`
		指定されたネイバーの BGP アップデート メッセージ内の指定されたパス属性をドロップし、ルーティングテーブルが最新であることを確認するために着信ルートリフレッシュをトリガーします。特定の属性または不要な属性の範囲全体を設定できます。
		このコマンドは、BGPテンプレートピアおよびBGPテンプレート ピア セッションでもサポートされます。
		（注）
		discard と treat-as-withdaw の両方に同じパス属性が設定されている場合、treat-as-withdaw の優先順位が高くなります。

拡張属性エラー処理のイネーブル化またはディセーブル化

BGP 拡張属性エラー処理はデフォルトで有効になっていますが、無効にすることもできます。この機能は、RFC 7606 に準拠しており、不正な更新によるピア セッションのフラッピングを防止します。デフォルトの動作は、eBGP ピアと iBGP ピアの両方に適用されます。

拡張エラー処理を無効または再度有効にするには、ルータ設定モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	[**no**] **enhanced-error**
	例：
	`switch(config)# router bgp 1000`
	`switch(config-router)# enhanced-error`
		BGP 拡張属性エラー処理をいネーブルまたはディセーブルにします。

取り消されたパス属性または破棄されたパス属性の表示

廃棄または不明なパス属性に関する情報を表示するには、次のいずれかのタスクを実行します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} **unicast path-attribute discard**]	属性が破棄されたすべてのプレフィックスを表示します。
show bgp {**ipv4** | **ipv6**} **unicast path-attribute unknown**]	不明な属性を持つすべてのプレフィックスを表示します。
show bgp {**ipv4** | **ipv6**} **unicast** *ip-address*	プレフィックスに関連付けられている不明な属性および破棄された属性を表示します。

次の例は、属性が廃棄されたプレフィックスを示しています。

```
switch# show bgp ipv4 unicast path-attribute discard
Network            Next Hop
1.1.1.1/32         20.1.1.1
1.1.1.2/32         20.1.1.1
1.1.1.3/32         20.1.1.1
```

次の例は、不明な属性を持つプレフィックスを示しています。

```
switch# show bgp ipv4 unicast path-attribute unknown
Network            Next Hop
2.2.2.2/32         20.1.1.1
2.2.2.3/32         20.1.1.1
```

次の例は、プレフィックスに関連付けられている不明な属性および破棄された属性を表示します。

```
switch# show bgp ipv4 unicast 2.2.2.2
BGP routing table entry for 2.2.2.2/32, version 6241
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  1000
    20.1.1.1 from 20.1.1.1 (20.1.1.1)
      Origin IGP, localpref 100, valid, external, best
      unknown transitive attribute: flag 0xE0 type 0x62 length 0x64
        value 0000 0000 0100 0000 0200 0000 0300 0000
              0400 0000 0500 0000 0600 0000 0700 0000
              0800 0000 0900 0000 0A00 0000 0B00 0000
              0C00 0000 0D00 0000 0E00 0000 0F00 0000
              1000 0000 1100 0000 1200 0000 1300 0000
              1400 0000 1500 0000 1600 0000 1700 0000
              1800 0000
      rx pathid: 0, tx pathid: 0x0
      Updated on Jul 20 2019 07:50:43 PST
```

BGP の調整

一連のオプション パラメータを使用することによって、BGP 特性を調整できます。

BGP を調整するには、ルータ コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
bestpath [**always-compare-med** | **as-pathmultipath-relax** | **compare-routerid** |**cost-community ignore** | **igp-metric ignore** |**med** {**confed** |**missing-as-worst**| **non-deterministic**}] 例： `switch(config-router)# bestpath always-compare-med`	ベストパス アルゴリズムを変更します。オプション パラメータは次のとおりです。 • **always-compare-med** ：異なる自律システム（AS）からのパスの MED を比較します。 • **as-path multipath-relax** — 異なる（ただし長さが等しい）ASパスを持つプロバイダ間でのロード シェアリングを許可します。このオプションを指定しないと、ASパスはロードシェアリングの場合に同一である必要があります。構成すると、BGPは異なるASNからの場合でも、潜在的なマルチパスの中からMEDが最も小さいベスト パスを選択します。 • **compare-routerid** ：同一の eBGP パスのルータ ID を比較します。 • **cost-community ignore** ：BGP ベストパス計算のコストコミュニティを無視します。 • **igp-metric ignore** ：ベスト パス選択時に内部ゲートウェイ プロトコル（IGP）メトリックを無視します。このオプションは、Cisco NX-OS リリース 9.2(2) 以降で使用可能です。 • **med confed** ：コンフェデレーション内からのパス間のみでMEDを比較するように最適なパスを強制します。 • **med missing-as-worst** ：消失 MED を最高の MED と見なします。 • **med non-deterministic** ：同じ自律システムからのパスの中から最適なMEDパスを決して選択しません。
enforce-first-as 例： `switch(config-router)# enforce-first-as`	ネイバー自律システムを eBGP の AS_path 属性で指定する最初の AS 番号にします。

コマンド	目的
log-neighbor-changes 例： `switch(config-router)# log-neighbor-changes`	ネイバーでステートが変化したときに、システム メッセージを生成します。 （注） 特定のネイバーのネイバー ステータス変化に関するメッセージを抑制するには、ルータ アドレスファミリ コンフィギュレーション モードで **log-neighbor-changes disable** コマンドを使用できます。
router-id *id* 例： `switch(config-router)# router-id 10.165.20.1`	この BGP スピーカのルータ ID を手動で設定します。
timers [*prefix-peer-wait* | *bgp holdtime* | **prefix-peer-timeout** *timeout* | **bestpath-limit** *bestpath-timeout*] 例： `switch(config-router)# timers bestpath-limit 300`	BGP タイマー値を設定します。オプション パラメータは次のとおりです。 • *prefix-peer-wait* ：プレフィックス ピアの待機タイマー。有効な範囲は 0 ～ 1200 秒です。デフォルトは 90 です。 • *bgp* ：BGP セッション キープアライブ時間。有効な範囲は 0 ～ 3600 秒です。デフォルト値は 60 です。 • *holdtime* ：異なる bgp キープアライブとホールド時間。範囲は 0 ～ 3600 秒で、デフォルト値は 60 秒です。 • *timeout* ：プレフィックス ピア タイムアウト値。有効な範囲は 0 ～ 1200 秒です。デフォルト値は 30 です。 • *bestpath-timeout* ：ベストパス タイムアウトを秒単位で設定します。デフォルト値は 300 です。大規模な BGP セットアップが予想される場合、スケールに基づいて、タイムアウト値を 480～1200 に設定する必要があります。 このコマンドの設定後、BGP セッションを手動でリセットする必要があります。

BGP を調整するには、ルータ アドレス ファミリ設定モードで次のオプション コマンドを使用します。

コマンド	目的
distance *ebgp-distance ibgp-distance local-distance* 例： `switch(config-router-af)# distance 20 100 200`	BGP のアドミニストレーティブディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトの設定は次のとおりです。 • *ebgp-distance* —20 • *ibgp-distance* —200 • *local-distance* —220 ローカル ディスタンスは、集約廃棄ルートが RIB に組み込まれている場合に、集約廃棄ルートに使用するアドミニストレーティブディスタンスです。 外部アドミニストレーティブディスタンスの値を入力したら、要件に応じて内部ルートのアドミニストレーティブディスタンスの値またはローカルルートのアドミニストレーティブディスタンスの値を入力する必要があります。内部/ローカルルートもルート管理で考慮されます。
log-neighbor-changes [**disable**] 例： `switch(config-router-af)# log-neighbor-changes disable`	この特定のネイバーの状態が変化すると、システム メッセージを生成します。 **disable** オプションを使用すると、この特定のネイバーのネイバー ステータス変化に関するメッセージが抑制されます。

BGP を調整するには、ネイバー コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
description *string* 例： `switch(config-router-neighbor)# description main site`	この BGP ピアを説明するストリングを設定します。ストリングには最大 80 の英数字を使用できます。
low-memory exempt 例： `switch(config-router-neighbor)# low-memory exempt`	メモリ不足状態によるシャットダウンからこの BGP ネイバーを除外します。

コマンド	目的
transport connection-mode passive 例： `switch(config-router-neighbor)# transport connection-mode passive`	受動接続の確立だけが可能です。この BGP スピーカは BGP ピアへの TCP 接続を開始しません。このコマンドの設定後、BGP セッションを手動でリセットする必要があります。
[**no** | **default**] **remove-private-as** [**all** | **replace-as**] 例： `switch(config-router-neighbor)# remove-private-as`	eBGP ピアへの発信ルート アップデートからプライベート AS 番号を削除します。このコマンドによって、BGP ネイバーセッションの自動ソフト クリアまたはリフレッシュが開始されます。 オプション パラメータは次のとおりです。 • **no** ：コマンドをディセーブルにします。 • **default** ：デフォルト モードにコマンドを移動します。 • **all** ：AS パスからすべてのプライベート AS 番号を削除します。 • **replace-as** ：すべてのプライベート AS 番号を replace-as AS-path 値に置き換えます。 このコマンドの詳細については、「*BGP* のガイドラインと制限事項」セクションを参照してください。
update-source *interface-type number* 例： `switch(config-router-neighbor)# update-source ethernet 2/1`	ピアとの BGP セッション用に設定されたインターフェイスの送信元 IP アドレスを使用するように、BGP スピーカを設定します。このコマンドによって、BGP ネイバーセッションの自動通知およびセッション リセットが開始されます。単一ホップ iBGP ピアでは、**update-source** が設定されている場合に、高速外部フォールオーバーをサポートします。

