



ITD の構成

この章では、Cisco NX-OS デバイスで Intelligent Traffic Director (ITD) を構成する方法について説明します。

- [ITDについて \(1 ページ\)](#)
- [ライセンス要件 \(23 ページ\)](#)
- [Guidelines and Limitations for ITD, on page 23](#)
- [ITD サポート サマリー \(28 ページ\)](#)
- [ITD のデフォルト設定 \(31 ページ\)](#)
- [ITD の構成 \(31 ページ\)](#)
- [ITD レイヤ 3 構成の確認 \(51 ページ\)](#)
- [ITD Layer-2 構成の確認 \(52 ページ\)](#)
- [ITD の構成例 \(52 ページ\)](#)
- [関連資料 \(100 ページ\)](#)

ITDについて

Intelligent Traffic Director (ITD) は、レイヤ 3 およびレイヤ 4 のトラフィック分散、ロードバランシング、およびリダイレクトのためのスケーラブルなアーキテクチャを構築できる、インテリジェントなハードウェアベースのマルチテラビット ソリューションです。

ITD のメリット :

- ライン レートでのマルチテラビット ソリューション
- エンドデバイスへの透過性とステートレス プロトコルのメリット
- Web Cache Communication Protocol (WCCP) やポリシーベースのルーティングなどの代替機能のための複雑さとアーキテクチャのスケーリングの軽減
- プロビジョニングが簡素化され導入が容易
- レガシー サービス アプライアンスは新しいものと共存できます
- 高価な外部ロードバランサの要件を削除します。

- デバイスと Cisco NX-OS スイッチ間の認証 / 統合 / 認定が不要。
- 大幅な OPEX 削減の順序：構成の簡素化、展開の容易さ
- サービス モジュールまたは外部 L4 ロードバランサは不要すべての Nexus ポートをロードバランサとして使用可能

ITD の機能：

- ワイヤスピードでのハードウェアベースのマルチテラビット / 秒 L3 / L4 ロードバランシング
- ゼロ遅延のロードバランシング
- ラインレート トラフィックを任意のデバイスにリダイレクト、たとえば web cache エンジン、Web アクセラレータ エンジン (WAE)、ビデオキャッシュ、など)
- ファイアウォール、侵入防御システム (IPS)、または Web アプリケーションファイアウォール (WAF)、Hadoop クラスタなどのデバイスのクラスタを作成する機能
- IP スティックネス
- ワイヤスピードでのハードウェアベースのマルチテラビット / 秒 L3 / L4 ロードバランシング
- ゼロ遅延のロードバランシング
- ラインレート トラフィックを任意のデバイスにリダイレクト、たとえば web cache エンジン、Web アクセラレータ エンジン (WAE)、ビデオキャッシュ、など)
- ファイアウォール、侵入防御システム (IPS)、または Web アプリケーションファイアウォール (WAF)、Hadoop クラスタなどのデバイスのクラスタを作成する機能
- IP スティックネス
- 回復力 (回復力のある ECMP など)、一貫したハッシュ
- 仮想 IP ベースの L4 ロードバランシング
- ノード間で加重負荷分散と Failaction がサポートされています
- 多数のデバイス / サーバーへの負荷分散
- リダイレクトおよびロードバランシングと同時の ACL
- 双方向のフロー一貫性。A->B および B->A からのトラフィックは同じノードに行きます
- サーバ/アプライアンスを Nexus スイッチに直接接続する必要なし
- IP SLA ベースのプロープを使用したサーバー / アプライアンスのヘルスの監視
- N+M 冗長 (N ノード数、M ホットスタンバイ数)
- サーバー / アプライアンスの自動障害処理
- VRF サポート、vPC サポート

- IPv4 と IPv6 の両方をサポート（すべてのプラットフォームは IPv6 をサポートしていません）
- ITD 機能によるスーパーバイザ CPU への負荷の追加なし
- 無制限のフロー数を処理。
- 無停止でのノードの追加または削除
- 同時リダイレクトと負荷分散
- 同じスイッチ内の複数の ITD サービス間でのレート共有

使用例：

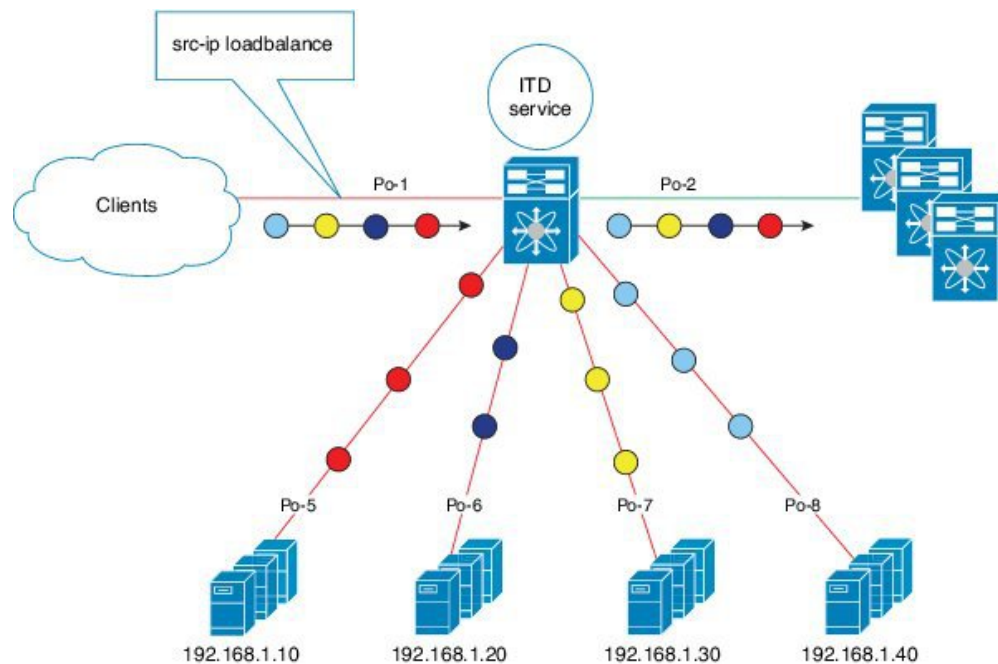
- ファイアウォールのクラスタへの負荷分散。
- NX-OS デバイスへのロードバランシングにより、IPS、IDS、および WAF を拡張します。
- 低コストの VM / コンテナ ベースの NFV にロードバランシングすることにより、NFV ソリューションを拡張します。
- WAAS / WAE ソリューションをスケールアップします。Wide Area Application Services (WAAS) または Web Accelerator Engine (WAE) ソリューションのトラフィック リダイレクトメカニズム
- VDS-TC (ビデオ キャッシュ) ソリューションのスケールアップ
- トラフィックを L7LB に分散することにより、レイヤ 7 ロードバランサーをスケールアップします。
- ECMP またはポートチャネルを置き換えて、再ハッシュを回避します。ITD は復元力があり、ノードの追加 / 削除 / 失敗時に再ハッシュを引き起こしません
- DSR (Direct Server Return) モードでのサーバー負荷分散
- NX-OS デバイスへのロードバランシングにより、NG 侵入防御システム (IPS) と Web アプリケーションファイアウォール (WAF) を拡大します。
- レイヤ 5 からレイヤ 7 のロードバランサーへの負荷分散

展開モード

ワンアーム展開モード

ワンアーム展開モードでサーバーをスイッチに接続できます。このトポロジでは、サーバーはクライアントトラフィックまたはサーバートラフィックの直接パスに存在しないため、既存のトポロジやネットワークを変更することなく、サーバーをネットワークに接続できます。

図 1: ワンアーム展開モード



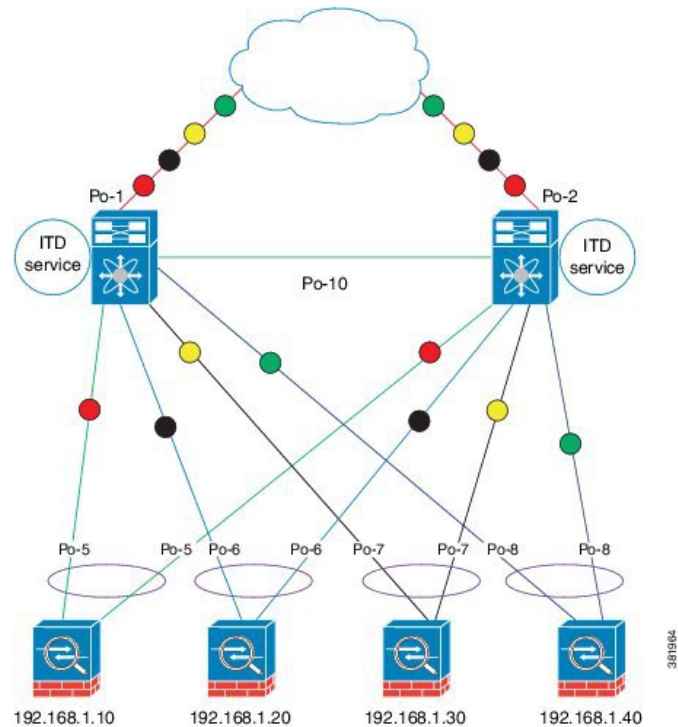
vPC でのワンアーム展開モード

ITDは、仮想ポートチャネル（vPC）に接続されたアプライアンスプールをサポートします。ITD サービスは各スイッチで実行されます。ITD は、フローがノードを通過する一貫したトラフィックを得られるように各スイッチをプログラムします。



(注) VPC シナリオ（ITD NAT を使用しない）に failaction バケット配布を使用して、VPC 経由で到達可能なノードの障害時にピア間で一貫した動作を維持することをお勧めします。

図 2: vPC でのワンアーム展開モード



サンドイッチ展開モード

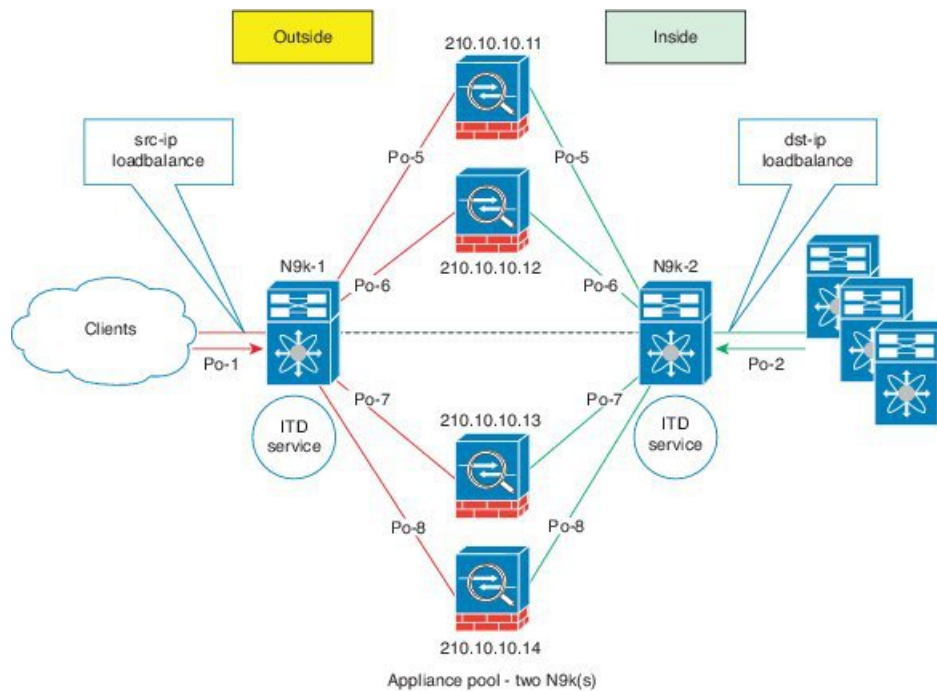
サンドイッチ展開モードでは、2 台のスイッチを使用してトラフィックをステートフルに処理します。

このモードの主な要件は、フローの転送トラフィックとリバーストラフィックの両方が同じアプライアンスを通過しなければならないことです。サンドイッチ展開の例としては、クライアントとサーバ間のトラフィックが同じアプライアンスを通過する必要があるファイアウォールおよびロードバランサの展開があります。

主な機能は次のとおりです。

- ネットワーク セグメントごとの ITD サービス（外部ネットワーク用に 1 つの ITD サービスおよび内部ネットワーク用にもう 1 つの ITD サービス）。
- 入力方向の外部に接続するインターフェイス上で ITD サービスが動作するソース IP アドレス ロードバランシング スキーム。
- 入力方向のサーバに接続するインターフェイスで ITD サービスが動作する宛先 IP アドレスのロードバランシング スキーム。
- ユーザー定義のアクセス リスト（ACL を含む）が外部ネットワークの ITD サービスで使用されている場合、逆の ACE ルールを持つアクセス リストを作成し、内部ネットワークの ITD サービスのユーザー ACL として適用する必要があります。

図 3: サンドイッチ展開モード



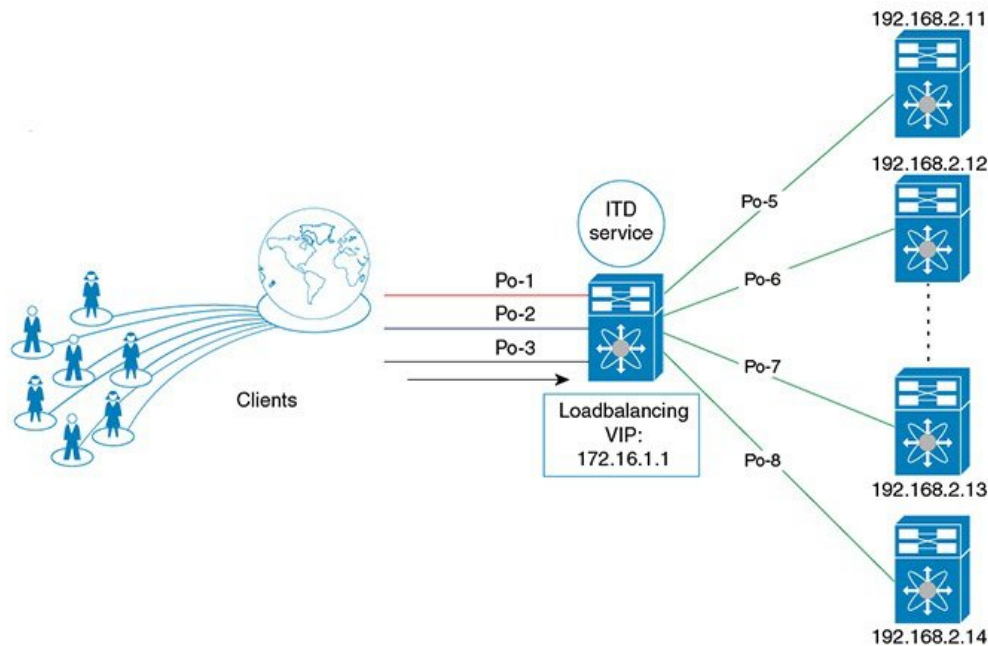
サーバー ロードバランシング展開モード

ITD サービスは、スイッチで仮想 IP (VIP) をホストするように構成できます。VIP を宛先とするインターネットトラフィックの負荷は、複数のアクティブなノードに分散されます。ITD サービスはステートフルロードバランサではありません。



(注) 各スイッチで同様の方法で、ITD サービスを手動で設定する必要があります。

図 4: VIP を使用した ITD 負荷分散



宛先 NAT

ネットワークアドレス変換 (NAT) は、ロードバランシング、ファイアウォール、およびサービス アプライアンスで一般的に導入されている機能です。接続先 NAT は、ロードバランシングで使用される NAT のタイプの 1 つです。