BGP を調整するには、ネイバー アドレス ファミリ コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
allowas in 例： `switch(config-router-neighbor-af)# allowas in`	BRIP にインストールする AS パスにルート自体の AS を持つことを可能にします。

コマンド	目的
default-originate [**route-map** *map-name*] 例： `switch(config-router-neighbor-af)# default-originate`	BGP ピアへのデフォルト ルートを作成します。
disable-peer-as-check 例： `switch(config-router-neighbor-af)# disable-peer-as-check`	デバイスが同じASパスで一方のノードからもう一方のノードに学習されたルートをアドバタイズすると同時に、ピアAS番号のチェックをディセーブルにします。
filter-list *list-name* {**in** | **out**} 例： `switch(config-router-neighbor-af)# filter-list BGPFilter in`	着信または発信ルートアップデートに関して、このBGPピアにAS_pathフィルタリストを適用します。このコマンドによって、BGP ネイバーセッションの自動ソフトクリアまたはリフレッシュが開始されます。
prefix-list *list-name* {**in** | **out**} 例： `switch(config-router-neighbor-af)# prefix-list PrefixFilter in`	着信または発信ルートアップデートに関して、このBGPピアにプレフィックスリストを適用します。このコマンドによって、BGPネイバーセッションの自動ソフトクリアまたはリフレッシュが開始されます。
send-community 例： `switch(config-router-neighbor-af)# send-community`	このBGPピアにコミュニティ属性を送信します。このコマンドによって、BGP ネイバーセッションの自動ソフトクリアまたはリフレッシュが開始されます。
send-community extended 例： `switch(config-router-neighbor-af)# send-community extended`	このBGPピアに拡張コミュニティ属性を送信します。このコマンドによって、BGPネイバーセッションの自動ソフトクリアまたはリフレッシュが開始されます。
suppress-inactive 例： `switch(config-router-neighbor-af)# suppress-inactive`	ベスト（アクティブ）ルートだけをBGPピアにアドバタイズします。このコマンドによって、BGPネイバーセッションの自動ソフトクリアまたはリフレッシュが開始されます。
[no | default] as-override 例： `switch(config-router-neighbor-af)# as-override`	**no**-（オプション）コマンドを無効にします。 **default**：（オプション）デフォルトモードにコマンドを移動します。 **as-override**：eBGP ピアに更新を送信する際に、パス属性内のピアのAS番号をすべてローカル AS 番号に置き換えます。

ポリシーベースのアドミニストレーティブ ディスタンスの設定

設定されたルートマップで説明されているポリシーに一致する外部BGP（eBGP）と内部BGP（iBGP）の距離を設定できます。ルート マップで設定された距離は、一致するルートとともにユニキャストRIBにダウンロードされます。BGPは最適パスを使用して、ユニキャストRIBテーブルのネクスト ホップをダウンロードするときのアドミニストレーティブ ディスタンスを決定します。ポリシーに match 句または deny 句がない場合、BGP は distance コマンドで設定された距離またはルートのデフォルトの距離を使用します。

ポリシーベースのアドミニストレーティブディスタンス機能は、2つの異なるルーティング プロトコルから同じ宛先に 2 つ以上のルートが存在する場合に役立ちます。

始める前に

BGP を有効にする必要があります。

手順の概要

1. switch# **configure terminal**
2. switch(config)# **ip prefix-list** *name* **seq** *number* **permit** *prefix-length*
3. switch(config)# **route-map** *map-tag* **permit** *sequence-number*
4. switch(config-route-map)# **match ip address prefix-list** *prefix-list-name*
5. switch(config-route-map)# **set distance** *value1 value2 value3*
6. switch(config-route-map)# **exit**
7. switch(config)# **router bgp** *as-number*
8. switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **unicast**
9. switch(config-router-af)# **table-map** *map-name*
10. （任意） switch(config-router-af)# **show forwarding distribution**
11. （任意） switch(config)# **copy running-config startup-config**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	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	switch(config)# **ip prefix-list** *name* **seq** *number* **permit** *prefix-length*	permit キーワードを使用して、IP パケットまたはルートを照合するためのプレフィクス リストを作成します。
ステップ 3	switch(config)# **route-map** *map-tag* **permit** *sequence-number*	permit キーワードを使用してルート マップを作成し、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。ルートの一致基準がポリシー内で満

	コマンドまたはアクション	目的
		たされると、パケットはポリシーでルーティングされます。
ステップ 4	switch(config-route-map)# **match ip address prefix-list** *prefix-list-name*	プレフィクス リストに基づいて IPv4 ネットワーク ルートを照合します。プレフィクス リスト名には最大 63 文字の英数字を使用できます。
ステップ 5	switch(config-route-map)# **set distance** *value1 value2 value3*	ローカル自律システムから発信される内部 BGP（iBGP）または外部 BGP （eBGP）ルートおよび BGP ルートのアドミニストレーティブ ディスタンスを指定します。範囲は 1 ～ 255 です。 外部アドミニストレーティブ ディスタンスの値を入力したら、要件に応じて内部ルートのアドミニストレーティブディスタンスの値またはローカルルートのアドミニストレーティブ ディスタンスの値を入力する必要があります。内部/ローカル ルートもルート管理で考慮されます。
ステップ 6	switch(config-route-map)# **exit**	ルート マップ設定モードを終了します。
ステップ 7	switch(config)# **router bgp** *as-number*	BGP モードを開始し、AS 番号をローカルの BGP スピーカに割り当てます。
ステップ 8	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **unicast**	アドレス ファミリ設定モードを開始します。
ステップ 9	switch(config-router-af)# **table-map** *map-name*	BGP ルートを RIB テーブルに転送する前にそのルートのルート マップの選択的アドミニストレーティブ ディスタンスを設定します。テーブル マップ名には最大 63 文字の英数字を使用できます。 （注） VRF アドレスファミリ設定モードで **table-map** コマンドを設定することもできます。
ステップ 10	（任意） switch(config-router-af)# **show forwarding distribution**	フォワーディング情報の配布を表示します。
ステップ 11	（任意） switch(config)# **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。

マルチプロトコル BGP の設定

複数のアドレス ファミリ（IPv4 および IPv6 のユニキャストおよびマルチキャスト ルートを含む）をサポートするように MP-BGP を設定できます。

始める前に

- BGP を有効にする必要があります（「BGP の有効化」の項を参照）。

BGPをイネーブルにする必要があります。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. switch(config-router-neighbor)# **address-family ipv4** {**unicast** | **multicast**}
5. switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** } {**unicast** | **multicast**}
6. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
7. （任意） **copy running-config startup-config**

手順の詳細

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** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65534`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー設定モードを開始し、ネイバー IP アドレスを設定します。

ステップ 4	switch(config-router-neighbor)# **address-family ipv4** {**unicast**	**multicast**}	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	switch(config-router-neighbor)# **address-family** {**ipv4**	**ipv6** } {**unicast**	**multicast**}	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 6	**address-family** {**ipv4**	**ipv6**} {**unicast**	**multicast**} 例：	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
	`switch(config-router-neighbor)# address-family ipv4 multicast` `switch(config-router-neighbor-af)#`	
ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

次に、ネイバーのマルチキャスト RPF に対して IPv4 ルートのアドバタイズおよび受信をイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# ipv4 address 2001:0DB8::1
switch(config-if)# router bgp 65536
switch(config-router)# neighbor 192.168.1.2 remote-as 35537
switch(config-router-neighbor)# address-family ipv4 multicast
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv4 multicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

次に、ネイバーのマルチキャスト RPF に対して IPv4 および IPv6 ルートのアドバタイズおよび受信をイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# ipv6 address 2001:0DB8::1
switch(config-if)# router bgp 65536
switch(config-router)# neighbor 192.168.1.2 remote-as 35537
switch(config-router-neighbor)# address-family ipv4 multicast
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 multicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

BMP の設定

Cisco NX-OS リリース 7.0(3)I5(2) 以降では、デバイスに BMP を設定できます。

始める前に

BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。

手順の概要

1. **configure terminal**
2. **router bgp as-number**

3. **bmp server** *server-number*
4. **address ip-address port-number port-number**
5. **description** *string*
6. **initial-refresh** { *skip* | *delay time*}
7. **initial-delay** *time*
8. **stats-reporting-period** *time*
9. shutdown
10. **vrf** *vrf-name*
11. **update-source** <*interface-name*>
12. **neighbor ip-address**
13. **remote-as** *as-number*
14. **bmp-activate-server** *server-number*
15. （任意） **show bgp bmp** *server [server-number] [detail]*
16. （任意） **copy running-config startup-config**

手順の詳細

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	**router bgp as-number** 例： `switch(config)# router bgp 200`	BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。
ステップ 3	**bmp server** *server-number* 例： `switch(config-router-bmp)# bmp-server 1`	BGP が情報を送信する BMP サーバを設定します。サーバ番号がキーとして使用されます。 （注） 最大 2 つの BMP サーバを設定できます。
ステップ 4	**address ip-address port-number port-number** 例： `switch(config-router-bmp)# address 10.1.1.1 port-number 2000`	ホストの IPv4 または IPv6 アドレスと、BMP スピーカーが BMP サーバに接続するポート番号を設定します。
ステップ 5	**description** *string* 例： `switch(config-router-bmp)# description BMPserver1`	BMP サーバの説明を設定します。最大 256 文字の英数字を入力できます。