接続先 NAT のメリット

ITD 展開で NAT を使用するメリットは次のとおりです。

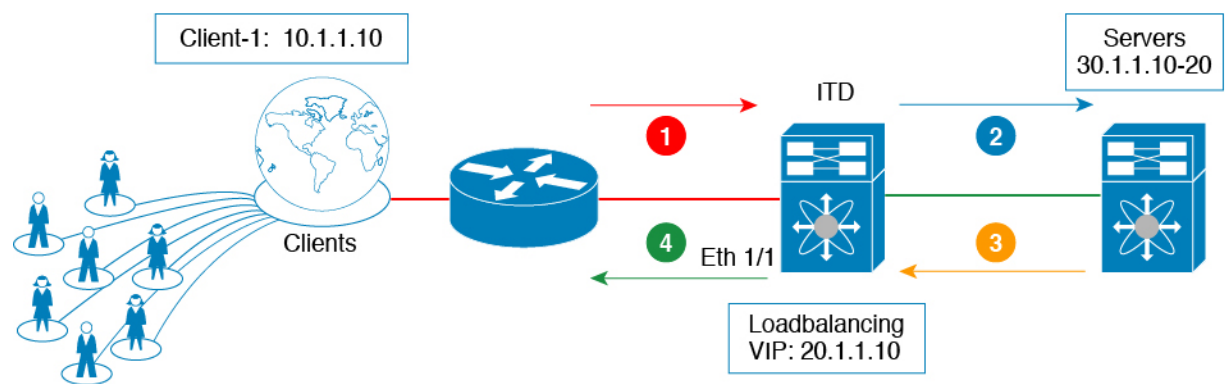
- DSR (Direct Server Return) モードの展開のように、サーバープール内のすべてのサーバーが仮想 IP アドレスをホストする必要はありません。
- サーバー IP を認識する必要がないクライアントは、常にトラフィックを仮想 IP アドレスに送信します。
- ロードバランサはサーバーの障害を検出し、クライアントがプライマリサーバーのステータスを認識しなくても、トラフィックを適切なサーバーにリダイレクトします。
- NAT は、クライアントから実サーバーの IP を隠すことでセキュリティを提供します。
- NAT により、異なるサーバープール間で実サーバーを移動する際の柔軟性が向上します。

さまざまなタイプの NAT の中で、接続先 NAT は、次のメリットがあるため、負荷分散で一般的に展開されます。

- 送信元またはクライアントから仮想 IP アドレスへのトラフィックは書き換えられ、サーバーにリダイレクトされます。
- 送信先またはクライアントから宛先またはサーバーへのトラフィック（転送パス）は、次のように処理されます。送信先またはクライアントから仮想 IP アドレスへのトラフィックは、ソースから接続先またはサーバーへのトラフィックとして変換およびリダイレクトされます。
- 接続先から送信元またはクライアントへのトラフィック（リバースパス）は、仮想 IP アドレスを送信元 IP アドレスとして再変換されます。

次の図は、仮想 IP アドレスを使用した NAT を示しています。

図 5: 仮想 IP アドレスによる NAT



| Step | dst-mac | src-mac | src-ip | dst-ip |
|------|------------|------------|-----------|-----------|
| 1 | Nexus MAC | Router MAC | 10.1.1.10 | 20.1.1.10 |
| 2 | Server MAC | Nexus MAC | 10.1.1.10 | 30.1.1.10 |
| 3 | Nexus MAC | Server MAC | 30.1.1.10 | 10.1.1.10 |
| 4 | Router MAC | Nexus MAC | 20.1.1.10 | 10.1.1.10 |

307357

ポートアドレス変換 (PAT)

PAT では、実際のアドレスおよび送信元ポートが 1 つのマッピングアドレスおよび固有のポートに変換されることによって、複数の実際のアドレスが 1 つのマッピング IP アドレスに変換されます。使用できる場合、実際の送信元ポート番号がマッピングポートに使用されます。ただし、実際のポートが使用できない場合は、デフォルトで、マッピングポートは実際のポート番号と同じポート範囲（0～511、512～1023、および 1024～65535）から選択されます。PAT では単一のマッピング先のアドレスを使用するため、ルーティング可能なアドレスの使用を抑えることができます。

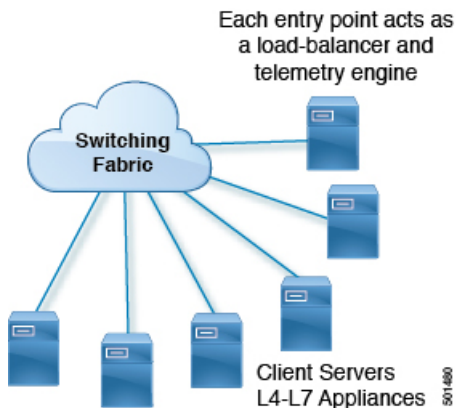
VXLAN 上の ITD

単一のスイッチソリューションであった ITD は、VxLAN ファブリックのロードバランサーとして機能するようになりました。

プログラム可能なファブリックでは、サーバー、仮想マシン（VM）、およびコンテナ（特定のサービスに固有）をファブリック全体に分散させ、さまざまな ToR またはリーフスイッチに接続できます。ITD Over VXLAN 機能により、ファブリック全体に分散されたサーバーへのロードバランシングが可能になります。

ITD Over VXLAN により、ファブリックは大規模なロードバランサーとして機能し、大規模なテレメトリと分析を提供できるようになります。ITD Over VXLAN をロードバランサーとして使用すると、ファブリック内の任意の場所にあるレイヤ4アプライアンスとレイヤ7アプライアンス間を接続できます。これは、図「ファブリック全体のロードバランシング」に示されています。

図 6: ファブリック全体のロードバランシング



データベースサーバー、アプリケーションサーバー、Webサーバー、ファイアウォール、WAAS、IPS、IDS、およびビデオキャッシュを含む多数のクライアント（ローカルおよびボーダーリーフを越えて）がある場合があります。トラフィックの高低に関する情報を含む、ファブリック内の各デバイスから各ファイアウォール、WAAS、IPS、IDS、およびサーバーに流れるトラフィックに関する情報は非常に貴重です。

ITD Over VXLAN は、クライアントとサーバーまたはレイヤ4およびレイヤ7サービス間のパス上にあり、トラフィック情報を認識します。この情報を使用して、貴重なトラフィック分析とテレメトリを提供します。

ロードバランシング機能では、仮想IP（VIP）が、DCファブリック全体に分散された物理サーバーファームによって提供されるサービスを抽象化します。さまざまなクライアント（ローカルからファブリックへ、またはリモートロケーションから）が特定のサービスの要求を送信すると、これらの要求は常にこれらのサーバーのVIPに送信されます。

ToR またはリーフスイッチでは、ITD は送信元 IP アドレスのビットとマスク、宛先 IP アドレス（仮想 IP アドレス）、および関連するレイヤ3またはレイヤ4フィールドを照合して、これらの要求をサーバー間で負荷分散します。

ITD Over VXLAN は、デバイスグループ内のサーバー（ノード）のクラスタを構成するためのインフラストラクチャを提供します。バケット（ビットマスク）に基づいてクライアントトラフィックと ITD サービスで構成されたテナント SVI を分離します。ノード（サーバー）とバケットの定義済みクラスタに基づいて、ITD は、クライアント IP トラフィックをバケット

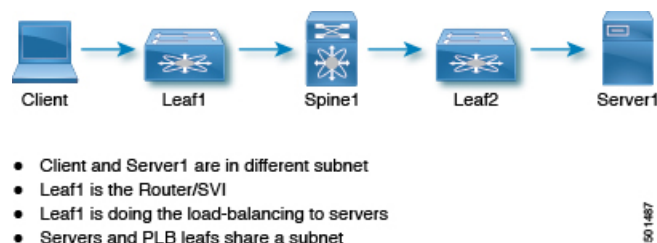
マスクに一致させるルールを自動的に作成し、一致したトラフィックを特定のサーバーノードにリダイレクトします。

サーバーが応答しなくなった場合や動作不能になった場合、ITDはクライアントトラフィックを非動作ノードから単一または構成済みのスタンバイノードのグループに自動的に切り替えます。トラフィックの割り当ては、フローをスタンバイノードに自動的に変更することによって実現されます。

ITD Over VXLANは現在、Direct Server Return (DSR) の概念と機能を使用しているため、サーバーの応答がクライアントに直接送信されます。これはファブリックに依存しませんが、現在 VXLANEVPN ファブリックでサポートされており、VXLAN 経由の PBR をサポートする Cisco Nexus 9000 シリーズスイッチで現在サポートされています。

ITD Over VXLAN は、ライン レートの速度で実現されます。

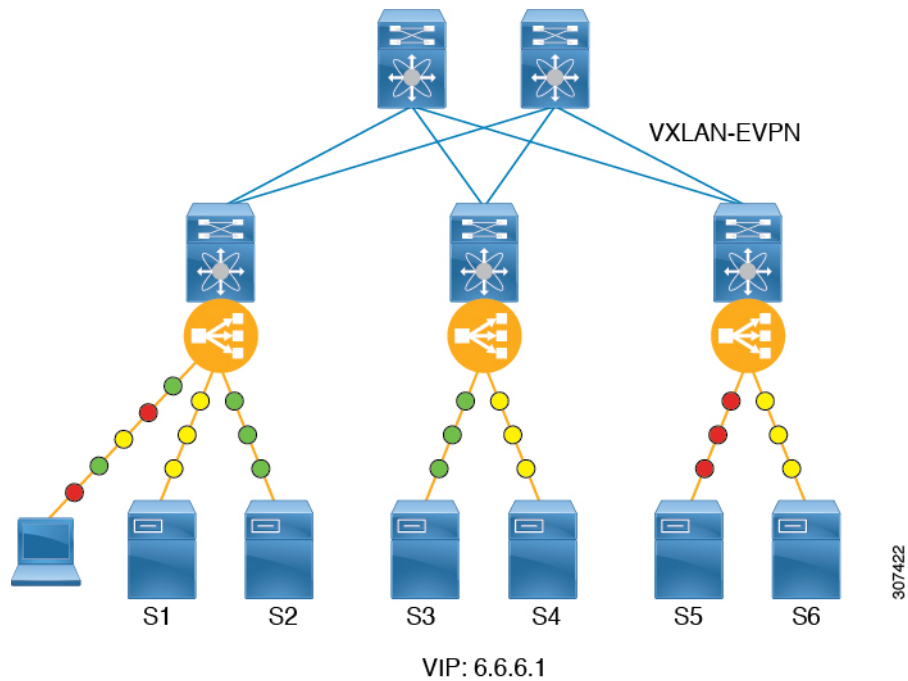
図 7: Direct Server Return



VXLAN トポロジを介した ITD の設定の概要

ToR スイッチでの VXLAN 経由の ITD 設定の概要は次のとおりです。

- 負荷分散サーバーを特定し、デバイス グループを作成します。
- グループの ITD サービス インスタンスを作成し、以下を完了します。
 - 着信する ITD Over VXLAN トラフィックに仮想 IP アドレス (VIP) を関連付けます。VIP は、デバイス グループ内のサーバーを表します。
 - 他の負荷分散構成を有効にします。
 - サービスをアクティブにする必要があるインターフェイスを、サービスの入インターフェイスとして構成します。ITD サービスをイネーブルにします。
 - サーバー (ITD ノード) が接続されているすべてのリーフスイッチに同一の ITD 設定を適用します。これらのリーフスイッチで、このサービスの入インターフェイスとして L3 VNI を設定します。ITD サービスをイネーブルにします。



VXLAN 上での ITD のメリット

- ファブリック内の任意の場所に分散されたサーバー / VM / コンテナの負荷分散
- ハードウェアに依存しない
- 直接接続されたノードのデータプレーン内のノードのヘルスマonitoringとプローブの要約。
- 分析とテレメトリは、サーバー（つまり、VM / コンテナの生成）およびアプライアンス（エラスティックデータセンター）の容量をいつ / どのように拡大するかについての詳細を提供します。
- エラスティック データセンターを構築します。
- VXLAN ネットワーク識別子（VNI）インターフェイス間の負荷分散。
- ファブリック内の複数のスイッチ間でのロードバランシングの同期。
- 障害情報の自動同期。
- 推奨システム
- 可能なすべてのデータセンタートポロジを備えた VXLAN-EVPN ファブリックで動作します。

レイヤ2ロードバランシングについて

レイヤ2 (ITD-L2) ロードバランシングは、Cisco Nexus スイッチでのレイヤ2トラフィック分散、ロードバランシング、およびリダイレクトのためのハードウェアベースのマルチテラビットソリューションです。



(注) ITD-L2 機能は、Cisco 9500 EX/FX ラインカードではサポートされていません。

ITD-L2 は、単一の論理リンクを作成する複数の物理リンクの集合体です。複数の物理リンクをポートグループにバンドルして、帯域幅（複数の物理リンクの集合体）と冗長性を向上させることができます。

レイヤ2内の1つのポートに障害が発生すると、トラフィックはレイヤ2の残りのポートに切り替わります。

ITD-L2 を使用すると、透過モードアプライアンスのクラスタを作成できます。

レイヤ2ロードバランシング機能

ITD-L2 の機能は次のとおりです。

- ラインレートでのマルチテラビットソリューション
- プロビジョニングが簡素化され導入が容易
- エンドデバイスへの透過性とステートレスプロトコルのメリット
- 高価な外部ロードバランサの要件を削除します。

ITD レイヤ2ロードバランシングのメリット

ITD レイヤ2ロードバランシングのメリットは次のとおりです。

- 同時リダイレクトおよびロードバランシング
- IP スティッキ性および復元力
- ポートのヘルス モニタリング
- 高価な外部ロードバランサの要件を削除します。
- ハッシングは、配線やポートの番号付けに依存しません
- スイッチのすべてのポートは、負荷分散とトラフィックのリダイレクトに使用されます

展開使用例

ITD-L2 機能の展開使用の例は次のとおりです。

- ファイアウォールのプールへの負荷分散。

- VDS-TC (ビデオ キャッシュ) ソリューションをスケールリングします。
- トランスペアレント モードのデバイスをスケールリングします。

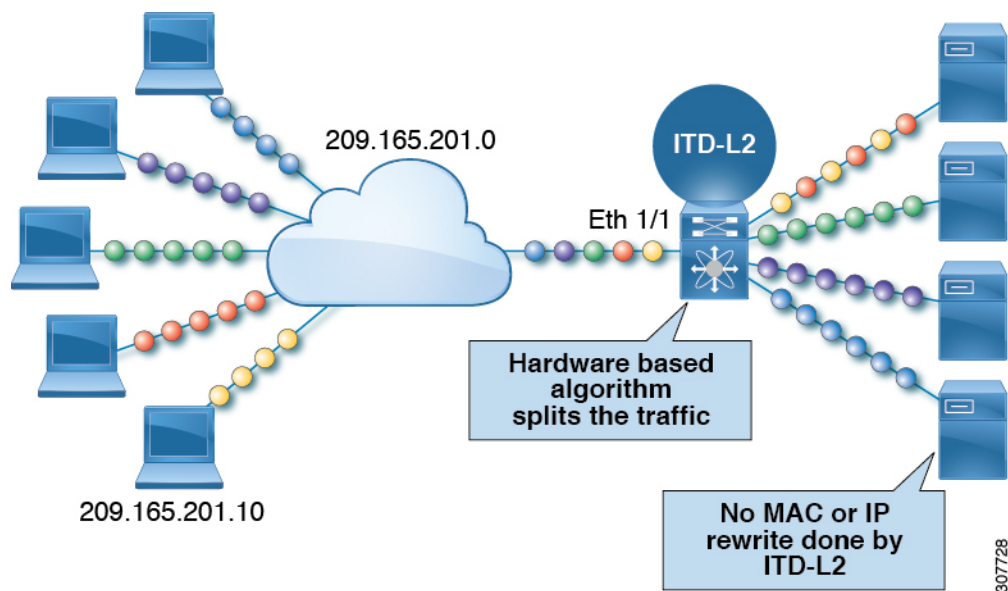
ITD-L2 のトポロジの例

このセクションには、次の例が表示されます。

- ITD-L2 の基本トポロジ
- ITD-L2 構成の使用例
- 回復力のあるハッシュの失敗アクション

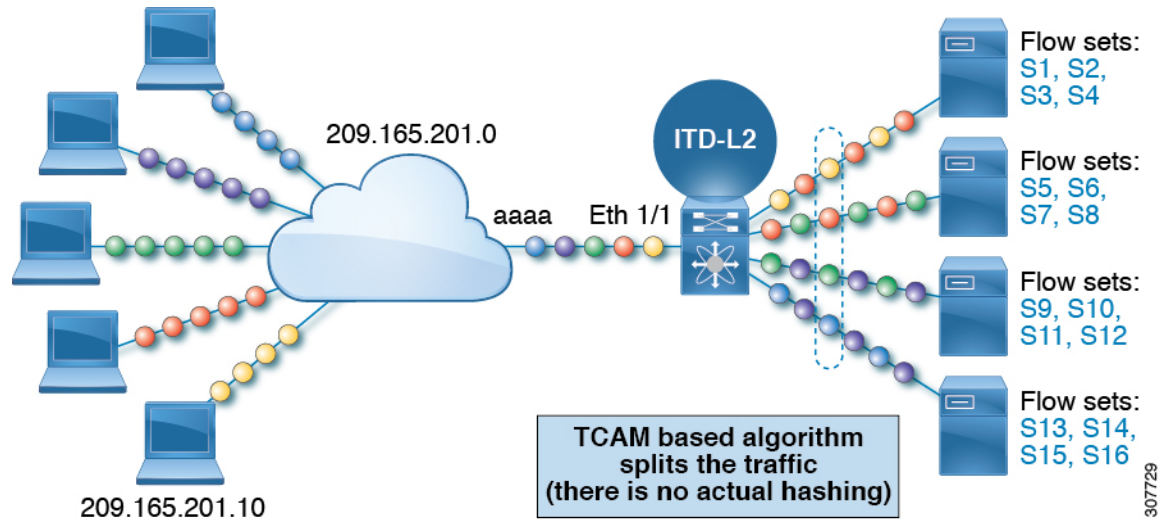
ITD-L2 機能を使用して、監視ネットワークで使用されるアプライアンスへのトラフィックを負荷分散できます。次の図は、IPS や IDS デバイスなどのトラフィックを負荷分散する必要があるアプライアンスにトラフィックが送信される基本的なトポロジを示しています。

図 8: レイヤ 2 ロード バランシングの標準トポロジ



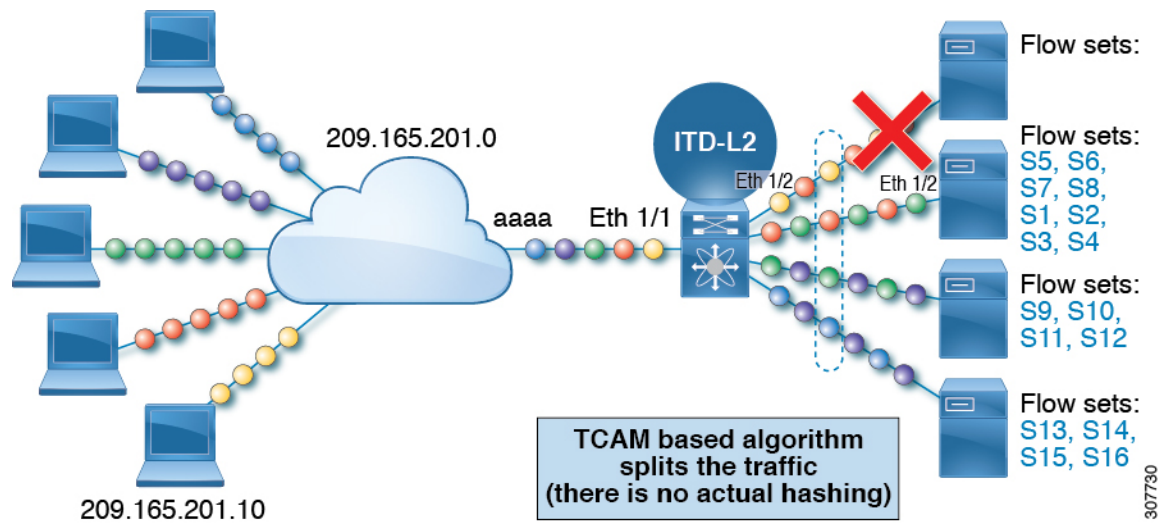
次の例は、トラフィックが本番環境から監視環境に及ぶネットワークでの ITD-L2 の一般的な使用例を示しています。この例では、Cisco Nexus Data Broker を使用して、監視トラフィックのコピーを送信し、監視ネットワークをスケールリングしています。

図 9: レイヤ 2 ロード バランシング 構成の使用例



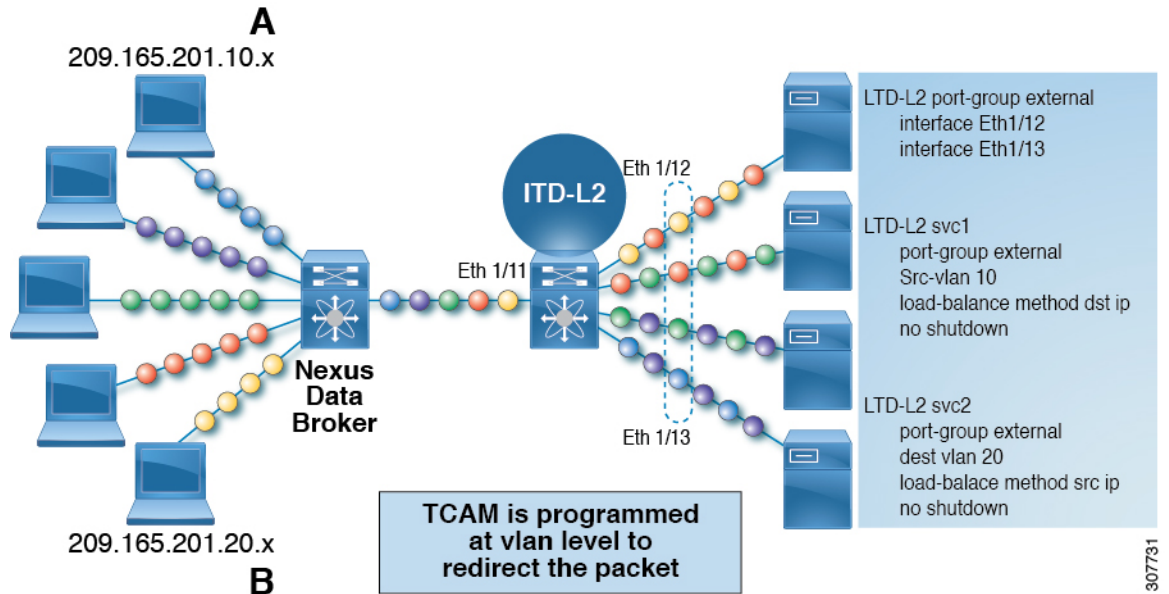
次の例は、ITD-L2 構成の失敗アクションを示しています。

図 10: ITD-L2 構成の Fail-Action



次の例は、弾力性のあるハッシュを使用した ITD-L2 構成の失敗アクションを示しています。

図 11: 回復力のあるハッシュを使用した ITD-L2 構成の失敗アクション



307731

レイヤ2 ロードバランシングの前提条件

レイヤ2 ロードバランシングには、次の前提条件があります。

- 十分な TCAM サイズが VACL に割り当てられていることを確認する必要があります。TCAM サイズを確認するには、**sh hardware access-list tcam region** コマンドを使用します。適切な TCAM サイズが割り当てられていない場合は、**hardware access-list tcam region VACL<256の倍数のサイズ>** コマンドを使用して、適切な TCAM サイズを割り当てます。

デバイス グループ (Device Groups)

ノードは、トラフィックを負荷分散できる物理サーバー、仮想サーバー、またはサービスアプライアンスにすることができます。これらのノードはデバイス グループの下にグループ化され、このデバイス グループをサービスにマップできます。

ITD はデバイス グループをサポートします。デバイス グループを構成するときは、次を指定できます。

- デバイス グループのノード
- デバイス グループのプロープ

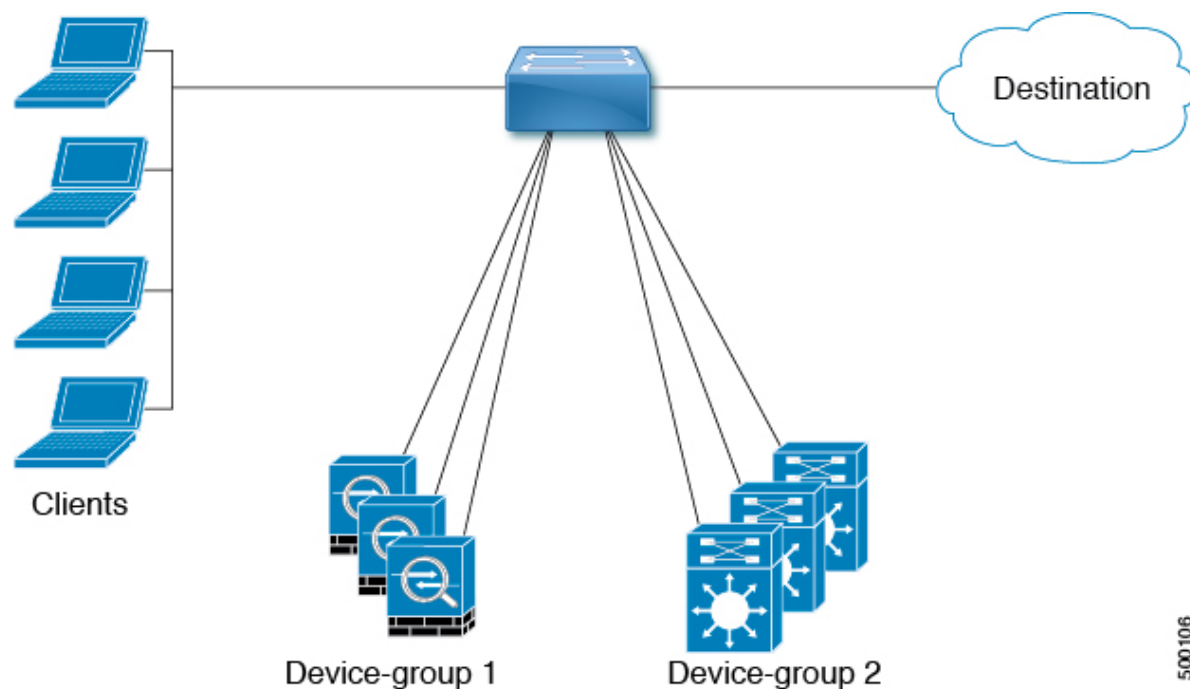
プロープは、デバイス グループ レベルまたはノード レベルで構成できます。ノードレベルのプロープを行う場合、それぞれのノードは自身のプロープで構成可能なため、ノードごとにさらにカスタマイズすることができます。ノードレベルのプロープは、障害状態について各ノードを別々に監視する必要があるシナリオで役立ちます。

ITD サービス内の複数のデバイス グループ

デバイスグループがサポートされています（下図を参照してください）。ITD サービスは、さまざまなデバイスグループを指すさまざまなシーケンスを持つ単一のルートマップを生成します。

各デバイスグループは、異なるサービスを必要としますが、同じ入力インターフェイスに到着する異なるタイプのトラフィックを表します。インターフェイス上のトラフィックは、仮想IPアドレスに基づいて適切なデバイスグループにリダイレクトされます。同じインターフェイスで ITD サービスごとに複数のデバイスグループをサポートすると、ITD を拡張できます。

図 12: ITD サービス内の複数のデバイス グループ



ITD サービスで複数のデバイスグループを設定する方法を示す構成例については、を参照してください。

VRF のサポート

ITD サービスは、デフォルト VRF でもデフォルト以外の VRF でも構成できます。

ITD サービスでトラフィックをリダイレクトするには、入力インターフェイスおよびデバイスグループノードのすべてが同じ VRF に属している必要があります。設定済み VRF で、関連するデバイスグループのすべての入力インターフェイスおよびノードメンバーが到達可能であることを確認する必要があります。

ルータ ACL

スイッチは、ITD を使用したルータ アクセス コントロール リスト (RACL) をサポートしません。

同じ入力インターフェイスで ITD と RACL を構成できます。TCAM にダウンロードされる構成結果の RACL は、ITD によって生成された ACL とユーザ構成 RACL を合わせた成果物です。RACL で構成された `permit` ステートメントと `deny` ステートメントは、ITD によって作成された ACL 許可およびリダイレクト エントリと結合されます。この機能により、選択したトラフィックのフィルタリングおよび負荷分散を行うことができます。



- (注)
- ITD 入力インターフェイスで RACL を構成すると、ITD 統計は機能しません。
 - 現用系な ITD サービスをホストしている ITD 入力インターフェイスでルータ ACL を使用する場合、どちらの機能に対しても統計情報を有効にすることはできません。この制限の詳細については、『Cisco Nexus 9000 シリーズ NX-OS ユニキャストルーティング構成ガイド』の「ポリシーベースルーティング」の章にある「ポリシーベースルーティングのガイドラインと制限事項」の項を参照してください。

ACL の組み込みと除外

インクルード ACL

インクルード ACL 機能を使用すると、ITD サービスにアクセス制御リスト (ACL) を割り当てることができます。ACE に一致するトラフィックのみがノードに向かって負荷分散され、他のトラフィックはデフォルトのルーティング ルールに従います。

Cisco NX-OS リリース 9.3(3) 以降、1 つの ITD サービスで最大 8 つのアクセス リストを設定できます。各アクセス リストを独自のデバイス グループ (マルチ ACL) に関連付けることができます。特定のデバイス グループが 1 つのユーザー ACL に関連付けられている場合、そのデバイス グループが優先され、デフォルトのデバイス グループが上書きされます。この機能により、ITD はさまざまな ACL に一致するトラフィックをさまざまなデバイス グループにロードバランシングできます。

除外 ACL

除外 ACL を設定して、ITD が ITD ロードバランサから除外するトラフィックを指定できます。除外 ACL が選択するトラフィックは RIB ルーティングされ、ITD をバイパスします。除外 ACL は、送信元フィールドと接続先フィールドの両方に基づいてフィルタリングできます。除外 ACL は、仮想 IP アドレスの前にあります。

仮想 IP アドレスのフィルタリング

仮想 IP アドレスを使用して、ITD のトラフィックをフィルタリングできます。トラフィックフィルタリング用の仮想 IP アドレスとサブネット マスクの組み合わせは、宛先フィールドでのみサポートされます。

ポート番号ベースのフィルタリング

ポート番号付けを使用して、ITD のトラフィックをフィルタリングできます。レイヤ 4 ポート（たとえば、ポート 80）に基づいてトラフィックをフィルタリングするために、次の方法がサポートされています。

- 一致する宛先ポート

宛先ポートが 80 の任意の送信元または宛先 IP アドレスが一致します。（例：仮想 IP アドレスは `0.0.0.0 0.0.0.0 tcp 80` として構成されています。）

- 一致する送信元ポート

80 以外のポートは ITD をバイパスし、ポート 80 はリダイレクトされます。（例：除外 ACL は、`permit tcp any neq 80 any` として設定されます。）

- 複数のポート番号の一致

ITD では、ポートごとに 1 つずつ、複数の仮想 IP アドレス行を設定できます。

ホットスタンバイ

ホットスタンバイ機能は、スイッチを再構成して、動作可能なホットスタンバイ ノードを探し、最初に使用可能なホットスタンバイ ノードを選択して、障害が発生したノードを置き換えます。ITD は、障害が発生したノードを当初宛先としていたトラフィックセグメントを、ホットスタンバイ ノードにリダイレクトするようにスイッチを再設定します。このサービスは、ホットスタンバイ ノードとアクティブ ノードとの固定マッピングを強要しません。

障害が発生したノードが再び動作可能になると、アクティブ ノードとして復元されます。動作中のホットスタンバイ ノードからのトラフィックは元のノードにリダイレクトされ、ホットスタンバイ ノードはスタンバイ ノードのプールに戻ります。

複数のノードで障害が発生した場合、それらすべてのノードを宛先とするトラフィックは、最初に使用可能なホットスタンバイ ノードにリダイレクトされます。

ホットスタンバイ ノードは、ノード レベルでのみ構成できます。ノード レベルで、関連付けられたアクティブ ノードが失敗した場合にのみホットスタンバイ ノードはトラフィックを受信します。

ITD は $N + M$ 冗長性をサポートしており、 M ノードは N アクティブ ノードのホットスタンバイ ノードとして機能できます。

複数の入インターフェイス

複数の入インターフェイスに対してトラフィック リダイレクト ポリシーを適用するように ITD サービスを構成できます。この機能では、単一の ITD サービスを使用して、さまざまなインターフェイスに到着するトラフィックを一連のノードにリダイレクトできます。