	コマンドまたはアクション	目的
ステップ 6	**initial-refresh** { *skip* | *delay time*} 例： `switch(config-router-bmp)# initial-refresh delay 100`	BGP がコンバージされ、後で BMP サーバ接続が確立されたときにルート リフレッシュを送信するオプションを設定します。 skip オプションは、BMP サーバ接続が後でアップした場合にルート リフレッシュを送信しないことを指定します。 delay オプションは、ルート更新を送信するまでの時間を秒単位で指定します。有効範囲は 30 ～ 720 秒で、デフォルトは 30 秒です。
ステップ 7	**initial-delay** *time* 例： `switch(config-router-bmp)# initial-delay 120`	BMP サーバへの接続が試行されるまでの遅延を設定します。有効範囲は 30 ～ 720 秒で、デフォルトは 45 秒です。
ステップ 8	**stats-reporting-period** *time* 例： `switch(config-router-bmp)# stats-reporting-period 50`	BMP サーバが BGP ネイバーから統計レポートを受信する時間間隔を設定します。有効範囲は 30 ～ 720 秒で、デフォルトはディスエーブルです。
ステップ 9	shutdown 例： `switch(config-router-bmp)# shutdown`	BMP サーバへの接続を無効にします。
ステップ 10	**vrf** *vrf-name* 例： `switch(config-router-bmp)# vrf BMP`	BMP サーバが到達可能な VRF を選択します。
ステップ 11	**update-source** <*interface-name*> 例： `switch(config-router-bmp)# update-source ethernet4/2`	BMP サーバ接続の確立に使用するローカル インターフェイスを選択します。
ステップ 12	**neighbor ip-address** 例： `switch(config-router-bmp)# neighbor 192.168.1.2`	BGP ルーティング用のネイバー コンフィギュレーション モードを開始し、ネイバー IP アドレスを設定します。
ステップ 13	**remote-as** *as-number* 例： `switch(config-router-neighbor)# remote-as 65535`	リモート BGP ピアの AS 番号を設定します。
ステップ 14	**bmp-activate-server** *server-number* 例：	ネイバーの情報の送信先となる BMP サーバを設定します。

	コマンドまたはアクション	目的
	`switch(config-router-neighbor)# bmp-activate-server 1`	
ステップ 15	（任意） **show bgp bmp** *server [server-number] [detail]* 例： `switch(config-router-neighbor)# show bgp bmp server`	BMP サーバ情報を表示します。
ステップ 16	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。

BGP ローカル ルート リーク

BGP ローカル ルート リークについて

リリース 9.3(1) 以降、NX-OS BGP は、次の間のインポートされた VPN ルートのリークをサポートします。

- VPN ルート テーブルとデフォルト VRF ルート テーブル
- VPN ルート テーブルと VRF-Lite ルート テーブル
- リーフからリーフへの接続用のボーダー リーフ（BL）スイッチ ルート テーブル

この機能により、ルート テーブル間のルートの伝播が可能になります。インポート マップまたはエクスポート マップを設定することで、VRF のルート リークを制御できます。このマップには、ローカルで発生した着信ルートを許可または禁止し、アドバタイズするかどうかを指定するオプションが含まれています。ローカル ルート リークは双方向であるため、ローカルに発信されたルートは VRF から BGP VPN にリークされ、BGP VPN からインポートされたルートは VRF にリークされます。

（注） NX-OS は、中央集中型ルート リークと呼ばれる同様の機能をサポートしています。詳細については、「レイヤ 3 仮想化の設定」を参照してください。

BGP ローカル ルート リークの注意事項と制約事項

BGPローカルルートリーク機能の注意事項と制約事項は次のとおりです。

- この機能は、次のシスコ ハードウェアによりサポートされます。
  - この機能は、Cisco Nexus 9332C、9364C、9300-EX、9300-FX/FXP/FX2/FX3、および 9300-GX プラットフォーム スイッチと、9700-EX/FX ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチに導入されました。

- -R ライン カード搭載の Cisco Nexus 9500 プラットフォーム スイッチ

- ルート ターゲットを使用する場合、同じルート ターゲットが同じリモート パスを指す重複パスを持っている可能性があり、これがスイッチのメモリとパフォーマンスに悪影響を及ぼす可能性があります。ルート ターゲットを使用する場合は注意してください。
- 同じ VRF 間で境界リーフ ルータ（BL）がリークするリーフツーリーフの場合に、ローカル ルート リークを使用する場合は注意してください。このシナリオでは、ルーティング ループが発生しやすくなります。インポートされたルートを他の BL から除外するには、インバウンド ルート マップを使用することを推奨します。
- リモート パスが取り消された後、BGP がパスを完全にクリーンアップするまでにさらに 20 秒かかることがあります。

デフォルト VRF にリークするために VPN からインポートされたルートを設定する

VRF を設定して、BGP VPN からインポートされたルートが、デフォルトの VRF へエクスポートされることを許可することができます。この手順は、デフォルト以外の VRF に使用します。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **config terminal**
2. **vrf context** *vrf-name*
3. **address-family** *address-family sub family*
4. **export vrf default [***prefix-limit***] map***route-map* **allow-vpn**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**config terminal**
	例：
	`switch-1# config terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(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-1(config)# vrf context vpn1`
	`switch-1(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

	コマンドまたはアクション	目的

ステップ 3	**address-family** *address-family sub family*
	例：
	`switch-1(config-vrf)# address-family ipv4 unicast`
	`switch-1(config-vrf-af-ipv4)#`	

ステップ 4	**export vrf default [***prefix-limit***] map***route-map* **allow-vpn**
	例：
	`switch-1(config-vrf-af-ipv4)# export vrf default map vpnmap1 allow-vpn`
	`switch-1(config-vrf-af-ipv4)#`
		現在の VRF を設定して、BGP VPN からインポートされたルートが、デフォルトの VRF へエクスポートされることを許可します。

デフォルト VRF からリークされたルートを VPN にエクスポートするための設定

デフォルト VRF からリークされたルートを BGP VPN にエクスポートできるように VRF を設定できます。この手順は、デフォルト以外の VRF に使用します。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **config terminal**
2. **vrf context** *vrf-name*
3. **address-family** *address-family sub family*
4. **import vrf default [***prefix-limt***] map***route-map* **advertise-vpn**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**config terminal**
	例：
	`switch-1# config terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(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-1(config)# vrf context vpn1`
	`switch-1(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

ステップ 3	**address-family** *address-family sub family*	

	コマンドまたはアクション	目的

	例：
	`switch-1(config-vrf)# address-family ipv4 unicast`
	`switch-1(config-vrf-af-ipv4)#`	

ステップ 4	**import vrf default [***prefix-limt***] map***route-map* **advertise-vpn**
	例：
	`switch-1(config-vrf-af-ipv4)# import vrf map vpnmap1 advertise-vpn`
	`switch-1(config-vrf-af-ipv4)#`
		デフォルト VRF からインポートされたルートを BGP VPN にエクスポートできるように現在の VRF を設定します。

VRF にエクスポートするために VPN からインポートしたルートの設定

VPN でインポートされたルートを別の VRF にエクスポートできるように VRF を設定できます。この手順は、デフォルト以外の VRF に使用してください。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **config terminal**
2. **vrf context** *vrf-name*
3. **address-family** *address-family sub family*
4. **export vrf allow-vpn**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**config terminal**
	例：
	`switch-1# config terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(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-1(config)# vrf context vpn1`
	`switch-1(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

ステップ 3	**address-family** *address-family sub family*	

	コマンドまたはアクション	目的

	例：
	`switch-1(config-vrf)# address-family ipv4 unicast`
	`switch-1(config-vrf-af-ipv4)#`	

ステップ 4	**export vrf allow-vpn**
	例：
	`switch-1(config-vrf-af-ipv4)# export vrf allow-vpn`
	`nxosv2(config-vrf-af-ipv4)#`
		BGP VPM からインポートしたルートをデフォルト以外の VRF にエクスポートできるように VRF を設定します。

VRF からインポートして VPN にエクスポートするルートの設定

VRF は、別の VRF からインポートされたルートを BGP VPN にエクスポートできるように設定することができます。この手順は、デフォルト以外の VRF に使用してください。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **config terminal**
2. **vrf context** *vrf-name*
3. **address-family** *address-family sub family*
4. **import vrf advertise-vpn**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**config terminal**
	例：
	`switch-1# config terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(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-1(config)# vrf context vpn1`
	`switch-1(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

ステップ 3	**address-family** *address-family sub family* 例：	

	コマンドまたはアクション	目的
	switch-1(config-vrf)# **address-family ipv4 unicast** switch-1(config-vrf-af-ipv4)#	
ステップ 4	**import vrf advertise-vpn** 例： switch-1(config-vrf-af-ipv4)# **import vrf advertise-vpn** nxosv2(config-vrf-af-ipv4)#	別の VRF からインポートされたルートを BGP VPN にエクスポートできるように現在の VRF を設定します。

設定例

次に、BGP ローカル ルート リーク機能の設定例を示します。

BGP VPN からデフォルト VPN への到達可能性の設定

この例では、VPN とデフォルト VRF の間にある、VRF_A と呼ばれる中間 VRF を介して、ルートの再インポートを有効にします。

```
vrf context VRF_A
   address-family ipv4 unicast
   route-target both auto evpn
   import vrf default map MAP_1 advertise-vpn
   export vrf default map MAP_1 allow-vpn
```

ルートの再インポートは、VPN から VRF_A へのルートのインポートを制御する **advertise-vpn** オプションを使用して、また、VRF_A からデフォルト VRF への VPN インポート ルートのエクスポートを制御する、エクスポート マップのための **allow-vpn** を使用して有効にできます。設定は中間 VRF で行われます。