Cisco NX-OS リリース 7.0(3)I7(3) 以降、同じ入インターフェイスを 2 つの ITD サービスに含めることができ、1 つの IPv4 ITD サービスと 1 つの IPv6 ITD サービスが可能になります。

IPv4 と IPv6 の両方の ITD サービスに同じ入インターフェイスを含めると、IPv4 と IPv6 の両方のトラフィックが同じ入インターフェイスに到着することができます。IPv4 トラフィックをリダイレクトするために IPv4 ITD ポリシーが適用され、IPv6 トラフィックをリダイレクトするために IPv6 ITD ポリシーが適用されます。



- (注) 同じ入インターフェイスが複数の IPv4 ITD サービスまたは複数の IPv6 ITD サービスで参照されていないことを確認してください。システムはそれを自動的に適用せず、サポートされていません。

システムヘルスマニタリング

ITD は、ノードとそれらのノードで実行されているアプリケーションの状態を定期的に監視して、障害を検出し、障害シナリオを処理します。

ICMP、TCP、UDP、DNS、および HTTP プロブがサポートされています。

ノードに接続されたインターフェイスの正常性

Cisco NX-OS リリース 7.0(3)I3(1) 以降、ITD ITD は IP サービスレベル アグリーメント (IP SLA) 機能を利用して、各ノードを定期的にプロブします。以前のリリースでは、ITD は Internet Control Message Protocol (ICMP) を使用して、各ノードを定期的にプロブします。プロブはデフォルトで 10 秒の頻度で送信され、1 秒まで設定できます。それらはすべてのノードに同時に送信されます。プール グループ構成の一部としてプロブを構成できます。

プロブは、デフォルトで 3 回再試行した後に障害が発生したと宣言されます。この時点で、ノードの状態は「機能不全」、ステータスは「PROBE_FAILED」になります。

ノード障害の処理

ノードがダウン状態としてマークされると、ITD はトラフィックの中断を最小限に抑えて、トラフィックを残りの運用可能なノードに再配布するために自動的に次のタスクを行います。

- 障害が発生したノードを引き継ぐようにスタンバイ ノードが構成されているかどうかを判別します。
- スタンバイ ノードが運用可能な場合、トラフィックを処理するノードの候補としてそのノードを識別します。

- 運用可能なスタンバイノードを使用できる場合、トラフィックを処理するアクティブノードとしてそのスタンバイノードを再定義します。
- 障害が発生したノードから新しくアクティブにされたスタンバイノードにトラフィックを再割り当てするように自動的にプログラムします。

プローブのユーザー定義トラック ID

ユーザーは独自のトラックを定義し、それらを各ノードに関連付けることができます。ノードにユーザー定義のトラックが割り当てられている場合、対応する **ip sla** 構成は、トラックを操作するユーザーによって構成される必要があります。ITD は、ノードに新しいトラックと **ip sla ID** を割り当てません。ユーザー定義のトラックは、プライマリ、スタンバイ、およびホットスタンバイノードに割り当てることができます。ユーザーは、ITD セッションによって追加された新しいノードにユーザー定義のトラックを割り当てることができます。ITD によって生成されたトラックは、ユーザー定義のトラックとして使用できません。

ユーザー定義のトラックを使用して新しいノードを追加する例：

```
itd device-group dg1
  node ip 1.1.1.2
    probe track 30
  node ip 1.1.1.3
    probe track 40
  node ip 1.1.1.4
    mode hot-standby
    probe track 50

itd device-group dg2
  node ip 1.1.1.6
    probe track 70
  standby ip 1.1.1.5
    probe track 60
```

ノードにユーザー定義のトラックがない場合、サービスが有効になったときに、ITD サービスは **track id** および **ip sla ID** を割り当てます。

ピア同期

ピア同期機能は、サンドイッチモードで2つの ITD ピア サービス間でノードのヘルスステータスを同期します。いずれかの ITD ピア サービスのリンクがダウンした場合のトラフィック損失を防ぐのに役立ちます。

各 ITD サービスは、ピアサービスを定期的にプローブして、障害を検出します。ping は毎秒 ITD ピア サービスに送信されます。応答が受信されない場合は、3回再試行されます。頻度と再試行回数は構成できません。



- (注) ピア サービス機能では、サービス間でノードの同期フェールオーバーを可能にするために、**fail-action** 最小バケットまたはバケットごとの **fail-action** ノードを構成する必要があります。さらに、いずれかのサービスがホットスタンバイノードまたはノードレベルのスタンバイを使用している場合、同期フェールオーバーはサポートされません。

Failaction 再割り当て

ITD の Failaction により、障害が発生したノードへのトラフィックを 1 つ以上の現用系ノードに再割り当てできます。障害が発生したノードが再び現用系になると、接続の処理が再開されます。すべてのノードがダウンした場合、パケットは自動的にルーティングされます。すべての Failaction メカニズムは、IPv4 サービスと IPv6 サービスの両方でサポートされます。



(注) Failaction 機能をイネーブルにする前に、ITD デバイス グループにプローブを設定する必要があります。

Failaction ノードの再割り当て

ノードがダウンすると、そのノードに関連付けられたトラフィック パケットは、構成されている一連のノードで最初に検出されたアクティブ ノードに再割り当てされます。新しく再割り当てされたノードでも障害が発生すると、トラフィックは次に使用可能なアクティブ ノードに再割り当てされます。

ノードが回復し、それ以上の障害イベントがない場合は、障害が発生する前にノードに最初に割り当てられていたトラフィック パケットがそのノードに再割り当てされます。

Failaction ノードの最小バケット (Failaction Node Least-Bucket)

ノードがダウンすると、そのノードに関連付けられたトラフィック パケットは、現在最小数のトラフィック パケットからトラフィックを受信している現用系ノードに再割り当てされます。後続のノード障害ごとに、トラフィック パケットが最も少ない現用系ノードが再計算され、障害が発生したノードに向けられたすべてのバケットがこのノードにリダイレクトされるため、再割り当てされたバケットを複数の現用系ノードに分散できます。

ノードが回復し、それ以上の障害イベントがない場合は、障害が発生する前にノードに最初に割り当てられていたトラフィック パケットがそのノードに再割り当てされます。

Failaction バケット分配 (Failaction Bucket Distribute)

サービスが有効な場合、ITD は内部アルゴリズムを使用して、プライマリ ノードのさまざまなシーケンスを、プライマリ ノードごとに異なる優先順位を持つ代替バックアップパスとして事前に選択します。ノードがダウンすると、そのノードへのトラフィックは、優先度が最も高い最初の現用系バックアップノードにリダイレクトされ、その後の障害についても同様にリダイレクトされ、それによってコンバージェンスの遅延が最小限に抑えられます。

ノードが回復すると、最初にプライマリとしてこのノードに割り当てられていたトラフィック パケットがそのノードに再割り当てされます。プライマリ ノードがまだ障害状態であり、新しく回復したノードが最も優先順位の高い現用系バックアップとして動作するトラフィック パケットも、そのトラフィック パケットに再割り当てされます。

のプライマリ ノード、またはデバイス グループの最大 32 のプライマリ ノード (いずれか少ない方) が、ノードごとに異なる優先順位で事前に選択されます。



- (注) このアルゴリズムは、比較的均等なトラフィック分散を目的としていますが、ノード障害が発生した場合の均等な分散を保証するものではありません。

Failaction 最適化

Cisco NX-OS リリース 9.2(2) より前では、ノードがダウンすると、そのノードに関連付けられたバケットは、fail-action アルゴリズムの決定に従ってアクティブ ノードに再割り当てされます。ただし、新しく再割り当てされたノードにも同時に障害が発生した場合、障害アクションの計算を再実行した後、元の障害ノードのトラフィック バケットを別のアクティブ ノードに再割り当てする必要があります。障害が発生したノード バケットをアクティブ ノードに再割り当てする際の遅延は、ネットワーク パフォーマンスに影響します。

fail-action の最適化では、ノードがダウンすると、利用可能なすべてのノードのステータスが最初に事前に取得されます。障害として検出されたすべてのノードの再割り当ては、失敗アクションメカニズムに基づいて実行されるため、再割り当ての繰り返しによる遅延が回避されます。

Cisco NX-OS リリース 9.3(3) 以降、この最適化は、ピア同期が設定されている場合を除き、すべてのサービスに対してデフォルトで有効になっています。

Failaction Node-Per-Bucket

特定のノードに障害が発生すると、バケットの数が最も少ないノードが識別され、バケットは、バケットの数が最も少ないノードから開始して、他のアクティブ ノードに分散されます。

ITD は、現在最も少ないバケット ノードを繰り返し識別し、すべてのバケットが再割り当てされるまで、そのノードに1つのバケットを割り当てます。したがって、すべてのバケットは、残りのすべてのアクティブ ノード間で均等に分散されます。



- (注) Cisco Nexus NX-OS リリース 9.3(5) 以降、ITD ITD は、ノードの重みに基づいて、フェールオーバーするノードを識別します。ノードに重みが設定されていない場合、デフォルトの重み1が使用されます。

Failaction 再割り当てを使用しない場合

Failaction によるノードの再割り当てを設定しない場合は、次の2つのシナリオが考えられます。

プローブを構成して Failaction 再割り当てをしない

ITD プローブでは、ノードの障害やサービス到達可能性の消失を検出できます。ノードに障害が発生した場合、failaction が設定されていないため、トラフィックはルーティングされ、再割

り当てされません。ノードが回復すると、その回復したノードがトラフィックの処理を開始します。

プローブの構成なしで **Failaction** 再割り当てをしない

プローブが構成されていないと、ITDはノードの障害を検出できません。ノードがダウンしても、ITDはアクティブノードへのトラフィックの再割り当てまたはリダイレクトを行いません。

ライセンス要件

Cisco NX-OS ライセンス方式の推奨の詳細と、ライセンスの取得および適用の方法については、『[Cisco NX-OS ライセンス ガイド](#)』および『[Cisco NX-OS ライセンス オプションガイド](#)』を参照してください。

Guidelines and Limitations for ITD

ITD has the following configuration guidelines and limitations:

- ITD is supported on the following platforms:

ITDv4 support

- Beginning with Cisco Nexus NX-OS Release 9.3(1), Cisco Nexus 9500 Series switches with Cisco Nexus X9788TC-FX , X97160YC-EX and X9732C-EX line cards.
- Beginning with Cisco Nexus NX-OS Release 9.2 (1), Cisco Nexus C9364C, C9336C-FX2, C93240YC-FX2 switches are supported.
- Cisco Nexus 9500 Series switches with Cisco Nexus X9432PQ, X9464PX, X9464TX, X9536PQ, X9564PX, X9564TX, and X9636PQ line cards.
- Cisco Nexus 9332PQ, 9372PX, 9372PX-E, 9372TX, 9372TX-E, 9396PX, 9396TX, 93120TX, and 93128TX switches.
- Cisco Nexus 9236C, 92160YC-X, and 92304QC switches, 93180YC-EX, 93108TC-EX, C93180YC-FX and C93108TC-FX switches

ITDv6 support

- Cisco Nexus 9236C, 92160YC-X, and 92304QC switches, 93180YC-EX, 93108TC-EX, C93180YC-FX, and C93108TC-FX switches
- Beginning with Cisco Nexus NX-OS Release 9.2 (1), Cisco Nexus C9364C, C9336C-FX2, C93240YC-FX2 switches are supported.
- Beginning with Cisco Nexus NX-OS Release 9.3 (5), Cisco Nexus 9500 Series switches with Cisco Nexus X9732C-FX and X97160YC-EX line cards and Sup B+ are supported.
- Beginning with Cisco Nexus NX-OS Release 9.3 (5), Cisco Nexus C9316D-GX, C93600CD-GX, C9364C-GX and C93180YC-FX3S switches are supported.

ITD NAT, and PAT

- Beginning with Cisco Nexus NX-OS Release 9.3(5) 9348GC-FXP switches are supported.
- ITD does not support using FEX ports for ingress or egress to the next-hop IP address.
- Configuration rollback and configuration replace are supported only when the ITD service is in shut mode on both the target and source configurations.
- Destination NAT is supported only for IPv4.
- Seamless switchover is supported on L3 ITD services.
- SNMP is not supported for ITD.
- Beginning with Cisco Nexus NX-OS Release 9.3(2), IPv6 supports node level probes and device group level probes.
- Node level IPv6 TCP, ICMP probes are supported.
- Beginning with Cisco Nexus NX-OS Release 9.3(5), ITD supports **fail-action node-per-bucket** with weights.
- The **bucket distribution** options is available for IPv4 and IPv6.



Note Fail-action bucket distribute is not recommended for services using hot-standby nodes or node-level standby nodes.

- The following guidelines and limitations apply to the exclude ACL feature:
 - The exclude ACL supports only permit access control entries (ACEs). Deny ACEs are not supported.
 - Traffic that is matched by a permit ACE in an exclude ACL bypasses ITD.
- **The following guidelines and limitations apply to the include ACL feature:**
 - Only 62 unique ACLs can be configured per slice of ASIC. Each ACL takes one label. If the same ACL is configured on multiple interfaces, the same label is shared. If each ACL has unique entries, the ACL labels are not shared, and the label limit is 62. In order to achieve 150 ITD services per switch, with the limitation of 62 ACLs per slice, the ingress interfaces should be spread across multiple slices of the ASIC. For more information see, [Configuring IP ACLs](#).
 - ACEs with object-groups specified as address-groups or port-groups in either source or destination parameters are not supported.
 - An IPv6 ACL can be configured as an include access-list for traffic selection for ITD service.
 - Ingress ACL doesn't have support for Layer 4 port range in the user-defined ACL.
 - Only ACEs with the **permit** method are supported in the ACL. ACEs with any other method (such as **deny** or **remark**) are ignored.
 - A maximum of 256 permit ACEs are supported in one ACL.
 - Failaction is supported among the nodes.

- ITD supports either the include ACL feature or the virtual IP address (VIP) feature but not both.
- If the user has configured ITD with include or exclude ACL, and the user is using source IP-based load balancing, then the subnet mask in the source IP address of the ACE cannot be more than /28 (eg, it cannot be /29, /30, /31).
If the user has configured ITD with include or exclude ACL, and the user is using destination IP-based load balancing, then the subnet mask in the destination IP address of the ACE cannot be more than /28 (eg, it cannot be /29, /30, /31).
- Beginning with Cisco Nexus NX-OS Release 9.3(5), mask position is supported for services filtering traffic using include ACLs.
- Beginning with Cisco Nexus NX-OS Release 9.3(5), Least-bit load-balancing is supported for the include ACL feature.
- We recommend that you classify probe traffic in a separate CoPP class. Otherwise, probe traffic will go in the default CoPP class by default and might be dropped, causing IP SLA bouncing for probe traffic. For configuration information, see [Configuring CoPP for IP SLA Packets](#)
- ITD sessions are not supported with the following:
 - Weights.
 - The include and exclude ACL features.
 - Fail-action node-per-bucket.
 - Node level probes.



Note Node level probes which use user-defined track are supported.

- Device-groups with hot-standby or node level standby nodes.
- Device-groups being used by services with peer synchronization enabled.
- Services with layer-4 load-balance options configured.
- Services with multiple Virtual IPs using different device-groups.
- Disabling atomic update may allow more TCAM resources to be made available for the ITD policies, but with possible disruption in traffic during changes to policies. For further details, please refer to [Security Configuration Guide 9.3\(x\)](#).
- ITD-L2 & ITD Layer 3 need to have separate interfaces.
- Check point & config rollback functionality in ITD is supported only when service is down
- If ITD services are being used with either IPv4 or IPv6 Virtual IP addresses, with or without NAT destination enabled, when not all the Virtual IP addresses are configured to use the same device-group, enable load-balance option with fixed number of buckets.
- It is recommended to de-activate the ITD services prior to an ASCII reload. In the scenario where user-defined tracks are not used, keeping the services active during the reload leads to stale tracks and IPSLAs in the system. If ASCII reload proceeds with the services in active state, you can identify

the tracks and SLAs in use by ITD via the **show itd** command and delete the stale or unused tracks and SLAs manually, as required.

• **The following guidelines and limitations apply to the Destination NAT feature:**

- NAT is supported with VIP and/or Protocol/Port. It is not supported without VIP.
- If using same set of servers to load balance, the Virtual IP (VIP) should have a unique L4 Port number.
- If the port number is same across multiple services, NAT cannot reuse the same device-groups and nodes.
- Limit of maximum 1024 NAT entries.
- NAT IPv6 is not supported, only IPv4 is supported.
- Only failactions supported are **least-bucket** and **node per bucket**



Note ITD Nat is not supported with **fail-action node reassign** and **bucket distribute** .

- ITD NAT is supported only on Nexus 9300.
- ITD Peersync is not supported with ITD NAT.
- ITD sessions are not supported NAT.
- Hot-Standby, device group and node level standby not supported with ITD NAT.
- Advertise enable option is mandatory for every VIP, in a service with ITD NAT enabled.
- VRF is not supported.
- NAT is not supported with ITD over vxlan .
- NAT is not supported with DST based load balancing.
- ITD NAT is not supported with exclude ACL and Include ACL.
- If atomic updates are enabled, the number of TCAM entries should be less than the TCAM carving.
- ITD sessions and non-disruptive addition or deletion of nodes is not supported.
- Seamless switchover is not supported on ITD NAT.



Note Prior to performing ISSD from Cisco NX-OS, Release 9.3(1) to a previous release, remove nat destination configuration from the service and proceed with downgrade.

• **The following limitations apply to the ITD over VXLAN feature and are not supported:**

- Fail action methods.

- Probes.
 - ITD sessions.
 - IPv6 nodes in device group.
 - VPC.
 - Peer synchronization.
 - Node-level standby.
- Legacy ITD & ITD over VXLAN service cannot share the same device group on a node.
 - Prior to using ISSU from previous release, **feature PLB** needs to be deactivated.
 - VIP and Hot-standby are mandatory configurations to enable ITD over VXLAN.
 - Irrespective of the configuration application method (using CLI or DME), the nodes in device-group order needs to be same across all leaf nodes.

ITD PAT has the following configuration guidelines and limitations:

- When using multiple vip in device group with PAT, we need to associate unique device-group per VIP.
- Port number is mandatory along with VIP when using PAT

ITD-L2 load balancing has the following configuration guidelines and limitations:

- Cisco Nexus 9372PX, 93108TC-EX, and the Cisco Nexus 9516 switches support Layer-2 load balancing servicing. Beginning with Cisco Nexus NX-OS Release 9.3(5), N9K-C93180YC-FX and N9K-C93240YC-FX2 are supported.



Note Layer-2 load balancing feature is not supported on Cisco 9500 EX / FX / R line cards.

Layer-2 load balancing does not support the vPC, port channel, and the L3 interfaces.

- Only the port group interfaces in a trunk are supported.
- You must not share the ITD-L2 port-group to more than two service.
- Ensure that the TCAM size is equal to the sum of the number of buckets in addition to the number of services.



Note For N9K-X9464PX, ensure that the TCAM size is equal to the sum of the number of the configured VLANs on the service by the number of buckets in addition to the number of services. .

- ITD allows configuration of 150 services. However, for ITD-L2, user cannot configure more than 4 services.

- Prior to using ISSU from previous release, **feature smart-channel** needs to be deactivated.
- include and exclude acl in not supported
- L4 port based load balancing is supported.
- The following limitations apply to the ITD-L2 feature and are not supported:
 - Fail action methods.
 - Probes.
 - ITD sessions.
 - IPv6 nodes in device group.
 - VPC.
 - Peer synchronization.
 - Node-level standby.

ITD サポート サマリー

ITD サポート レベルのリストについては、次の表を参照してください。

表 1: ITD サポート レベル

| 機能 | ITDv4 | ITDv6 | 説明 |
|---------------|---|---|----|
| デバイス グループ レベル | <ul style="list-style-type: none"> • TCP • ICMP • HTTP • UDP • DNS | <ul style="list-style-type: none"> • TCPv6 • ICMPv3 | |
| ノードごとのプローブレベル | はい | はい | |
| Hot-Standby | はい | はい | |
| 重量 | はい | はい | |
| 中断のない運用 | | | |
| ACL リフレッシュ | はい | はい | |
| プライマリ ノード | はい | はい | |

| 機能 | ITDv4 | ITDv6 | 説明 |
|-----------------|---|---|--------------------------|
| ホットスタンバイ ノード | いいえ | いいえ | |
| サービス レベル | | | |
| インクルード ACL | はい | はい | |
| Failaction メソッド | <ul style="list-style-type: none">• reassign• least-bucket• node-per-bucket• bucket distribute | <ul style="list-style-type: none">• reassign• least-bucket• node-per-bucket• bucket distribute | |
| 除外-ACL | はい | はい | 拒否 ACE はサポート されていません。 |

| 機能 | ITDv4 | ITDv6 | 説明 |
|---------------------|---|--|----|
| サポートされるプラットフォームフォーム | <p>EX/FX ラインカードを搭載した Cisco Nexus 9500 スイッチ： X9788TC-FX、 X97160YC-EX および X9732C-EX</p> <p>Cisco Nexus 9236C、 92160YC-X、92304QC スイッチ および 9300-EX Series スイッチ</p> <p>Cisco Nexus 9332PQ、 9372PX、9372PX-E、 9372TX、9372TX-E、 9396PX、9396TX、 93120TX と 93128TX スイッチ。</p> <p>Trident 2 ラインカード を搭載した Cisco Nexus シリーズ スイッチ： N9K-X9564PX、 N9K-X9536PQ</p> <p>Cisco Nexus X9432PQ、X9464PX、 X9464TX、X9536PQ、 X9564PX、X9564TX、 および X9636PQ ライン カード。</p> | <p>Cisco Nexus 93180YC-EX、 93108TC-EX、 C93180YC-FX および C93108TC-FX スイッチ。</p> <p>Cisco Nexus C9364C、 C9336C-FX2、 C93240YC-FX2 スイッチ。</p> | |
| 宛先 NAT | <p>Cisco Nexus 93180YC-EX、 93108TC-EX、 C93180YC-FX、 C93180YC-FX3S、 C93108TC-FX3P、 93108TC-FX、 93240YC-FX2、 C9336C-FX2 と 9300-GX プラット フォームスイッチが、 サポートされています。</p> | いいえ | |

| 機能 | ITDv4 | ITDv6 | 説明 |
|----------------|-------|-------|----|
| ITD over VXLAN | はい | 非対応 | |

ITD のデフォルト設定

次の表に、ITD パラメータのデフォルト設定を示します。

表 2: デフォルトの ITD パラメータ

| パラメータ | デフォルト |
|------------------|-------|
| プローブの頻度 | 10 秒 |
| プローブの再試行ダウン カウント | 3 |
| プローブの再試行アップ カウント | 3 |
| プローブ タイムアウト | 5 秒 |

ITD の構成

ITD のイネーブル化

ITD コマンドにアクセスする前に、ITD 機能を有効にする必要があります。

始める前に

ネットワーク サービス ライセンスがインストールされていることを確認してください。

ポリシーベース ルーティング (PBR) が有効になっていることを確認します。

手順の概要

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

手順の詳細

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

| | コマンドまたはアクション | 目的 |
|--------|--|---|
| | switch# configure terminal switch(config)# | |
| ステップ 2 | [no] feature itd 例： switch(config)# feature itd | ITD 機能をイネーブルにします。デフォルトでは、ITD は無効になっています。 |
| ステップ 3 | (任意) copy running-config startup-config 例： switch(config)# copy running-config startup-config | 実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。 |

デバイス グループの構成

ITD デバイス グループを作成してから、グループのノードとプローブを指定できます。Cisco NX-OS リリース 7.0(3)I3(1) 以降では、複数のデバイス グループを構成できます。

始める前に

ITD 機能がイネーブルであることを確認します。

デバイスが Cisco NX-OS リリース 7.0(3)I3(1) 以降を実行している場合は、次のコマンドが設定されていることを確認します。 **feature sla senderfeature sla responder**

手順の概要

1. **configure terminal**
2. **[no] itd device-group name**
3. **[no] node {ip | ipv6} {ipv4-address | ipv6-address}**
4. **[no] probe track id**
5. **[no] weight weight**
6. **[no] port port value**
7. **[no] mode hot-standby**
8. **[no] shutdown**
9. **exit**
10. ノードごとに手順 3 ~ 5 を繰り返します。
11. **[no] probe {icmp | http | tcp port port-number | udp port port-number | dns [frequency seconds] [[retry-down-count | retry-up-count] number] [timeout seconds]}**
12. **[no] hold-down threshold count <count> [time <time>]**
13. (任意) **copy running-config startup-config**

手順の詳細

| | コマンドまたはアクション | 目的 |
|--------|---|--|
| ステップ 1 | configure terminal 例： switch# configure terminal switch(config)# | グローバル コンフィギュレーション モードを開始します。 |
| ステップ 2 | [no] itd device-group name 例： switch(config)# itd device-group dg1 switch(config-device-group)# | ITD デバイス グループを作成し、デバイス グループ コンフィギュレーション モードを開始します。最大 32 文字の英数字を入力できます。 |
| ステップ 3 | [no] node {ip ipv6} {ipv4-address ipv6-address} 例： switch(config-device-group)# node ip 20.20.20.3 switch(config-dg-node)# 例： switch(config-device-group)# node ipv6 2001::198:1:1:11 switch(config-dg-node)# | ITD のノードを指定します。 |
| ステップ 4 | [no] probe track id 例： switch (config-device-group)# probe track 30 switch(config-device-group-node)# | プローブのユーザー定義トラック ID を構成します。 |
| ステップ 5 | [no] weight weight 例： switch(config-dg-node)# weight 6 | ITD のノードの重みを指定します。有効な範囲は 1 ~ 256 です。 |
| ステップ 6 | [no] port port value 例： switch(config-dg-node)# node ip 10.10.10.10 port 1000 | 機能ポート アドレス変換のポート番号を指定します。値の範囲は 1 ~ 65535 です。 |
| ステップ 7 | [no] mode hot-standby 例： switch (config-device-group)# node ipv6 50::1 switch(config-device-group-node)# mode hot-standby | ノードをデバイス グループのホット スタンバイ ノードとして構成します。 |
| ステップ 8 | [no] shutdown 例： switch(config-dg-node)# node ip 2.1.1.1 switch(config-dg-node)# shutdown | ノードをメンテナンス モードに移動または終了します。 |

| | コマンドまたはアクション | 目的 |
|---------|---|--|
| | <pre>switch(config-dg-node)# no shutdown switch(config-dg-node)#</pre> | |
| ステップ 9 | <p>exit</p> <p>例 :</p> <pre>switch(config-dg-node)# exit switch(config-device-group)#</pre> | デバイス グループ ノード 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 | ノードごとに手順 3 ~ 5 を繰り返します。 | |
| ステップ 11 | <p>[no] probe {icmp http tcp port port-number udp port port-number dns [frequency seconds] [[retry-down-count retry-up-count] number] [timeout seconds]}</p> <p>例 :</p> <pre>switch(config-device-group)# probe icmp frequency 100</pre> | <p>クラスタ グループのサービス プローブを構成します。</p> <p>Cisco NX-OS リリース 7.0(3)I3(1) 以降、ITD サービスのプローブとして次のプロトコルを指定できます。</p> <ul style="list-style-type: none"> • ICMP • [TCP] • [UDP] • HTTP • DNS <p>以前のリリースでは、ITD サービスのプローブとして ICMP が使用されていました。</p> <p>オプションは次のとおりです。</p> <ul style="list-style-type: none"> • frequency : プローブの頻度を秒単位で指定します。値の範囲は 1 ~ 604800 です。 • retry-down-count : ノードがダウンしたときにプローブによって実行される再カウントの数を指定します。指定できる範囲は 1 ~ 5 です。 • retry-up-count : ノードが復帰したときにプローブが実行する再カウントの数を指定します。指定できる範囲は 1 ~ 5 です。 • timeout : タイムアウト期間を秒単位で指定します。値の範囲は 1 ~ 604800 です。 |
| ステップ 12 | <p>[no] hold-down threshold count <count> [time <time>]</p> <p>例 :</p> <pre>switch(config-itd)# itd device-group dg switch(config-device-group)# hold-down threshold count 1 switch(config-device-group)# node ip 1.1.1.1</pre> | ノードまたはデバイス グループの保留しきい値障害カウントとしきい値タイマーを指定します。 |

| | コマンドまたはアクション | 目的 |
|---------|--|---|
| | <code>switch(config-dg-node)# hold-down threshold count 3 time 200</code> | |
| ステップ 13 | (任意) copy running-config startup-config 例： <code>switch(config-device-group)# copy running-config startup-config</code> | 実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。 |

ITD サービスの構成

始める前に

ITD 機能がイネーブルであることを確認します。

ITD サービスに追加されるデバイス グループが構成されたことを確認します。

手順の概要

1. **configure terminal**
2. `[no] itd service-name`
3. `[no] device-group device-group-name`
4. `[no] ingress interface interface`
5. `[no] load-balance {method {src {ip | ip-l4port [tcp | udp] range x y} | dst {ip | ip-l4port [tcp | udp] range x y}} | buckets bucket-number | mask-position mask-position | least-bit}`
6. `[no] virtual [ip | ipv6] { ipv4-address ipv4-network-mask | ipv6-address ipv6-network-mask } [{ proto {port_num | port_any}}] [{advertise} {enable | disable}] [device-group dgrp_name]`
7. 次のいずれかのコマンドを入力して、ノード障害後にトラフィックを再割り当てする方法を決定します。
 - `[no] failaction node reassign`
 - `[no] failaction node least-bucket`
 - `[no] failaction bucket distribute`
 - `[no] failaction node per-bucket`
8. `[no] vrf vrf-name`
9. `[no] exclude access-list acl-name`
10. (任意) `[no] peer local service peer-service-name`
11. **no shutdown**
12. (任意) `show itd [itd-name]`
13. (任意) **copy running-config startup-config**

手順の詳細

| | コマンドまたはアクション | 目的 |
|--------|--|---|
| ステップ 1 | configure terminal 例 : <pre>switch# configure terminal switch(config)#</pre> | 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] itd service-name 例 : <pre>switch(config)# itd service1 switch(config-itd)#</pre> | ITD サービスを設定し、ITD 構成モードを開始します。最大 32 文字の英数字を入力できます。 |
| ステップ 3 | [no] device-group device-group-name 例 : <pre>switch(config-itd)# device-group dg1</pre> | ITD サービスに既存のデバイス グループを追加します。 <i>device-group-name</i> は、デバイス グループの名前を指定します。最大 32 文字の英数字を入力できます。 (注) Cisco NX-OS リリース 7.0(3)I3(1) 以降では、複数のデバイス グループを ITD サービスに追加できます。 |
| ステップ 4 | [no] ingress interface interface 例 : <pre>switch(config-itd)# ingress interface ethernet 4/1-10</pre> | ITD サービスに1つ以上のインターフェイスを追加します。 複数のインターフェイスは、カンマを（「,」）を使用して区切ります。インターフェイスの範囲は、ハイフン（「-」）を使用して指定します。 インターフェイスをサービスに関連付ける前に、必要な VRF およびインターフェイス モードを設定します。 |
| ステップ 5 | [no] load-balance {method {src {ip ip-l4port [tcp udp] range x y} dst {ip ip-l4port [tcp udp] range x y}} buckets bucket-number mask-position mask-position least-bit} 例 : <pre>switch(config-itd)# load-balance method src ip buckets 16</pre> | ITD サービスのロード バランシング オプションを設定します。 オプションは次のとおりです。 <ul style="list-style-type: none"> • method : 送信先または接続先の IP アドレスベースの負荷またはトラフィック分散を指定します。 • buckets : 作成するバケットの数を指定します。1つ以上のバケットが1つのノードにマップされています。バケットは2のべき乗数で設定する必要があります。範囲は2～256です。 |