VPN から VRF-Lite への到達可能性の設定

この例では、VPN は VRF_A と呼ばれるテナント VRF に接続します。VRF_A は、VRF-B と呼ばれる VRF-Lite に接続します。この設定により、VPN でインポートされたルートを VRF_A から VRF_B にリークできます。

```
vrf context VRF_A
   address-family ipv4 unicast
   route-target both auto
   route-target both auto evpn
   route-target import 3:3
   route-target export 2:2
   import vrf advertise-vpn
   export vrf allow-vpn
vrf context VRF_B
   address-family ipv4 unicast
   route-target both 1:1
   route-target import 2:2
   route-target export 3:3
```

2つの間のルート リークは、VRF_A（テナント）で設定されたエクスポート マップで **allow-vpn** を使用してイネーブルにします。VRF_A のエクスポート マップでは、VPN からインポートされたルートを VRF_B にリークできます。エクスポート マップによって処理されたルートは、ルート ターゲットのルート セットに追加される、**route-mapexport** および **export-map** 属性を

持ちます。インポート マップは、**advertise-vpn** を使用して、VRF-Lite からインポートされたルートを VPN にエクスポートできるようにします。

VRF 間でルート リークが発生すると、ルートは再発信され、そのルート ターゲットは、新しい VRF の設定で指定されたルート ターゲット エクスポートおよびエクスポート マップ属性で置き換えられます。

リーフからリーフへの到達可能性

この例では、2 つの VPN と 2 つの VRF が存在します。VPN_1 は VRF_A に接続され、VPN_2 は VRF_B に接続されます。両方の VRF はルート識別子（RD）です。

```
vrf context VRF_A
   address-family ipv4 unicast
   route-target both auto
   route-target both auto evpn
   route-target import 3:3
   route-target export 2:2
   import vrf advertise-vpn
   export vrf allow-vpn
vrf context VRF_B
   address-family ipv4 unicast
   route-target both 1:1
   route-target import 2:2
   route-target export 3:3
   import vrf advertise-vpn
   export vrf allow-vpn
```

この 2 つの間のルート リークは、VRF_A および VRF_B で設定されたエクスポート マップの **allow-vpn** で有効にされます。VPN によってインポートされたルートには、ルート ターゲットのルート セットに追加された **route-mapexport** と **export-map** 属性があります。インポート マップのマップは、各 VRF からインポートされたルートが VPN にエクスポートされるようにする advertise-vpn オプションを使用します。

VRF 間でルート リークが発生すると、ルートは再発信され、そのルート ターゲットは、新しい VRF の設定で指定されたルート ターゲット エクスポートおよびエクスポート マップ属性で置き換えられます。

ループ防止付きリーフツーリーフ

リーフツーリーフ設定では、ルート マップに注意を払わないでいると、同じ VRF 間でリークしている BL 間のループが誤って発生する可能性があります。

- 各 BL でインバウンド ルート マップを使用すれば、他のすべての BL からの更新を拒否できます。
- BL がルートを発信する場合には、標準コミュニティを適用できます。これにより、他の BL はルートを受け入れることができます。このコミュニティは、受信側の BL で削除されます。

次の例では、VTEP 3.3.3.3、4.4.4.4、および 5.5.5.5 が BL です。

```
ip prefix-list BL_PREFIX_LIST seq 5 permit 3.3.3.3/32
ip prefix-list BL_PREFIX_LIST seq 10 permit 4.4.4.4/32
ip prefix-list BL_PREFIX_LIST seq 20 permit 5.5.5.5/32
ip community-list standard BL_COMMUNITY seq 10 permit 123:123
```

```
route-map INBOUND_MAP permit 5
  match community BL_COMMUNITY
  set community none
route-map INBOUND_MAP deny 10
  match ip next-hop prefix-list BL_PREFIX_LIST
route-map INBOUND_MAP permit 20
route-map OUTBOUND_SET_COMM permit 10
  match evpn route-type 2 mac-ip
  set community 123:123
route-map SET_COMM permit 10
  set community 123:123
route-map allow permit 10

vrf context vni100
  vni 100
  address-family ipv4 unicast
    route-target import 2:2
    route-target export 1:1
    route-target both auto
    route-target both auto evpn
    import vrf advertise-vpn
    export vrf allow-vpn

vrf context vni200
  vni 200
  address-family ipv4 unicast
    route-target import 1:1
    route-target export 2:2
    route-target both auto
    route-target both auto evpn
    import vrf advertise-vpn
    export vrf allow-vpn

router bgp 100
  template peer rr
    remote-as 100
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      route-map INBOUND_MAP in
      route-map OUTBOUND_SET_COMM out
  neighbor 101.101.101.101
    inherit peer rr
  neighbor 102.102.102.102
    inherit peer rr
  vrf vni100
    address-family ipv4 unicast
      network 3.3.3.100/32 route-map SET_COMM
  vrf vni200
    address-family ipv4 unicast
      network 3.3.3.200/32 route-map SET_COMM
```

この例では、ボーダーリーフ（BL）ルータのテナント VRF は追加のインポート エクスポート フローを有効にすることで、トラフィックをリークできます。ルート マップ内のルート ターゲットは、ルートのインポート元またはエクスポート先を決定します。

VRF のマルチパス

この例では、VPN に複数の着信パスがあります。この設定により、VRF_A と呼ばれる中間 VRF（VPN と別の VRF の間にあり、VRF_B と呼ばれるもの）を介したルート リークが可能になります。マルチパスが VRF_A で有効になっているとします。

```
vrf context VRF_A
   address-family ipv4 unicast
   route-target both auto evpn
   route-target export 3:3
   export vrf allow-vpn
vrf context VRF_B
   address-family ipv4 unicast
   route-target import 3:3
```

ルートリークは、VRF_A で設定されたエクスポート マップの **allow-vpn** で有効になっています。特定のプレフィックスの 2 つのパスが VPN から学習されて VRF_A にインポートされると、同じ送信元 RD（VRF_A のローカル RD）を持つ 2 つの異なるパスが VRF_B に存在するようになります。各ルートは、元の送信元 RD（リモート RD）によって区別されます。

パスの重複

この例では、設定により単一の VPN パスを VRF_A と VRF_B の両方にインポートできるようになっています。VRF_A は **export vrf allow-vpn** で設定されているため、VRF_A もそのルートを VRF_B にリークします。VRF_B には同じ送信元 RD（VRF_A のローカル RD）を持つ 2 つのパスがありますが、それらは元の送信元 RD（リモート RD）によって区別されます。

```
vrf context VRF_A
   address-family ipv4 unicast
      route-target import 1:1 evpn
      route-target export 1:1 evpn
      route-target export 2:2
      export vrf allow-vpn
vrf context VRF_B
   address-family ipv4 unicast
     route-target import 1:1 evpn
     route-target import 2:2
```

この設定では、マルチパスが存在しない状況が発生します。

BGP ローカル ルート リーク情報の表示

次の show コマンドには、BGP ローカル ルート リーク機能に関する情報が含まれています。

コマンド	アクション
show bgp vrf *vrf-name* **process**	デフォルトまたはデフォルト以外のVRFの場合、**import advertise-vpn** および **export allow-vpn** オプションのイネーブル状態（YesまたはNo）が表示されます。
show bgp vrf *vrf-name* **ipv4 unicast** *prefix*	ルートのインポート元の宛先のリストなど、インポートされたパスに関する情報を表示します。

BGP グレースフル シャットダウン

BGP グレース フル シャットダウンに関する情報

リリース 9.3(1) 以降、BGP はグレースフル シャットダウン機能をサポートしています。この BGP 機能は、BGP **shutdown** コマンドと連携して次のことを行います。

- ルータまたはリンクがオフラインになったときのネットワーク コンバージェンス時間を大幅に短縮します。
- ルータまたはリンクがオフラインになったときに、転送中のドロップされたパケットを削減または排除します。

名前にかかわらず、BGP グレースフル シャットダウンは実際にはシャットダウンを引き起こしません。代わりに、ルータまたはリンクが間もなくダウンすることを、接続されているルータに通知します。

グレースフル シャットダウン機能は、GRACEFUL_SHUTDOWN ウェルノウン コミュニティ（0xFFFF0000 または 65535:0）を使用します。これは、IANA および IETF によって RFC 8326 によって識別されます。この既知のコミュニティは任意のルートにアタッチでき、ルートの他の属性と同様に処理されます。

この機能は、ルータまたはリンクがダウンすることを通知するため、メンテナンス時間帯または計画停止の準備に役立ちます。トラフィックへの影響を制限するには、BGP をシャットダウンする前にこの機能を使用します。

グレースフル シャットダウンの認識とアクティブ化

BGPルータは、すべてのルートの優先事項を、GRACEFUL SHUTDOWN 対応というコンセプトを通し、GRACEFUL_SHUTDOWNコミュニティによって制御できます。グレースフル シャットダウン対応は、デフォルトでイネーブルになっています。これにより、受信側ピアは、GRACEFUL_SHUTDOWN コミュニティを伝える着信ルートを優先しなくなります。一般的な使用例ではありませんが、**graceful-shutdown aware** コマンドを使用して、グレースフル シャットダウン対応を無効にしてから再度有効にすることもできます。

グレースフル シャットダウン対応は、BGP グローバル コンテキストでのみ適用されます。コンテキストの詳細については、グレースフル シャットダウンのコンテキスト（157 ページ）を参照してください。対応のためのオプションは、**activate** という別のオプションと一緒に動作します。このオプションをルートマップに割り当てると、グレースフル シャットダウンのルートをより詳細に制御できます。

グレースフル シャットダウン対応オプションとアクティブ化オプションの協同作用

グレースフル シャットダウンがアクティブな場合、**activate** キーワードを指定した場合にのみ、GRACEFUL_SHUTDOWNコミュニティがルート更新に追加されます。この時点で、コミュニティを含む新しいルート更新が生成され、送信されます。**graceful-shutdown aware** コマンドが設定されると、コミュニティを受信するすべてのルータは、アップデート内のルートの優先を解除します（そのルート優先度を下げます）。**graceful-shutdown aware** コマンドを使用しな

かった場合、BGPはGRACEFUL_SHUTDOWN コミュニティの設定されたルートの優先度を下げません。

この機能がアクティブになり、ルータがグレースフル シャットダウンの対応状態になった場合でも、BGP は引き続き、GRACEFUL_SHUTDOWN コミュニティが有効だとしてルートを考慮します。ただし、これらのルートには、最適パスの計算で最低の優先度が与えられます。代替パスが使用可能な場合は、新しい最適パスが選択され、まもなくダウンするルータまたはリンクに対応するためのコンバージェンスが行われます。

グレースフル シャットダウンのコンテキスト

BGPのグレースフルシャットダウン機能には、機能の影響と使用可能な機能を決定する2つのコンテキストがあります。

コンテキスト	影響	コマンド
グローバル	スイッチ全体と、スイッチによって処理されるすべてのルート。たとえば、GRACEFUL_SHUTDOWN コミュニティを持つすべてのルートを再アドバタイズします。	**graceful-shutdown activate [route-map** ルート マップ**]** **graceful-shutdown aware**
Peer	BGP ピアまたはネイバー間のリンク。たとえば、ピア間のリンクを 1 つだけGRACEFUL_SHUTDOWNコミュニティでアドバタイズします。	**graceful-shutdown activate [route-map** ルート マップ**]**

ルート マップによるグレースフル シャットダウン

グレースフル シャットダウンは、ルート ポリシー マネージャ（RPM）機能と連携して、スイッチの BGP ルータが GRACEFUL_SHUTDOWN コミュニティを使用してルートを送受信する方法を制御します。ルート マップは、インバウンドおよびアウトバウンド方向でコミュニティとのルート更新を処理できます。通常、ルートマップは必要ありません。ただし、必要に応じて、グレースフル シャットダウン ルートの制御をカスタマイズするために使用できます。

通常のインバウンド ルート マップ

通常のインバウンド ルート マップは、BGP ルータに着信するルートに影響します。ルータはデフォルトでグレースフル シャットダウンを認識するため、通常のインバウンド ルート マップはグレースフル シャットダウン機能では一般的に使用されません。

Cisco NX-OS リリース 9.3 (1) 以降を実行している Cisco Nexus スイッチでは、グレースフル シャットダウン機能のインバウンドルートマップは必要ありません。Cisco NX-OSリリース9.3（1）以降には、BGPルータがグレースフルシャットダウン対応である場合に

GRACEFUL_SHUTDOWNコミュニティを持つすべてのルートを自動的に非優先にする、暗黙のインバウンドルート マップがあります。

通常のインバウンド ルート マップは、既知の GRACEFUL_SHUTDOWN コミュニティと一致するように設定できます。これらの着信ルートマップは一般的ではありませんが、使用される場合があります。

- スイッチが 9.3 (1) よりも前の Cisco NX-OS リリースを実行している場合、NX-OS 9.3 (1) には暗黙的なインバウンド ルート マップがありません。これらのスイッチでグレースフル シャットダウン機能を使用するには、グレースフル シャットダウン インバウンド ルートマップを作成する必要があります。ルートマップは、既知のGRACEFUL_SHUTDOWNコミュニティを持つインバウンドルートと一致し、それらを許可し、それらを非優先にする必要があります。着信ルート マップが必要な場合は、9.3 (1) より前のバージョンの NX-OS を実行し、グレースフル シャットダウン ルートを受信しているBGPピアで作成します。
- グレースフル シャットダウン認識をディセーブルにし、一部の BGP ネイバーからの GRACEFUL_SHUTDOWN コミュニティを持つ着信ルートでルータを動作させる場合は、それぞれのピアでインバウンド ルート マップを設定できます。

通常のアウトバウンド ルート マップ

通常のアウトバウンド ルート マップは、BGP ルータが送信するルートの転送を制御します。通常のアウトバウンド ルート マップは、グレースフル シャットダウン機能に影響を与える可能性があります。たとえば、GRACEFUL_SHUTDOWN コミュニティで一致するようにアウトバウンド ルート マップを設定し、属性を設定できます。これは、グレースフル シャットダウン アウトバウンド ルート マップよりも優先されます。

グレースフル シャットダウン アウトバウンド ルート マップ

アウトバウンドグレースフルシャットダウンルートマップは、グレースフルシャットダウン機能のアウトバウンド ルート マップの特定のタイプです。これらはオプションですが、ルート マップに関連付けられているコミュニティ リストがすでにある場合に役立ちます。通常のグレースフル シャットダウン アウトバウンド ルート マップには、特定の属性を設定または変更するための `set` 句のみが含まれています。

アウトバウンド ルート マップは、次の方法で使用できます。

- 既存のアウトバウンド ルート マップをすでに持っている顧客の場合は、より大きいシーケンス番号を持つ新しいエントリを追加し、GRACEFUL_SHUTDOWN ウェルノウンコミュニティで照合し、必要な属性を追加できます。
- **graceful-shutdown activate route-map** *name* オプションを使用してグレースフル シャットダウン アウトバウンドルート マップを使用することもできます。これが一般的な使用例です。

  このルート マップには match 句が必要ないため、ルート マップはネイバーに送信されるすべてのルートで一致します。

ルート マップの優先順位

同じルータ上に複数のルートマップが存在する場合は、次の優先順位が適用されて、コミュニティとのルートの処理方法が決定されます。次の例を考慮してください。60のローカル設定を設定する標準の発信ルートマップ名Redがあるとします。また、Blueという名前のピア グレースフル シャットダウン ルートマップがあり、local-prefが30に設定されているとします。ルート更新が処理されると、RedはBlueを上書きするため、ローカル プリファレンスは60に設定されます。

- 通常の発信ルート マップは、ピア グレースフル シャットダウン マップよりも優先されます。
- ピア グレースフル シャットダウン マップは、グローバル グレースフル シャットダウン マップよりも優先されます。

注意事項と制約事項

BGP グローバル シャットダウンの制限事項と注意事項は、次のとおりです。

- グレースフルシャットダウン機能は、影響を受けるルータの代替ルートがネットワークに存在する場合にのみ、トラフィック損失を回避するのに役立ちます。ルータに代替ルートがない場合は、GRACEFUL_SHUTDOWN コミュニティを伝送するルートが使用可能な唯一のルートであるため、最適パスの計算に使用されます。この状況では、機能の目的が失われます。
- GRACEFUL_SHUTDOWN コミュニティを送信するには、BGP 送信コミュニティの設定が必要です。
- ルート マップの場合:
  - グローバル ルート マップとネイバー ルート マップが設定されている場合、ネイバー単位のルート マップが優先されます。
  - 発信ルート マップは、グレースフル シャットダウン用に設定されたグローバル ルート マップよりも優先されます。
  - 発信ルート マップは、グレースフル シャットダウン用に設定されたピア ルート マップよりも優先されます。
  - レガシー（既存の）インバウンド ルート マップにグレースフル シャットダウン機能を追加するには、次の手順を実行します。
    1. graceful shutdown match 句をルート マップの先頭に追加します。これには、句に低いシーケンス番号（たとえば、シーケンス番号 0）を設定します。
    2. graceful shutdown 句の後に continue ステートメントを追加します。continue ステートメントを省略すると、graceful shutdown 句と一致するルートマップ処理が停止します。シーケンス番号が大きい他の句（たとえば、1以上）は処理されません。

グレースフル シャットダウン タスクの概要

グレースフル シャットダウン機能を使用するには、通常、すべての Cisco Nexus スイッチでグレースフル シャットダウン対応をイネーブルにし、機能をイネーブルのままにします。BGP ルータをオフラインにする必要がある場合は、graceful-shutdown activate を設定します。

次の詳細に、グレースフル シャットダウン機能を使用するためのベスト プラクティスを示します。

ルータまたはリンクをダウンさせるには、次の手順を実行します。

1. グレースフル シャットダウン機能を設定します。
2. ネイバーでベスト パスを確認します。
3. 最適パスが再計算されたら、BGP を無効にする **shutdown** コマンドを発行します。
4. ルータまたはリンクをシャットダウンする必要がある作業を実行します。

ルータまたはリンクをオンラインに戻すには、次の手順を実行します。

1. シャットダウンが必要な作業が完了したら、BGP を再度イネーブルにします（**no shutdown**）。
2. グレースフル シャットダウン機能を無効にします（config モードの **no graceful-shutdown activate**）。

リンクのグレースフル シャットダウンの設定

この作業では、2 つの BGP ルータ間の特定のリンクでグレースフル シャットダウンを設定できます。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **config terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor {** *ipv4-address*|*ipv6-address* **} remote-as** *as-number*
4. **graceful-shutdown activate [route-map** *map-name***]**

手順の詳細

手順

	コマンドまたはアクション	目的

ステップ 1	**config terminal**
	例：
	`switch-1# configure terminal`
	`switch-1(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** *autonomous-system-number*
	例：
	`switch-1(config)# router bgp 110`
	`switch-1(config-router)#`
		ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成または設定します。