| | コマンドまたはアクション | 目的 |
|--------|---|--|
| | | <p>(注) 設定するバケットの数がノードの数より多い場合、バケットはすべてのノードにラウンドロビン方式で適用されます。</p> <ul style="list-style-type: none"> • mask-position : ロードバランスのマスク位置を指定します。 • least-bit : 最小ビットのロードバランススキームを可能にします。このスキームにより、バケット生成メカニズムが連続する少数のクライアント IP プレフィクスを同じバケットに分配できるようにします。 • include-acl を使用するサービスの場合、最小ビット (マスク位置の有無にかかわらず) を使用して、同じバケットに分散する連続する IP ホストを減らします。 <p>(注) マスク位置がバケット数と負荷分散モードに基づいて使用可能なビットを超えると、バケットの生成中に内部的にデフォルトで 0 になります。</p> |
| ステップ 6 | <p>[no] virtual [ip ipv6] { ipv4-address ipv4-network-mask ipv6-address ipv6-network-mask } [{ proto { port_num port_any } }] [{ advertise } { enable disable }] [device-group dgrp_name]</p> <p>例 :</p> <pre>switch(config-itd)# virtual ip 100.100.100.100 255.255.255.255 udp 443 advertise enable active</pre> <p>例 :</p> <pre>switch(config-itd)# virtual ipv6 100::100 128 udp 443</pre> | <p>ITD サービスの仮想 IPv4 または IPv6 アドレスを設定します。</p> <p>proto オプション (TCP または UDP) は、仮想 IP アドレスが指定されたプロトコルからのフローを受け入れることを指定します。ポート範囲は 0 ~ 65535 です。</p> <p>[advertise {enable disable}] オプションは、仮想 IP ルートを隣接デバイスにアドバタイズするかどうかを指定します。VIP アドバタイズオプションが有効になっている場合、1つ以上のプライマリノードまたはホットスタンバイノードが仮想 IP またはサービスの下のデフォルトのデバイスグループに関連付けられたデバイスグループでアクティブになっている場合、ITD はルートを仮想 IP アドレスにアドバタイズします (該当する場合)。VIP アドバタイズオプションを有効にするには、すべてのプライマリノードとホットスタンバイノードを、デバイスグループまたはノードレベルでプローブを介して追跡できる必要があります。</p> |

| | コマンドまたはアクション | 目的 |
|----------------------|--|--|
| | | <p>(注) Cisco NX-OS リリース 9.3(2) 以降、advertise {enable disable} [active] オプションは Warning (注意) を発行して [advertise {enable disable}] オプションを使用します。</p> <p>(注) Cisco NX-OS リリース 9.3(3) 以降、IPv6 ITD では、advertise enable および advertise enable active オプションがサポートされています。</p> <p>仮想 IP の複数のインスタンスは、同じ IP アドレスを持つサービスの下で構成できますが、ネットマスク (またはプレフィックス長)、プロトコル、またはポートが異なります。ユーザーは、トラフィックフローが意図したとおりに負荷分散できるように、仮想 IP、マスクプロトコル、およびポートの一致が一意であることを確認する必要があります。</p> |
| <p>ステップ 7</p> | <p>次のいずれかのコマンドを入力して、ノード障害後にトラフィックを再割り当てする方法を決定します。</p> <ul style="list-style-type: none"> • [no] failaction node reassign • [no] failaction node least-bucket • [no] failaction bucket distribute • [no] failaction node per-bucket <p>例 :</p> <pre>switch(config-itd)# failaction node reassign</pre> <p>例 :</p> <pre>switch(config-itd)# failaction node least-bucket</pre> <p>例 :</p> <pre>switch(config-itd)# failaction bucket distribute</pre> | <p>サービスが使用する fail-action メカニズムを構成します。</p> <p>(注) このアルゴリズムは、比較的均等なトラフィック分散を目的としていますが、均等な分散を保証するものではありません。</p> <p>(注) failaction bucket distribute コマンドは、IPv4 と IPv6 の両方でサポートされています。</p> |
| <p>ステップ 8</p> | <p>[no] vrf vrf-name</p> <p>例 :</p> <pre>switch(config-itd)# vrf RED</pre> | <p>ITD サービスの VRF を指定します。</p> |

| | コマンドまたはアクション | 目的 |
|---------|--|--|
| ステップ 9 | [no] exclude access-list <i>acl-name</i> 例： switch(config-itd)# exclude access-list acl1 | ITD が ITD ロードバランサから除外するトラフィックを指定します。 |
| ステップ 10 | (任意) [no] peer local service <i>peer-service-name</i> 例： switch(config-itd)# peer local service service-A | 同じ (ローカル) スイッチ上にあるサンドイッチモードの 2 つの ITD ピア サービスの 1 つを指定します。別の ITD サービスを作成し、このコマンドを使用して 2 番目の ITD ピア サービスを指定する必要があります。両方のサービスでこのコマンドを実行すると、ノードのヘルス ステータスが 2 つのサービス間で同期されます。 (注) 2 つのデバイス グループのノードは、同じ順序である必要があります。具体的には、順序が保持されるように、両方のデバイス グループの最初のエントリは同じサンドイッチモード用である必要があります。 |
| ステップ 11 | no shutdown 例： switch(config-itd)# no shutdown | ITD サービスをイネーブルにします。 |
| ステップ 12 | (任意) show itd [<i>itd-name</i>] 例： switch(config-itd)# show itd | 特定の ITD インスタンスのステータスおよび構成を表示します。 |
| ステップ 13 | (任意) copy running-config startup-config 例： switch(config-itd)# copy running-config startup-config | 実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。 |

接続先 NAT および PAT の構成

始める前に

機能 ITD と機能 NAT を有効にします。

手順

| | コマンドまたはアクション | 目的 |
|--------|--|---|
| ステップ 1 | configure terminal 例： switch# configure terminal | 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 nat {outside inside } 例： switch (config) # interface Ethernet1/9 ip address 9.9.9.1/24 ip nat outside no shutdown interface Ethernet1/10 ip address 10.10.10.1/24 ip nat inside no shutdown | これは、ITD NAT が機能するために必須です。 負荷分散のためにトラフィックが入力であるインターフェイスで ip nat {outside} を構成します。 サーバーが接続されているインターフェイスに ip nat {inside} を構成します。 |
| ステップ 3 | itd service-name 例： switch (config) # itd nat1 | ITD サービスを設定し、ITD 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 | device-group device-group-name 例： switch(config-itd)# device-group dg1 | ITD サービスに既存のデバイス グループを追加します。 device-group-name は、デバイス グループの名前を指定します。最大 32 文字の英数字を入力できます。 |
| ステップ 5 | virtual ip ipv4-address ipv4-network-mask 80 advertise enable 例： virtual ip 6.6.6.1 255.255.255.255 tcp 80 advertise enable device-group dg1 | ITD サービスの TCP ポートで仮想 IPv4 アドレスを設定します。 (注) アドバタイズ可能オプションは、ITD NAT 可能なサービスでのすべての VIP にとって必須です。 (注) ポートアドレス変換が有効になっている場合、VIP ではポート番号が必須です。 |
| ステップ 6 | nat destination 例： switch(config-itd)# nat destination | 接続先 NAT を設定します。 |
| ステップ 7 | ingress interface interface 例： switch(config-itd)# ingress interface Eth1/1 | 入力インターフェイスまたは複数のインターフェイスを ITD サービスに追加し、設定する入力インターフェイスに直接接続されているインターフェイスの IP アドレスであるネクストホップ 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 |
|---------|---|--|
| ステップ 8 | <p>[no] failaction <i>failaction</i> メソッド</p> <p>例 :</p> <pre>switch(config-itd)# failaction node per-bucket</pre> | <p>failaction メソッドを割り当てます。</p> |
| ステップ 9 | <p>[no] load-balance {method {src {ip ip-l4port [tcp udp] range <i>x y</i>} dst {ip ip-l4port [tcp udp] range <i>x y</i>} } buckets <i>bucket-number</i> mask-position <i>mask-position</i> least-bit}</p> <p>例 :</p> <pre>switch(config-itd)# load-balance method src ip buckets 64</pre> | <p>ITD サービスのロード バランシング オプションを設定します。</p> <p>オプションは次のとおりです。</p> <ul style="list-style-type: none"> • method : 送信先または接続先の IP アドレスベースの負荷またはトラフィック分散を指定します。 • buckets : 作成するバケットの数を指定します。1つ以上のバケットが1つのノードにマップされています。バケットは2のべき乗数で設定する必要があります。範囲は2～256です。 <ul style="list-style-type: none"> (注) 設定するバケットの数がノードの数より多い場合、バケットはすべてのノードにラウンドロビン方式で適用されます。 • mask-position : ロードバランスのマスク位置を指定します。 • least-bit : 最小ビットのロードバランススキームを可能にします。このスキームにより、バケット生成メカニズムが連続する少数のクライアント IP プレフィクスを同じバケットに分配できるようにします。 <ul style="list-style-type: none"> (注) マスク位置がロードバランス モードで、バケット数に基づく利用可能なビット数を超える場合、バケットを生成する間、内部のデフォルトである 0 に設定されます。 |
| ステップ 10 | <p>no shutdown</p> <p>例 :</p> <pre>switch(config-itd)# no shutdown</pre> | <p>ITD サービスをイネーブルにします。</p> |

VXLAN を介した ITD の構成

始める前に

機能 pbr および機能 sla 送信者 CLI は、VXLAN 経由で ITD を構成するための前提条件です。

ITD 機能がイネーブルであることを確認します。

ITD サービスに追加されるデバイス グループが構成されたことを確認します。

手順の概要

1. **configure terminal**
2. **itd device-group dg-name**
3. **no node ip ip-address**
4. **[no] mode hot-standby ip**
5. **[no] virtual ip ipv4-address ipv4-network-mask [tcp | udp {port-number | any}] [device-group dg-name]**
6. **[no] source-interface loopback loopback-id**
7. **[no] ingress interface interface**

手順の詳細

| | コマンドまたはアクション | 目的 |
|--------|--|--|
| ステップ 1 | configure terminal 例 : <pre>switch# configure terminal switch(config)#</pre> | 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 | itd device-group dg-name 例 : <pre>switch(config)# itd device-group dg200 switch(config-device-group)# switch# show running-config itd services</pre> | ITD デバイス グループを構成します。すべてのノード（サーバー）は、デバイス グループ サブモードで構成されます。 no itd device-group CLI は、構成から ITD デバイス グループを削除します。 |
| ステップ 3 | no node ip ip-address 例 : <pre>switch(config)# itd device-group dg200 switch(config-device-group)# node ip 10.0.0.31 leaf3(config-dg-node)# node ip 10.0.0.31</pre> | デバイス グループ サブモードでノード（サーバー クラスタ）を構成します。デバイス グループ内には最大 32 のノードを構成できます。 デバイス グループの下のノードまたはサーバー構成を削除するには、CLI の no 形式を使用します。 |
| ステップ 4 | [no] mode hot-standby ip 例 : <pre>switch (config-device-group)# node ipv4 50::1 switch(config-device-group-node)# mode hot-standby</pre> | ノードをデバイス グループのホットスタンバイ ノードとして構成します。 デバイス グループの下のノードまたはサーバー構成を削除するには、CLI の no 形式を使用します。 |

| | コマンドまたはアクション | 目的 |
|--------|--|--|
| | | no mode hot-standby ip CLI を使用して、アクティブノードのホットスタンバイノード構成を削除します。 |
| ステップ 5 | <p>[no] virtual ip <i>ipv4-address ipv4-network-mask</i> [tcp udp {port-number any}] [device-group <i>dg-name</i>]</p> <p>例 :</p> <pre>switch (config-device-group)# virtual ip 6.6.6.1 255.255.255.0 device-group sf</pre> | <p>ノードのクラスタのサービスで仮想 IP アドレスを構成します。</p> <p>同じサービス内の異なるデバイスグループで同じ VIP を使用することはできません。</p> <p>最大 64 の VIP をサービス内に構成できます。</p> <p>デバイスグループを VIP に関連付けることもできます。このオプションを使用すると、複数のデバイスグループを 1 つのサービスの一部にすることができます。</p> <p>サービスで構成されているすべての VIP でデバイスグループが指定されている場合、デフォルトのデバイスグループ構成は必要ありません。</p> |
| ステップ 6 | <p>[no] source-interface loopback <i>loopback-id</i></p> <p>例 :</p> <pre>switch (config-device-group)# source-interface loopback9</pre> | <p>サービスの source-interface を構成します。</p> <p>(注) サービスに割り当てられるループバックインターフェイスは、ITD vrf (テナント VRF) と同じ vrf で最初に作成する必要があり、サービスの送信元インターフェイスとしてこのループバックを使用する前に、/32 マスクを持つ IP アドレスを割り当てる必要があります。</p> |
| ステップ 7 | <p>[no] ingress interface <i>interface</i></p> <p>例 :</p> <pre>switch (config-device-group)# source-interface loopback9</pre> | <p>サービスの入力インターフェイスの構成</p> <p>入力インターフェイスは、ボーダーリーフの dot1q カプセル化を使用したエニーキャストゲートウェイ SVI またはサブインターフェイスです。</p> <p>入力インターフェイスは、サービスと同じ VRF にある必要があります。</p> |

例

次に、VXLAN を介した ITD の構成例を示します。

```
itd device-group sf
node ip 10.200.1.2
node ip 10.200.6.2
node ip 10.200.7.2
node ip 10.200.2.2
```

```

mode hot-standby
node ip 10.200.3.2
node ip 10.200.4.2
node ip 10.200.5.2
node ip 10.200.9.2
mode hot-standby
node ip 10.200.12.2
itd ser1
source-interface loopback9
virtual ip 6.6.6.1 255.255.255.0 device-group sf
ingress interface Vlan100
ingress interface Eth1/1
load-balance method src ip buckets 256

```

ポートグループの構成

ITD-L2を設定するには、ポートグループを作成し、そのグループでアクティブインターフェイスを設定する必要があります。

手順の概要

1. **configure terminal**
2. **[no] ITD port-group port-group-name**
3. **[no] interface ethernet slot/port**
4. (任意) **copy running-config startup-config**

手順の詳細

| | コマンドまたはアクション | 目的 |
|--------|---|---|
| ステップ 1 | configure terminal 例： switch# configure terminal switch(config)# | グローバル コンフィギュレーション モードを開始します |
| ステップ 2 | [no] ITD port-group port-group-name 例： switch(config)# ITD port-group P1 switch(config-port-group)# | ポートグループを作成または削除します。 |
| ステップ 3 | 必須: [no] interface ethernet slot/port 例： switch(config)# interface ethernet 3/1 switch(config-port-group)# | ポートグループのアクティブ インターフェイスを設定します。 (注) ブレイクアウトポートは使用しないでください。 |
| ステップ 4 | (任意) copy running-config startup-config 例： switch(config-port-group)# copy running-config startup-config | 実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。 |

ITD レイヤ 2 サービスの構成

ITD レイヤ 2 サービスを構成するには、次の手順を実行する必要があります。

1. サービス名を構成します。
2. 送信元 VLAN を構成します。
3. ポート グループを ITD-L2 サービスに関連付ける
4. 負荷分散スキームを指定する

手順の概要

1. **configure terminal**
2. **[no] itd service-name**
3. **[no] port-group port-group-name**
4. (任意) **[no] load-balance method { src | dst } ip [buckets bucket-number] | mask-position mask-position**
5. **[no] source vlan vlan-range**
6. **no shut**

手順の詳細

| | 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 | 必須: [no] itd service-name 例： switch(config)# itd SER3 | ITD-L2 サービスを構成または無効にします。 |
| ステップ 3 | [no] port-group port-group-name 例： switch(config-itd)# port-group pg | ポートグループを ITD-L2 サービスに関連付けます。 |
| ステップ 4 | (任意) [no] load-balance method { src dst } ip [buckets bucket-number] mask-position mask-position 例： switch(config-itd)# load-balance method src ip | 負荷分散メソッドを構成します。 |
| ステップ 5 | [no] source vlan vlan-range 例： switch(config-itd)# source vlan 10-20 switch(config-itd)# | ITD-L2 サービスの VLAN のリストを構成します。 送信元 VLAN はトラフィックを分類します。 |

| | コマンドまたはアクション | 目的 |
|--------|--|--------------------|
| ステップ 6 | no shut 例： <pre>switch(config-itd)# no shut</pre> | ITD-L2 サービスのアクティブ化 |

ACL を ITD サービスに割り当てる

インクルードアクセスコントロールリスト (ACL) 機能を使用して、ITD サービスに ACL を割り当てることができます。この機能は、ACL 内の **permit** メソッドを使用するアクセスコントロールエントリ (ACE) ごとに、不要なトラフィックをフィルタリングし、IP アクセスリストとルートマップを生成して、許可されたトラフィックのロードバランシングを行います。ロードバランシングは、送信元または宛先 IP アドレスのいずれかを使用してサポートされます。

始める前に

ITD 機能がイネーブルであることを確認します。

ITD サービスに追加されるデバイス グループが構成されたことを確認します。

ITD サービスに割り当てられる ACL が構成されたことを確認します。

手順の概要

1. **configure terminal**
2. **[no] itd *itd-name***
3. **[no] device-group *device-group-name***
4. **[no] ingress interface *interface***
5. **[no] load-balance {method {src {ip | ip-l4port [tcp | udp] range *x y*} | dst {ip | ip-l4port [tcp | udp] range *x y*}} | buckets *bucket-number*}**
6. **[no] failaction node-per-bucket**
7. **access-list *acl-name***
 - IPv4 の場合： **access-list *acl4-name***
 - IPv6 の場合： **access-list IPv6 *acl6-name***
8. **[no] shutdown**
9. (任意) **copy running-config startup-config**

手順の詳細

| | コマンドまたはアクション | 目的 |
|--------|--|-----------------------------|
| ステップ 1 | configure terminal 例： <pre>switch# configure terminal switch(config)#</pre> | グローバル コンフィギュレーション モードを開始します |

| | コマンドまたはアクション | 目的 |
|--------|--|---|
| ステップ 2 | <p>[no] itd <i>itd-name</i></p> <p>例 :</p> <pre>switch(config)# itd servicel switch(config-itd)#</pre> | ITD サービスを設定し、ITD 構成モードを開始します。最大 32 文字の英数字を入力できます。 |
| ステップ 3 | <p>[no] device-group <i>device-group-name</i></p> <p>例 :</p> <pre>switch(config-itd)# device-group dg1</pre> | ITD サービスに既存のデバイス グループを追加します。 <i>device-group-name</i> は、デバイス グループの名前を指定します。最大 32 文字の英数字を入力できます。 |
| ステップ 4 | <p>[no] ingress interface <i>interface</i></p> <p>例 :</p> <pre>switch(config-itd)# ingress interface ethernet 4/1-10</pre> | ITD サービスに 1 つ以上のインターフェイスを追加します。 複数のインターフェイスは、カンマを (「,」) を使用して区切ります。インターフェイスの範囲は、ハイフン (「-」) を使用して指定します。 |
| ステップ 5 | <p>[no] load-balance {method {src {ip ip-l4port [tcp udp] range <i>x y</i>} dst {ip ip-l4port [tcp udp] range <i>x y</i>} buckets <i>bucket-number</i>}</p> <p>例 :</p> <pre>switch(config-itd)# load-balance method src ip buckets 16</pre> | <p>ITD サービスのロードバランシング オプションを設定します。</p> <p>オプションは次のとおりです。</p> <ul style="list-style-type: none"> • method : 送信先または接続先の IP アドレスベースの負荷またはトラフィック分散を指定します。 • buckets : 作成するバケットの数を指定します。1 つ以上のバケットが 1 つのノードにマップされています。バケットは 2 のべき乗数で設定する必要があります。範囲は 2 ~ 256 です。 <p>(注) 設定するバケットの数がノードの数より多い場合、バケットはすべてのノードにラウンドロビン方式で適用されます。</p> |
| ステップ 6 | <p>[no] failaction node-per-bucket</p> <p>例 :</p> <pre>switch(config-itd)# failaction node-per-bucket</pre> | ノード障害が発生すると、このノードに割り当てられたバケットは、残りのアクティブ ノードに分散されます。重みがノードに割り当てられている場合、分布はノードの重みに基づいています。 |
| ステップ 7 | <p>access-list <i>acl-name</i></p> <ul style="list-style-type: none"> • IPv4 の場合 : access-list <i>acl4-name</i> | ITD サービスに ACL を割り当てます。 |

| | コマンドまたはアクション | 目的 |
|--------|---|--|
| | <ul style="list-style-type: none"> IPv6 の場合 : access-list IPv6 acl6-name 例 : IPv4 : <pre>switch(config-itd)# access-list itd_d</pre> 例 : IPv6 <pre>switch(config-itd)# access-list ipv6 itd1_d</pre> 例 : マルチ ACL : <pre>switch(config-itd)# access-list test1 device-group-dg1 switch(config-itd)# access-list test2 device-group-dg2</pre> | (注) Cisco NX-OS リリース 9.3(3) 以降、ユーザーは 1 つの ITD サービスで最大 8 つのアクセスリストを設定でき、それぞれを独自のデバイス グループ (マルチ ACL) に関連付けるオプションを使用できます。特定のデバイス グループが 1 つのユーザー ACL に関連付けられている場合、そのデバイス グループが優先され、デフォルトのデバイス グループが上書きされます。この機能により、ITD はさまざまな ACL に一致するトラフィックをさまざまなデバイス グループにロード バランシングできます。 |
| ステップ 8 | [no] shutdown 例 : <pre>switch(config-itd)# no shutdown</pre> | ITD サービスをイネーブルにします。 |
| ステップ 9 | (任意) copy running-config startup-config 例 : <pre>switch(config-itd)# copy running-config startup-config</pre> | 実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。 |

無停止でのノードの追加または削除

ITD サービスをシャットダウンせずにデバイス グループ内のノードを追加または削除できる ITD セッションを構成できます。それによって、ITD サービスのシャットダウン時にトラフィックの中断を最小限にすることができます。

始める前に

ITD 機能がイネーブルであることを確認します。

デバイス グループと ITD サービスが構成されたことを確認します。

手順の概要

1. **configure terminal**
2. **itd session device-group device-group-name**
3. **[no] node ip {ipv4-address | ipv6-address}**
4. (任意) **probe track id**
5. **{commit | abort}**
6. (任意) **show itd session device-group [name]**

7. (任意) copy running-config startup-config

手順の詳細

| | 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 | itd session device-group device-group-name 例： switch(config)# itd session device-group dg1 switch(config-session-device-group)# | 指定されたデバイス グループの ITD セッションを作成します。 |
| ステップ 3 | [no] node ip {ipv4-address ipv6-address} 例： switch(config-session-device-group)# node ip 2.2.2.1 | 指定されたノードを ITD デバイス グループに追加します。このコマンドの no 形式は、指定されたノードを ITD デバイス グループから削除します。 追加または削除するノードごとに、この手順を繰り返します。 |
| ステップ 4 | (任意) probe track id 例： switch(config)# itd session device-group dg2 switch(config-session-device-group)#node ip 1.1.1.5 switch(config-session-device-group)#probe track 60 | ユーザー定義のトラックで新しいノードを追加します。 |
| ステップ 5 | {commit abort} 例： switch(config-session-device-group)# commit switch(config)# | commit コマンドは、新しいノードセットまたは変更されたノードセットで ITD デバイス グループを更新し、バケットを再割り当てして、ITD セッション設定をクリーンアップします。 abort コマンドは ITD セッション設定を無視し、ITD デバイス グループを更新しません。 |
| ステップ 6 | (任意) show itd session device-group [name] 例： switch(config)# show itd session device-group dg1 | 構成されたすべての ITD セッション、または指定されたデバイス グループの ITD セッションを表示します。 |
| ステップ 7 | (任意) copy running-config startup-config 例： switch(config)# copy running-config startup-config | 実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。 |

インクルード ACL での ACE の無停止の追加または削除

ITD サービスをシャットダウンせずに、インクルード ACL のアクセス コントロール エントリ (ACE) を追加または削除できます。それによって、ITD サービスのシャットダウン時にトラフィックの中断を最小限にすることができます。

始める前に

ITD 機能がイネーブルであることを確認します。

デバイス グループと ITD サービスが構成されたことを確認します。

ACL が ITD サービスに割り当てられていることを確認します。

手順の概要

1. **configure terminal**
2. **itd session access-list *acl-name* refresh**
3. (任意) **copy running-config startup-config**

手順の詳細

| | コマンドまたはアクション | 目的 |
|--------|---|--|
| ステップ 1 | configure terminal 例 : <pre>switch# configure terminal switch(config)#</pre> | 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 | itd session access-list <i>acl-name</i> refresh 例 : <pre>switch(config)# itd session access-list test1 refresh</pre> | インクルード ACL を内部的に読み取り、TCAM をプログラムします。ITD は、古い ACL ACE と新しい ACL ACE をチェックし、ITD によって生成された ACL を更新します。 (注) このコマンドは、インクルード ACL のみ必要です。 |
| ステップ 3 | (任意) copy running-config startup-config 例 : <pre>switch(config)# copy running-config startup-config</pre> | 実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。 |

インクルード ACL での ACE の無停止の追加または削除

ITD サービスをシャットダウンせずに、除外 ACL のアクセス コントロール エントリ (ACE) を追加または削除できます。そしてこれは、直ちに効果が出ます。

始める前に

ITD 機能がイネーブルであることを確認します。

デバイス グループと ITD サービスが構成されたことを確認します。

ACL が ITD サービスに割り当てられていることを確認します。

ITD レイヤ 3 構成の確認

ITD レイヤ 3 構成を表示するには、次のタスクのうちのいずれかを実行します。

| コマンド | 目的 |
|---|---|
| <code>show itd [itd-name] [brief vrf [vrf-name]]</code> | <p>特定の ITD インスタンスのステータスおよび構成を表示します。</p> <ul style="list-style-type: none"> 特定の ITD インスタンスのステータスおよび構成を表示するには、<code>itd-name</code> 引数を使用します。 ステータスおよび構成の要約情報を表示するには、<code>brief</code> キーワードを使用します。 <code>vrf</code> キーワードを使用して、指定された ITD インスタンスの VRF を表示します。 |
| <code>show itd session device-group [name]</code> | 構成されたすべての ITD セッションまたは指定されたデバイス グループの ITD セッションを表示します。 |
| <code>show running-config services</code> | 構成された ITD デバイス グループとサービスを表示します。 |

以下に、ITD 構成を確認する例を示します。

```
switch# show itd

Name          Probe LB Scheme  Status  Buckets
-----
WEB           ICMP  src-ip        ACTIVE  2

Device Group                                VRF-Name
-----
WEB-SERVERS

Pool          Interface  Status  Track_id
-----
WEB_itd_pool  Po-1      UP      -

Virtual IP    Netmask/Prefix  Protocol  Port
-----
```

```

10.10.10.100 / 255.255.255.255          IP          0

Node  IP          Config-State  Weight  Status  Track_id
-----
1     10.10.10.11   Active        1       OK      -

Bucket List
-----
WEB_itd_vip_1_bucket_1

Node  IP          Config-State  Weight  Status  Track_id
-----
2     10.10.10.12   Active        1       OK      -

Bucket List
-----
WEB_itd_vip_1_bucket_2

```

ITD Layer-2 構成の確認

ITD-L2 構成を表示するには、次のタスクのうちのいずれかを実行します。

表 3: ITD-L2 Show コマンド

| コマンド | 目的 |
|--|--|
| show ITD [<i>service-name</i>] brief | ITD の動作ステータスを表示します。 |
| show vlan access-map <i>vlan access-map name</i> | ITD-L2 サービスのトラフィックリダイレクトのネクストホップを表示します。 N9K-X9464PX では、VACL カウンタは、同じ VLAN で送受信している場合、送信されたパケット数が 2 倍であることを示します。 |
| show vlan access-list <i>vlan access-map name</i> | トラフィックの一致基準を指定する VLAN アクセスマップに関連付けられたパケットを表示します。 |
| show vlan-filter access-map <i>vlan access-map name</i> | アクセス VLAN マップへのマッピング元 VLAN を表示します。 |
| show running-config services | ITD-L2 の実行コンフィギュレーションを表示します。 |

ITD の構成例

以下に、ITD デバイス グループを設定する例を示します。

```

switch(config)# feature itd
switch(config)# itd device-group dg
switch(config-device-group)# node ip 210.10.10.11
switch(config-dg-node)# weight 6
switch(config-dg-node)# exit
switch(config-device-group)# node ip 210.10.10.12
switch(config-dg-node)# weight 6
switch(config-dg-node)# exit
switch(config-device-group)# node ip 210.10.10.13
switch(config-dg-node)# weight 2
switch(config-dg-node)# exit
switch(config-device-group)# node ip 210.10.10.14
switch(config-dg-node)# weight 2
switch(config-dg-node)# exit
switch(config-device-group)# probe icmp

```

この例は、複数の ITD デバイス グループ (http_servers および telnet_servers) を構成する方法を示しています。仮想 IP アドレスはデバイス グループごとに構成され、負荷分散バケットは仮想 IP アドレスごとにあります。

```

switch(config)# itd device-group http_servers
  probe icmp
  node ip 10.10.10.9
  node ip 10.10.10.10

switch(config)# itd device-group telnet_servers
  probe icmp
  node ip 1.1.1.1
  node ip 1.1.1.2

switch(config)# itd test
virtual ip 40.1.1.100 255.255.255.255 tcp 23 device-group telnet_servers
virtual ip 30.1.1.100 255.255.255.255 tcp 80 device-group http_servers
  ingress interface Eth3/1
  no shut

```

この例は、(デバイス グループ レベルのプローブではなく) ノード レベルのプローブを構成する方法を示しています。ノードレベルのプローブを行う場合、それぞれのノードは自身のプローブで構成可能なため、ノードごとにさらにカスタマイズすることができます。

```

switch(config)# feature itd
switch(config)# itd device-group Servers
switch(config-device-group)# node ip 192.168.1.10
switch(config-dg-node)# probe icmp frequency 10 retry-down-count 5
switch(config-device-group)# node ip 192.168.1.20
switch(config-dg-node)# probe icmp frequency 5 retry-down-count 5
switch(config-device-group)# node ip 192.168.1.30
switch(config-dg-node)# probe icmp frequency 20 retry-down-count 3

```

この例は、接続先 NAT を構成する方法を示しています

```

Itd device-group <dg1>
probe icmp
node ip 1.1.1.1
node ip 2.2.2.2

Itd device-group <dg2>
probe icmp
node ip 3.3.3.3
node ip 4.4.4.4

```

```

Itd test1
device-group <dg1>
virtual ip 10.10.10.10 255.255.255.255 tcp 80
nat destination

Itd test2
device-group <dg2>
virtual ip 30.30.30.30 255.255.255.255 tcp 80
nat destination
switch(config)# sh nat itd
      ACL (Bucket_List)           Global_IP(Node_IP):Port   Local_IP(Virtual_IP):Port
      Protocol
-----+-----+-----
ser1_itd_vip_1_bucket_1          8.8.1.2:0                 6.6.1.1:101
      TCP
ser1_itd_vip_1_bucket_21         8.8.1.2:0                 6.6.1.1:101
      TCP
ser1_itd_vip_1_bucket_2          8.8.1.3:0                 6.6.1.1:101
      TCP
ser1_itd_vip_1_bucket_22         8.8.1.3:0                 6.6.1.1:101
      TCP

```

ITD NAT および PAT の構成

```

feature itd

itd device-group dg1
  probe icmp
  node ip 10.10.10.10
  port 1000
  node ip 20.20.20.20
  port 2000
  node ip 30.30.30.30
  port 3000
  node ip 40.40.40.40
  port 4000

itd device-group dg2
  probe icmp
  node ip 10.10.10.11
  node ip 20.20.20.21
  port 2000
  node ip 30.30.30.31
  port 3000
  node ip 40.40.40.41
  port 4000

itd ser1
  virtual ip 6.6.6.1 255.255.255.255 tcp 80 advertise enable device-group dg1
  virtual ip 6.6.6.11 255.255.255.255 tcp 81 advertise enable device-group dg2
  ingress interface Eth1/1
  nat destination
  failaction node per-bucket
  load-balance method src ip buckets 64
  no shut

```

以下に、仮想 IPv4 アドレスを構成する例を示します。

```

switch(config)# feature itd
switch(config)# itd s4-101
switch(config-itd)# device-group dg_v4

```

```
switch(config-device-group)# ingress interface Vlan913
switch(config-device-group)# virtual ip 100.100.100.100 255.255.255.255 udp 443 advertise
enable active
```