ステップ 3	**neighbor {** *ipv4-address*|*ipv6-address* **} remote-as** *as-number*
	例：
	`switch-1(config-router)# neighbor 10.0.0.3 remote-as 200`
	`switch-1(config-router-neighbor)#`
		ネイバーが属する自律システム（AS）を設定します。

ステップ 4	**graceful-shutdown activate [route-map** *map-name***]**
	例：
	`switch-1(config-router-neighbor)# graceful-shutdown activate route-map gshutPeer`
	`switch-1(config-router-neighbor)#`
		ネイバーへのリンクでグレースフル シャットダウンを設定します。また、既知の GRACEFUL_SHUTDOWN コミュニティを使用してルートをアドバタイズし、アウトバウンドルート更新にルートマップを適用します。
		ルートは、デフォルトでグレースフル シャットダウン コミュニティでアドバタイズされます。この例では、ルートは gshutPeer という名前のルート マップを使用して、グレースフル シャットダウン コミュニティを持つネイバーにアドバタイズされます。
		gshut コミュニティを受信したデバイスは、ルートのコミュニティを確認し、オプションでコミュニティを使用してルーティング ポリシーを適用します。

GRACEFUL_SHUTDOWN コミュニティに基づく BGP ルートのフィルタリングとローカルプリファレンスの設定

まだ 9.3(1) を実行していないスイッチには、GRACEFUL_SHUTDOWN コミュニティ名と一致するインバウンドルートマップがありません。したがって、正しいルートを識別して先送りする方法はありません。

9.3(1) よりも前のリリースの NX-OS を実行しているスイッチでは、グレースフル シャットダウン（65535:0）のコミュニティ値と一致するインバウンド ルート マップを設定し、ルートを非優先にする必要があります。

スイッチが 9.3(1) 以降を実行している場合、着信ルートマップを設定する必要はありません。

手順の概要

1. **configure terminal**
2. **ip community list standard** *community-list-name* **seq** *sequence-number* **{ permit | deny }** *value*
3. **route map** *map-tag* **{deny | permit}** *sequence-number*
4. **match community** *community-list-name*
5. **set local-preference** *local-pref-value*
6. **exit**
7. **router bgp** *community-list-name*
8. **neighbor {** *ipv4-address|ipv6-address* **}**
9. **address-family {** *address-family sub family* **}**
10. **send community**
11. **route map** *map-tag* **in**

手順の詳細

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-1# configure terminal`
	`switch-1<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 community list standard** *community-list-name* **seq** *sequence-number* **{ permit | deny }** *value*
	例：
	`switch-1(config)# ip community-list standard GSHUT seq 10 permit 65535:0`
	`switch-1(config)#`
		コミュニティ リストを設定し、よく知られたグレースフル シャットダウン コミュニティ値を持つルートを許可または拒否します。

ステップ 3	**route map** *map-tag* **{deny | permit}** *sequence-number*
	例：
	`switch-1(config)# route-map RM_GSHUT permit 10`
	`switch-1(config-route-map)#`
		ルート マップをシーケンス 10 として設定し、GRACEFUL_SHUTDOWN コミュニティを持つルートを許可します。

ステップ 4	**match community** *community-list-name*
	例：
	`switch-1(config-route-map)# match community GSHUT`
	`switch-1(config-route-map)#`
		IP コミュニティ リスト GSHUT に一致するルートがルート ポリシー マネージャ（RPM）により処理されるように設定します。

	コマンドまたはアクション	目的

ステップ 5	**set local-preference** *local-pref-value*
	例：
	`switch-1(config-route-map)# set local-preference 10`
	`switch-1(config-route-map)#`
		IP コミュニティ リスト GSHUT に一致するルートに、指定されたローカル プリファレンスが与えられるように設定します。

ステップ 6	**exit**
	例：
	`switch-1(config-route-map)# exit`
	`switch-1(config)#`
		ルート マップ設定モードを終了し、グローバル設定モードに戻ります。

ステップ 7	**router bgp** *community-list-name*
	例：
	`switch-1(config)# router bgp 100`
	`switch-1(config-router)#`
		ルータ設定モードを開始し、BGP インスタンスを作成します。

ステップ 8	**neighbor {** *ipv4-address*|*ipv6-address* **}**
	例：
	`switch-1(config-router)# neighbor 10.0.0.3`
	`switch-1(config-router-neighbor)#`
		指定したネイバーのルート BGP ネイバー モードを開始します。

ステップ 9	**address-family {** *address-family sub family* **}**
	例：
	`nxosv2(config-router-neighbor)# address-family ipv4 unicast`
	`nxosv2(config-router-neighbor-af)#`
		ネイバーをアドレス ファミリ（AF）設定モードにします。

ステップ 10	**send community**
	例：
	`nxosv2(config-router-neighbor-af)# send-community`
	`nxosv2(config-router-neighbor-af)#`
		ネイバーとの BGP コミュニティ交換を可能にします。

ステップ 11	**route map** *map-tag* **in**
	例：
	`nxosv2(config-router-neighbor-af)# route-map RM_GSHUT in`
	`nxosv2(config-router-neighbor-af)#`
		ネイバーからの着信ルートにルート マップを適用します。この例では、RM_GSHUT という名前のルート マップは、ネイバーからの GRACEFUL_SHUTDOWN コミュニティを持つルートを許可します。

すべての BGP ネイバーのグレースフル シャットダウンの設定

グレースフル シャットダウン イニシエータのすべてのネイバーに GRACEFUL_SHUTDOWN ウェルノウン コミュニティを手動で適用できます。

すべての BGP ネイバーに対して、グローバル レベルでグレースフル シャットダウンを設定できます。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **graceful-shutdown activate [route-map** *map-name***]**

手順の詳細

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-1# configure terminal`
	`switch-1(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** *autonomous-system-number*
	例：
	`switch-1(config)# router bgp 110`
	`switch-1(config-router)#`
		ルータ コンフィギュレーションモードを開始して、BGP ルーティング プロセスを作成または設定します。

ステップ 3	**graceful-shutdown activate [route-map** *map-name***]**
	例：
	`switch-1(config-router-neighbor)#`
	`graceful-shutdown activate route-map gshutPeer`
	`switch-1(config-router-neighbor)#`
		すべてのネイバーへのリンクのグレースフル シャットダウン ルート マップを設定します。また、既知の GRACEFUL_SHUTDOWN コミュニティを持つすべてのルートをアドバタイズし、ルートマップをアウトバウンド ルート アップデートに適用します。
		ルートはデフォルトで GRACEFUL_SHUTDOWN コミュニティでアドバタイズされます。この例では、ルートが gshutPeer という名前のルートマップを持つコミュニティを持つすべてのネイバーにアドバタイズされます。ルートマップには set 句のみを含める必要があります。
		GRACEFUL_SHUTDOWN コミュニティを受信したデバイスは、ルートのコミュニティを確認し、オプションでコミュニティを使用してルーティング ポリシーを適用します。

GRACEFUL_SHUTDOWNコミュニティを使用したすべてのルートのプリファレンスの制御

Cisco NX-OS では、GRACEFUL_SHUTDOWN コミュニティを持つ着信ルートの優先順位を下げることができます。**graceful shutdown aware** が有効になっている場合、最適パス計算時に、

BGPはコミュニティを伝送するルートを最も低い優先順位と見なします。デフォルトでは、プレファレンスの引き下げが有効になっていますが、このオプションを選択的に無効にすることもできます。

このオプションをイネーブルまたはディセーブルにするたびに、BGPのベストパス計算がトリガーされます。このオプションを使用すると、グレースフルシャットダウンのウェルノウンコミュニティにおける BGP のベストパス計算の動作を柔軟に制御できます。

始める前に

BGPを有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonoums-system*
3. （任意） **no graceful-shutdown aware**

手順の詳細

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-1(config)# config terminal`
	`switch-1(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** *autonoums-system*
	例：
	`switch-1(config)# router bgp 100`
	`switch-1(config-router)#`
		ルータ コンフィギュレーション モードを開始し、BGP ルーティング プロセスを設定します。

ステップ 3	（任意） **no graceful-shutdown aware**
	例：
	`switch-1(config-router)# no graceful-shutdown aware`
	`switch-1(config-router)#`
		このBGPルータでは、GRACEFUL_SHUTDOWN コミュニティを持つすべてのルートに低い優先順位を指定しないという意味です。グレースフルシャットダウン認識機能がディセーブルになっている場合、デフォルトアクションはルートを非優先にします。そのため、コマンドには **no** 形式というオプションが存在しており、これを使用すると、グレースフルシャットダウン ルートは非優先になりません。

GRACEFUL_SHUTDOWN コミュニティのピアへの送信の防止

発信ルート更新にルート属性として追加されたGRACEFUL_SHUTDOWNコミュニティが不要になった場合は、コミュニティを削除して、指定されたネイバーに送信しなくなります。1 つ

の使用例は、ルータが自律システム境界にあり、グレースフルシャットダウン機能が自律システム境界の外部に伝播しないようにする場合です。

GRACEFUL_SHUTDOWN がピアに送信されないようにするには、send community オプションを無効にするか、コミュニティを発信ルート マップから削除します。

次の方法の中から 1 つを選択してください。

- 実行コンフィギュレーションで send-community を無効にします。

  例：

  ```
  nxosv2(config-router-neighbor-af)# no send-community standard
  nxosv2(config-router-neighbor-af)#
  ```

  このオプションを使用すると、スイッチはGRACEFUL_SHUTDOWNコミュニティを受信しますが、発信ルート マップを介してダウンストリーム ネイバーに送信されません。すべての標準コミュニティも送信されません。

- 次の手順に従って、発信ルートマップを介してGRACEFUL_SHUTDOWNコミュニティを削除します。

  1. GRACEFUL_SHUTDOWNコミュニティと一致するIPコミュニティ リストを作成します。
  2. GRACEFUL_SHUTDOWN コミュニティと照合する発信ルート マップを作成します。
  3. **set community-list delete** 句を使用して GRACEFUL_SHUTDOWN コミュニティを削除します。

  このオプションを使用すると、コミュニティ リストはGRACEFUL_SHUTDOWNコミュニティと一致し、許可されます。その後、発信ルートマップはコミュニティと照合され、発信ルート マップから削除されます。他のすべてのコミュニティは、問題なく発信ルートマップを通過します。

グレースフル シャットダウン情報の表示

グレースフル シャットダウン機能に関する情報は、次の **show** コマンドで確認できます。

コマンド	アクション
show ip bgp community-list graceful-shutdown	GRACEFUL_SHUTDOWNコミュニティを持つBGP ルーティング テーブル内のすべてのエントリを表示します。
show running-config bgp	実行中の BGP のデフォルト設定を示します。
show running-config bgp all	グレースフル シャットダウン機能に関する情報など、実行中の BGP 設定のすべての情報を表示します。

コマンド	アクション
show bgp *address-family* **neighbors** *neighbor-address*	機能がピアに設定されている場合、次のように表示されます。 • 指定されたネイバーの graceful-shutdown-activate 機能の状態 • 指定されたネイバーに設定されたグレースフル シャットダウン ルートマップの名前
show bgp process	コンテキストに応じて異なる情報を表示します。 graceful-shutdown-activate オプションがピア コンテキストで設定されている場合、`graceful-shutdown-active` を介して機能の有効または無効状態を示します。 graceful-shutdown-activate オプションがグローバル コンテキストで設定され、graceful-shutdown ルートマップがある場合は、次のように機能の有効状態が表示されます。 • `graceful-shutdown-active` • `graceful-shutdown-aware` • `graceful-shutdown route-map`
show ip bgp *address*	指定されたアドレスについて、次を含む BGP ルーティング テーブル情報を表示します。 • 最適パスとして指定されたアドレスの状態 • 指定されたアドレスが GRACEFUL_SHUTDOWN コミュニティの一部であるかどうか

グレースフル シャットダウンの設定例

次に、グレースフル シャットダウン機能を使用するための設定例を示します。

BGP リンクのグレースフル シャットダウンの設定

次に、ローカル プリファレンスとコミュニティを設定しながらグレースフル シャットダウンを設定する例を示します。

- 指定されたネイバーへのリンクのグレースフル シャットダウン アクティブ化の設定
- ルートへの GRACEFUL_SHUTDOWN コミュニティの追加
- コミュニティとのアウトバウンド ルートに対して set 句のみを使用して gshutPeer という名前のルートマップを設定します。

```
router bgp 100
    neighbor 20.0.0.3 remote-as 200
         graceful-shutdown activate route-map gshutPeer
         address-family ipv4 unicast
              send-community

route-map gshutPeer permit 10
    set local-preference 0
    set community 200:30
```

All-Neighbor BGP リンクのグレースフル シャットダウンの設定

次に例を示します。

- ローカル ルータとそのすべてのネイバーを接続するすべてのリンクに対してグレース フルシャットダウン アクティブ化を設定します。
- GRACEFUL_SHUTDOWN コミュニティをルートに追加しています。
- すべての発信ルートに対して set 句のみを使用して gshutAall という名前のルートマップを設定します。

```
router bgp 200
   graceful-shutdown activate route-map gshutAll

route-map gshutAll permit 10
   set as-path prepend 10 100 110
   set community 100:80

route-map Red permit 10
   set local-pref 20

router bgp 100
   graceful-shutdown activate route-map gshutAll
      router-id 2.2.2.2
         address-family ipv4 unicast
         network 2.2.2.2/32
         neighbor 1.1.1.1 remote-as 100
         update-source loopback0
         address-family ipv4 unicast
            send-community
         neighbor 20.0.0.3 remote-as 200
         address-family ipv4 unicast
            send-community
               route-map Red out
```

この例では、ネイバー 1.1.1.1 に対して `gshutAll` ルート マップが有効になりますが、ネイバー 20.0.0.3 で設定された発信ルートマップ `Red` が優先されるため、ネイバー 20.0.0.3 に対しては有効になりません。

ピアテンプレートでのグレースフル シャットダウンの設定

この例では、ピアセッションテンプレートでグレースフルシャットダウン機能を設定します。これはネイバーによって継承されます。

```
router bgp 200
   template peer-session p1
      graceful-shutdown activate route-map gshut_out
   neighbor 1.1.1.1 remote-as 100
      inherit peer-session p1
      address-family ipv4 unicast
         send-community
```

GRACEFUL_SHUTDOWN コミュニティの使用およびインバウンドルートマップに基づく BGP ルートのフィルタリングとローカル プリファレンスの設定

次に、コミュニティ リストを使用して、GRACEFUL_SHUTDOWN コミュニティを持つ着信ルートをフィルタリングする例を示します。この設定は、Cisco NX-OS 9.3(1) を最小バージョンとして実行していないレガシー スイッチに役立ちます。

次に例を示します。

- GRACEFUL_SHUTDOWN コミュニティを持つルートを許可する IP コミュニティ リスト。
- RM_GSHUT という名前のルート マップは、GSHUT という名前の標準コミュニティ リストに基づいてルートを許可します。
- また、ルートマップは、処理するルートの優先順位を 0 に設定します。これにより、ルータがオフラインになったときに、それらのルートに最適パス計算の優先順位が低くなります。ネイバー (20.0.0.2) からの着信 IPv4 ルートにルート マップが適用されます。

```
ip community-list standard GSHUT permit 65535:0

route-map RM_GSHUT permit 10
   match community GSHUT
   set local-preference 0

router bgp 200
   neighbor 20.0.0.2 remote-as 100
      address-family ipv4 unicast
         send-community
         route-map RM_GSHUT in
```

グレースフル リスタートの設定

グレースフル リスタートを設定し、BGP に対してグレースフル リスタート ヘルパー機能をイネーブルにできます。

（注） Cisco NX-OS リリース 10.1(1) は、より多くの BFD セッションをサポートします。BGP セッションが BFD に関連付けられている場合、ISSU 中にピア接続を維持するために BGP **restart-time** を増やす必要が生じることがあります。

（注） BGP グレースフル リスタートの観点からは、ノードの再起動中にアイドル状態のピアがある場合、最初のベストパスの確立が遅延する可能性があるため、ISSU 中にトラフィック損失が発生する可能性があります。これらのアイドル状態のネイバーをすべて起動するか、それぞれで「shutdown」を構成するか、構成から完全に削除することをお勧めします。

始める前に

BGP をイネーブルにする必要があります（「BGP のイネーブル化」の項を参照）。

VRF を作成します。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. （任意） **timers prefix-peer-timeout** *timeout*
4. **graceful-restart**
5. **graceful-restart** {**restart-time** *time*|**stalepath-time** *time*}
6. **graceful-restart-helper**
7. （任意） **show running-config bgp**
8. （任意） **copy running-config startup-config**

手順の詳細

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)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		自律システム番号を設定して、新しいBGPプロセスを作成します。

ステップ 3	（任意） **timers prefix-peer-timeout** *timeout* 例： `switch(config-router)# timers prefix-peer-timeout 20`	BGPプレフィックスピアのタイムアウト値を設定します（秒単位）。デフォルト値は 90 秒です。 （注） このコマンドは、Cisco NX-OS リリース 9.3(3) 以降でサポートされます。

	コマンドまたはアクション	目的
ステップ 4	**graceful-restart** 例： `switch(config-router)# graceful-restart`	グレースフル リスタートおよびグレースフル リスタート ヘルパー機能をイネーブルにします。このコマンドは、デフォルトでイネーブルになっています。 このコマンドによって、BGP ネイバーセッションの自動通知およびセッション リセットが開始されます。
ステップ 5	**graceful-restart** {**restart-time** *time*|**stalepath-time** *time*} 例： `switch(config-router)# graceful-restart restart-time 300`	グレースフル リスタート タイマーを設定します。 オプション パラメータは次のとおりです。 • **restart-time**：BGP ピアに送信されたリスタートの最大時間。有効な範囲は 1 ～ 3600 秒です。デフォルトは 120 です。 （注） Cisco NX-OS リリース 10.1(1) は、より多くの BFD セッションをサポートします。BGP セッションが BFD に関連付けられている場合、ISSU 中にピア接続を維持するために BGP **restart-time** を増やす必要が生じることがあります。 • **stalepath-time**：BGP が再起動中の BGP ピアからの古いルートを維持する最大時間有効な範囲は 1 ～ 3600 秒です。デフォルトは 300 です。 NX-OS ソフトウェア リリース 10.2(1) では、BGP セッションがグレースフル リスタート機能をアドバタイズするために、BGP セッションの手動リセットが必要です。NX-OS ソフトウェア リリース 10.2(2) 以降では、このコマンドが有効になっている場合、BGP セッションは、BGP セッションを再起動する必要なく、グレースフル リスタート機能を動的にアドバタイズします。
ステップ 6	**graceful-restart-helper** 例： `switch(config-router)# graceful-restart restart-time 300`	BGP GR が無効になっている場合、SSO や BGP プロセスの再起動などの特定の GR 対応イベントが N9K でローカルに発生している間、N9K 自体は必ずしも自身の転送状態を保持しません。ただし、GR ヘルパーとして、GR 機能をアドバタイズして再起動しているピアをサポートします。つまり、N9K は、ピアリングがダウンしたことを検出すると（ホールドタイマーの期限切れまたは通知メッセージの受信以外）、ピアを指すルートを失効させ、ピアの EOR（または失効パス タイムアウト）を待機します。ピ

	コマンドまたはアクション	目的
		アが再起動して N9K とのピアリングを再確立すると、ピアは自身のすべてのルートを再アドバタイズし、N9K は BGP およびルーティング テーブルでそれらのルートを更新します。ピアから EOR を受信するか、または古いパスタイムアウト（どちらか先に発生した方）を受信すると、N9K はそのピアから残りの古いルートをフラッシュします。ヘルパーモードがない場合、N9K は再起動中のリモート ピアから学習したルートを即座にクリアし、トラフィック損失につながる可能性があります。
ステップ 7	（任意） **show running-config bgp** 例： `switch(config-router)# show running-config bgp`	BGP の設定を表示します。
ステップ 8	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、グレースフル リスタートを有効にする例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# graceful-restart
switch(config-router)# graceful-restart restart-time 300
switch(config-router)# copy running-config startup-config
```

仮想化の設定

各 VDC 内で複数の VRF を作成できます。また、各 VRF で同じ BGP プロセスを使用できます。

VDC ごとに 1 つの BGP プロセスを設定できます。各 VDC 内で複数の VRF を作成できます。また、各 VRF で同じ BGP プロセスを使用できます。

1 つの BGP プロセスを設定し、複数の VRF を作成できます。また、各 VRF で同じ BGP プロセスを使用できます。

始める前に

BGP 機能が有効になっていることを確認します

- BGP を有効にする必要があります（「BGP の有効化」の項を参照）。
- 正しい VDC を使用していることを確認します（または **switchto vdc** コマンドを使用します）。

BGPを有効にする必要があります。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **exit**
4. **router bgp** *as-number*
5. **vrf** *vrf-name*
6. **neighbor** *ip-address* **remote-as** *as-number*
7. （任意） **copy running-config startup-config**

手順の詳細

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 RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。

ステップ 3	**exit**
	例：
	`switch(config-vrf)# exit`
	`switch(config)#`
		VRF設定モードを終了します。

ステップ 4	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		自律システム番号を設定して、新しいBGPプロセスを作成します。

ステップ 5	**vrf** *vrf-name* 例：	ルータ VRF設定モードを開始し、この BGP インスタンスと VRF を関連付けます。

	コマンドまたはアクション	目的
	`switch(config-router)# vrf RemoteOfficeVRF` `switch(config-router-vrf)#`	

ステップ 6	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65535`
	`switch(config-router--vrf-neighbor)#`
		リモート BGP ピアの IP アドレスおよび AS 番号を設定します。

ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config-router-vrf-neighbor)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成し、VRF でルータ ID を設定する例を示します。

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config-vrf)# exit
switch(config)# router bgp 65536
switch(config-router)# vrf NewVRF
switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65536
switch(config-router-vrf-neighbor)# copy running-config startup-config
```

拡張 BGP の設定の確認

BGP 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 all [**summary**] [**vrf** *vrf-name*]	すべてのアドレス ファミリについて、BGP 情報を表示します。
show bgp convergence [vrf *vrf-name]*	すべてのアドレス ファミリについて、BGP 情報を表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community** {**regexp** *expression* | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]} [**vrf** *vrf-name*]	BGP コミュニティと一致する BGP ルートを表示します。
show bgp [**vrf** *vrf-name*] {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community-list** *list-name* [**vrf** *vrf-name*]	BGP コミュニティ リストと一致する BGP ルートを表示します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity** {**regexp** *expression* | **generic** [**non-transitive** | **transitive**] *aa4:nn* [**exact-match**]} [**vrf** *vrf-name*]	BGP 拡張コミュニティと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity-list** *list-name* [**exact-match**]} [**vrf** vrf-name]	BGP 拡張コミュニティ リストと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity-list** *list-name* [**exact-match**]} [**vrf** vrf-name]	BGP ルート ダンプニングの情報を表示します。ルート フラップ ダンプニング情報を消去するには、**clear bgp dampening** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**dampening dampened-paths** [**regexp** *expression*]} [**vrf** vrf-name]	BGP ルート ヒストリ パスを表示します。
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **filter-list** *list-name* [**vrf** *vrf-name*]	BGP フィルタ リストの情報を表示します。
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*]	BGP ピアの情報を表示します。これらのネイバーを消去するには、**clear bgp neighbors** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**nexthop** | **nexthop-database**} [**vrf** *vrf-name*]	BGP ルート ネクスト ホップの情報を表示します。
show bgp paths	BGP パス情報を表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **policy** *name* [**vrf** *vrf-name*]	BGP ポリシー情報を表示します。ポリシー情報を消去するには、**clear bgp policy** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **prefix-list** *list-name* [**vrf** *vrf-name*]	プレフィックス リストと一致する BGP ルートを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **received-paths** [**vrf** *vrf-name*]	ソフト再構成用に保管されている BGP パスを表示します。
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **regexp** *expression* [**vrf** *vrf-name*]	AS_path 正規表現と一致する BGP ルートを表示します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **route-map** *map-name* [**vrf** *vrf-name*]	ルート マップと一致する BGP ルートを表示します。
show bgp peer-policy *name* [**vrf** *vrf-name*]	BGP ピア ポリシー情報を表示します。
show bgp peer-session *name* [**vrf** *vrf-name*]	BGP ピア セッション情報を表示します。
show bgp peer-template *name* [**vrf** *vrf-name*]	BGP ピア テンプレート情報を表示します。ピア テンプレートのすべてのネイバーを消去するには、**clear bgp peer-template** コマンドを使用します。
show bgp process	BGP プロセス情報を表示します。
show bgp {**ipv4** | **ipv6**} **unicast neighbors** *interface*	指定されたインターフェイスの BGP ピアに関する情報を表示します。
show ip bgp neighbors *interface-name*	BGP ピアとして使用されるインターフェイスを表示します。
show ip route *ip-address* **detail vrf all | i bw**	リンク帯域幅の EXTCOMM フィールドを表示します。出力の bw：xx（bw：40 など）は、BGP ピアが帯域幅付きの BGP 拡張属性を送信していることを示します（重み付け ECMP の場合）。
show {**ipv4** | **ipv6**} **bgp** *options*	BGP のステータスと構成情報を表示します。
show {**ipv4** | **ipv6**} **mbgp** *options*	BGP のステータスと構成情報を表示します。
show ipv6 routers interface *interface*	IPv6 ICMP ルータ アドバタイズメントによって学習されたリモート IPv6 ルータのリンク ローカル アドレスを表示します。

コマンド	目的
show running-configuration bgp	現在実行中の BGP コンフィギュレーションを表示します。

BGP 統計情報のモニタリング

BGP の統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show bgp {**ipv4** | **vpnv4** | **vpnv6** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **flap-statistics** [**vrf** *vrf-name*] **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **flap-statistics** [**vrf** *vrf-name*]	BGP ルートフラップの統計情報を表示します。これらの統計情報をクリアするには、**clear bgp flap-statistics** コマンドを使用します。
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **unicast injected-routes** **show bgp** {**ipv4** | **ipv6**} **unicast injected-routes**	ルーティング テーブルに挿入されたルートを表示します。
show bgp sessions [**vrf** *vrf-name*]	すべてのピアの BGP セッションを表示します。これらの統計情報をクリアするには、**clear bgp sessions** コマンドを使用します。
show bgp statistics	BGP 統計情報を表示します。

設定例

この例は、個々の BGP ネイバーの BFD をイネーブルにする方法を示します。

```
router bgp 400
  router-id 2.2.2.2
  neighbor 172.16.2.3
    bfd
    remote-as 400
    update-source Vlan1002
    address-family ipv4 unicast
```

この例は、BGP プレフィックス ピアの BFD をイネーブルにする方法を示します。

```
router bgp 400
  router-id 1.1.1.1
  neighbor 172.16.2.0/24
    bfd
    remote-as 400
    update-source Vlan1002
    address-family ipv4 unicast
```

プレフィックス ベース ネイバーの MD5 認証を設定する例を示します。

```
template peer BasePeer-V6
    description BasePeer-V6
    password 3 f4200cfc725bbd28
    transport connection-mode passive
    address-family ipv6 unicast
template peer BasePeer-V4
    bfd
    description BasePeer-V4
    password 3 f4200cfc725bbd28
    address-family ipv4 unicast
--
    neighbor fc00::10:3:11:0/127 remote-as 65006
      inherit peer BasePeer-V6
    neighbor 10.3.11.0/31 remote-as 65006
      inherit peer BasePeer-V4
```

次に、ネイバー ステータスの変化に関するメッセージをグローバルに有効にし、特定のネイバーについてはメッセージを抑制する方法を示します。

```
router bgp 65100
   log-neighbor-changes
     neighbor 209.165.201.1 remote-as 65535
       description test
       address-family ipv4 unicast
         soft-reconfiguration inbound
          disable log-neighbor-changes
```

関連項目

BGP の詳細については、次の項目を参照してください。

- 基本的 *BGP* の設定
- *Route Policy Manager* の設定

その他の参考資料

BGP の実装に関連する詳細情報については、次の項を参照してください。

MIB

MIB	MIB のリンク
BGP に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

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。