以下に、仮想 IPv6 アドレスを構成する例を示します。

この例は、トラフィックを比例的に分散するように加重ロードバランシングを構成する方法を示しています。この例では、ノード 1 と 2 は、ノード 3 と 4 の 3 倍のトラフィックを受け取ります。

```
switch(config)# feature itd
switch(config)# itd device-group dg
switch(config-device-group)# probe icmp
switch(config-device-group)# node ip 210.10.10.11
switch(config-dg-node)# weight 3
switch(config-device-group)# node ip 210.10.10.12
switch(config-dg-node)# weight 3
switch(config-device-group)# node ip 210.10.10.13
switch(config-device-group)# node ip 210.10.10.14
```

この例は、除外 ACL を構成して、ITD が ITD ロードバランサから除外するトラフィックを指定する方法を示しています。たとえば、ファイアウォールインスペクションを必要としない開発者 VLAN およびテストベッド VLAN は、ITD をバイパスできます。

```
switch(config)# feature itd
switch(config)# itd Service_Test
switch(config-itd)# device-group test-group
switch(config-itd)# ingress interface vlan10
switch(config-itd)# exclude access-list ITDExclude
switch(config-itd)# no shutdown

switch(config)# ip access-list ITDExclude
switch(config-acl)# 10 permit ip 5.5.5.0/24 any
switch(config-acl)# 20 permit ip 192.168.100.0/24 192.168.200.0/24
```

この例は、acl1 を作成して ITD サービスに割り当てる方法を示しています。show コマンドは、生成された IP アクセス リストとルートマップを表示します。

```
switch(config)# ip access-list acl1
switch(config-acl)# 2460 permit tcp 100.1.1.0/24 any
switch(config-acl)# exit

switch(config)# itd test
switch(config-itd)# device-group dg1
switch(config-itd)# ingress interface Eth3/1
switch(config-itd)# load-balance method src ip
switch(config-itd)# access-list acl1
switch(config-itd)# show itd test
Legend:
ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name          LB Scheme  Status    Buckets
-----
test          src-ip     ACTIVE    4

Exclude ACL
-----
```

```

Device Group                               Probe Port
-----
dgl                                         ICMP

Pool                               Interface  Status Track_id
-----
test_itd_pool                       Eth3/1    UP      1

ACL Name/SeqNo                       IP/Netmask/Prefix                Protocol Port
-----
acl1/2460                            100.1.1.0/24                    TCP      0

Node  IP           Cfg-S  WGT Probe Port      Probe-IP  STS Trk# Sla_id
-----
1     1.1.1.1       Active  1   ICMP                    OK      2   10002
Bucket List
-----
test_itd_ace_1_bucket_1

Node  IP           Cfg-S  WGT Probe Port      Probe-IP  STS Trk# Sla_id
-----
2     1.1.1.2       Active  1   ICMP                    OK      3   10003
Bucket List
-----
test_itd_ace_1_bucket_2

Node  IP           Cfg-S  WGT Probe Port      Probe-IP  STS Trk# Sla_id
-----
3     10.10.10.9   Active  1   ICMP                    OK      4   10004
Bucket List
-----
test_itd_ace_1_bucket_3

Node  IP           Cfg-S  WGT Probe Port      Probe-IP  STS Trk# Sla_id
-----
4     10.10.10.10  Active  1   ICMP                    OK      5   10005
Bucket List
-----
test_itd_ace_1_bucket_4

```

Cisco NX-OS リリース 7.0(3)I7(3) 以降、ITD は IPv6 をサポートします。この例は、acl を作成し、ITDv4 および ITDv6 サービスに割り当てる方法を示しています。show コマンドは、生成された IP アクセスリストとルートマップを表示します。

```

switch(config)# IPv6 access list acl6-101
switch(config-acl)# 10 permit udp 2405:200:1412:2000::/96 any
switch(config-acl)# exit
switch(config)# IP access list acl4-101
switch(config)# 10 permit tcp 10.0.0.0/10 any
switch(config-acl)# exit

switch(config-itd)# device-group dg6-101
switch(config-itd)# ingress interface Vlan913
switch(config-itd)# failaction node reassign
switch(config-itd)# load-balance method src ip
switch(config-itd)# access-list ipv6 acl6-101
switch(config-itd)# no shut

switch(config-itd)# device-group dg4-101

```



```
switch(config-itd)# ingress interface Vlan913
switch(config-itd)# failaction node reassign
switch(config-itd)# load-balance method src ip
switch(config-itd)# access-list acl4-101
switch(config-itd)# no shut
```

この例では、ノード障害後に、障害が発生したノードバケットを、バケットの数が最も少ないアクティブノードに割り当てるように ITD サービスを構成する方法を示します。

```
switch(config-itd)# show run services

!Command: show running-config services
!Time: Thu Sep 22 22:22:01 2016
```

```
version 7.0(3)I5(1)
feature itd
```

```
itd session device-group dg
```

```
itd device-group dg
  probe icmp
  node ip 1.1.1.1
  node ip 2.2.2.2
  node ip 3.3.3.3
```

```
itd test
  device-group dg
  ingress interface Eth1/1
  failaction node least-bucket
  no shut
```

```
switch(config-itd)#
```

```
switch(config-itd)# show itd
```

Legend:

ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

| Name | LB Scheme | Status | Buckets |
|------|-----------|--------|---------|
| test | src-ip | ACTIVE | 4 |

Exclude ACL

| Device Group | Probe | Port |
|--------------|-------|------|
| dg | ICMP | |

| Pool | Interface | Status | Track_id |
|---------------|-----------|--------|----------|
| test_itd_pool | Eth1/1 | UP | 1 |

| Node | IP | Cfg-S | WGT | Probe | Port | Probe-IP | STS | Trk# | Sla_id |
|------|---------|--------|-----|-------|------|----------|-----|------|--------|
| 1 | 1.1.1.1 | Active | 1 | ICMP | | | OK | 2 | 10002 |

```

Bucket List
-----
test_itd_bucket_1, 4

Node  IP          Cfg-S  WGT Probe Port  Probe-IP  STS Trk# Sla_id
-----
2      2.2.2.2  Active  1 ICMP                OK   3  10003

Bucket List
-----
test_itd_bucket_2

Node  IP          Cfg-S  WGT Probe Port  Probe-IP  STS Trk# Sla_id
-----
3      3.3.3.3  Active  1 ICMP                OK   4  10004

Bucket List
-----
test_itd_bucket_3

switch(config-itd)#

# Brought down Node 3, and the failed node buckets are send to Node 2.

switch# show itd

Legend:
ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name          LB Scheme  Status  Buckets
-----
test          src-ip    ACTIVE  4

Exclude ACL
-----

Device Group          Probe Port
-----
dg                    ICMP

Pool          Interface  Status Track_id
-----
test_itd_pool Eth1/1    UP      1

Node  IP          Cfg-S  WGT Probe Port  Probe-IP  STS Trk# Sla_id
-----
1      1.1.1.1  Active  1 ICMP                OK   2  10002

Bucket List
-----
test_itd_bucket_1, 4

Node  IP          Cfg-S  WGT Probe Port  Probe-IP  STS Trk# Sla_id
-----
2      2.2.2.2  Active  1 ICMP                OK   3  10003

Bucket List
-----
test_itd_bucket_2

```

```

Node  IP              Cfg-S  WGT Probe Port  Probe-IP  STS Trk# Sla_id
-----
3     3.3.3.3  Active  1  ICMP
-----
          Bucket List
          -----
          test_itd_bucket_3

switch#
switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# end

switch#

この例では、ノード障害後に（1つのアクティブノードだけにではなく）使用可能なすべてのノードにトラフィックを均等に分散するようにITDサービスを構成する方法を示しています。

switch# show run services

!Command: show running-config services
!Time: Thu Sep 22 22:30:21 2016

version 7.0(3)I5(1)
feature itd

itd session device-group dg

itd device-group dg
  probe icmp
  node ip 1.1.1.1
  node ip 2.2.2.2
  node ip 3.3.3.3

itd test
  device-group dg
  ingress interface Eth1/1
  failaction bucket distribute
  no shut

switch#

switch# show itd
Legend:
  ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name          LB Scheme  Status  Buckets
-----
test          src-ip    ACTIVE  4

Exclude ACL
-----

Device Group          Probe  Port
-----
dg                    ICMP

```

```

Pool                Interface  Status Track_id
-----
test_itd_pool      Eth1/1    UP      1

```

```

Node  IP          Cfg-S  WGT Probe Port  Probe-IP  STS Trk# Sla_id
-----
1     1.1.1.1  Active  1 ICMP          OK      2  10002

```

```

Bucket List
-----
test_itd_bucket_1, 4

```

```

Node  IP          Cfg-S  WGT Probe Port  Probe-IP  STS Trk# Sla_id
-----
2     2.2.2.2  Active  1 ICMP          OK      3  10003

```

```

Bucket List
-----
test_itd_bucket_2

```

```

Node  IP          Cfg-S  WGT Probe Port  Probe-IP  STS Trk# Sla_id
-----
3     3.3.3.3  Active  1 ICMP          PF      4  10004

```

```

Bucket List
-----
test_itd_bucket_3

```

```
switch#
```

次の例は、ITDセッションを作成して、dgl デバイスグループにノードを無停止で追加する方法を示しています。

```

switch(config)# feature itd
switch(config)# itd device-group dgl
switch(config-device-group)# probe icmp
switch(config-device-group)# node ip 1.1.1.1
switch(config-dg-node)# node ip 2.1.1.1
switch(config-dg-node)# node ip 3.1.1.1
switch(config-dg-node)#
switch(config-dg-node)# itd test
switch(config-itd)# device-group dgl
switch(config-itd)# ingress interface Eth1/11
switch(config-itd)# load-balance method dst ip
Note: Configure buckets equal or more than the total number of nodes.

```

```

switch(config-itd)# access-list acl1
switch(config-itd)# no shut
switch(config-itd)# show itd test

```

```
Legend:
```

```
ST (Status): ST-Standby, LF-Link Failed, PF-Probe Failed, PD-Peer Down, IA-Inactive
```

```

Name          LB Scheme  Status  Buckets
-----
test          dst-ip    ACTIVE  4

```

```
Exclude ACL
-----
```

```

Device Group                Probe Port
-----
dgl                          ICMP

```

```

Pool                               Interface   Status Track_id
-----
test_itd_pool                       Eth1/11    UP        2

ACL Name
-----
acl1

Node  IP                Cfg-S  WGT Probe Port    Probe-IP  STS Trk# Sla_id
-----
1      1.1.1.1  Active  1 ICMP                               OK   3   10003
Bucket List
-----
test_itd_bucket_1, 4

Node  IP                Cfg-S  WGT Probe Port    Probe-IP  STS Trk# Sla_id
-----
2      2.1.1.1  Active  1 ICMP                               OK   4   10004
Bucket List
-----
test_itd_bucket_2

Node  IP                Cfg-S  WGT Probe Port    Probe-IP  STS Trk# Sla_id
-----
3      3.1.1.1  Active  1 ICMP                               OK   5   10005
Bucket List
-----
test_itd_bucket_3

switch(config-itd)# show run service
!Command: show running-config services
!Time: Tue Sep 20 20:36:04 2016
version 7.0(3)I5(1)
feature itd

itd device-group dg1
  probe icmp
  node ip 1.1.1.1
  node ip 2.1.1.1
  node ip 3.1.1.1
itd test
  device-group dg1
  ingress interface Eth1/11
  load-balance method dst ip
  access-list acl1
  no shut

switch(config-itd)# itd session device-group dg1
switch(config-session-device-group)# node ip 4.1.1.1
switch(config-session-dg-node)# commit
switch(config)# show itd test

Legend:
  ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name          LB Scheme  Status  Buckets
-----
test          dst-ip    ACTIVE  4

```

```

Exclude ACL
-----
Device Group                               Probe  Port
-----
dgl                                         ICMP

Pool           Interface  Status  Track_id
-----
test_itd_pool  Eth1/11  UP      2

ACL Name
-----
acl1

```

```

Node  IP           Cfg-S  WGT  Probe Port  Probe-IP  STS Trk#  Sla_id
-----
1     1.1.1.1  Active  1    ICMP                OK    3    10003

```

Bucket List

```
test_itd_bucket_1
```

```

Node  IP           Cfg-S  WGT  Probe Port  Probe-IP  STS Trk#  Sla_id
-----
2     2.1.1.1  Active  1    ICMP                OK    4    10004

```

Bucket List

```
test_itd_bucket_2
```

```

Node  IP           Cfg-S  WGT  Probe Port  Probe-IP  STS Trk#  Sla_id
-----
3     3.1.1.1  Active  1    ICMP                OK    5    10005

```

Bucket List

```
test_itd_bucket_3
```

```

Node  IP           Cfg-S  WGT  Probe Port  Probe-IP  STS Trk#  Sla_id
-----
4     4.1.1.1  Active  1    ICMP                OK    6    10006

```

Bucket List

```
test_itd_bucket_4
```

```
switch(config)# show run service
```

```
!Command: show running-config services
!Time: Tue Sep 20 20:37:14 2016
```

```
version 7.0(3)I5(1)
feature itd
```

```
itd device-group dgl
  probe icmp
  node ip 1.1.1.1
  node ip 2.1.1.1
  node ip 3.1.1.1
  node ip 4.1.1.1
```

```

itd test
  device-group dg1
  ingress interface Eth1/11
  load-balance method dst ip
  access-list acl1
  no shut

```

次の例は、ITDセッションを作成して、dg1 デバイスグループにノードを無停止で削除する方法を示しています。

```

switch(config)# feature itd
switch(config)#
switch(config)# itd device-group dg1
switch(config-device-group)# probe icmp
switch(config-device-group)# node ip 1.1.1.1
switch(config-dg-node)# node ip 2.1.1.1
switch(config-dg-node)# node ip 3.1.1.1
switch(config-dg-node)# node ip 4.1.1.1
switch(config-dg-node)#
switch(config-dg-node)# itd test
switch(config-itd)# device-group dg1
switch(config-itd)# ingress interface Eth1/11
switch(config-itd)# load-balance method dst ip
Note: Configure buckets equal or more than the total number of nodes.

switch(config-itd)# access-list acl1
switch(config-itd)# no shut

```

```
switch(config-itd)# show itd test
```

Legend:

ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

| Name | LB Scheme | Status | Buckets |
|------|-----------|--------|---------|
| test | dst-ip | ACTIVE | 4 |

Exclude ACL

| Device Group | Probe | Port |
|--------------|-------|------|
| dg1 | ICMP | |

| Pool | Interface | Status | Track_id |
|---------------|-----------|--------|----------|
| test_itd_pool | Eth1/11 | UP | 2 |

ACL Name

| Node | IP | Cfg-S | WGT | Probe Port | Probe-IP | STS | Trk# | Sla_id |
|------|---------|--------|-----|------------|----------|-----|------|--------|
| 1 | 1.1.1.1 | Active | 1 | ICMP | | OK | 3 | 10003 |

Bucket List

test_itd_bucket_1

| Node | IP | Cfg-S | WGT | Probe Port | Probe-IP | STS | Trk# | Sla_id |
|------|---------|--------|-----|------------|----------|-----|------|--------|
| 2 | 2.1.1.1 | Active | 1 | ICMP | | OK | 4 | 10004 |

```

Bucket List
-----
test_itd_bucket_2

Node  IP           Cfg-S  WGT  Probe Port      Probe-IP  STS Trk# Sla_id
-----
3     3.1.1.1  Active  1  ICMP              OK      5  10005

Bucket List
-----
test_itd_bucket_3

Node  IP           Cfg-S  WGT  Probe Port      Probe-IP  STS Trk# Sla_id
-----
4     4.1.1.1  Active  1  ICMP              OK      6  10006

Bucket List
-----
test_itd_bucket_4

switch(config-itd)# sh run service

!Command: show running-config services
!Time: Tue Sep 20 20:39:55 2016
version 7.0(3)I5(1)
feature itd

itd device-group dgl
  probe icmp
  node ip 1.1.1.1
  node ip 2.1.1.1
  node ip 3.1.1.1
  node ip 4.1.1.1

itd test
  device-group dgl
  ingress interface Eth1/11
  load-balance method dst ip
  access-list acl1
  no shut

switch(config-itd)# itd session device-group dgl
switch(config-session-device-group)# no node ip 4.1.1.1
switch(config-session-device-group)# commit
switch(config)# show itd test

Legend:
  ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name          LB Scheme  Status  Buckets
-----
test          dst-ip    ACTIVE  4

Exclude ACL
-----

Device Group          Probe  Port
-----
dgl                   ICMP

Pool                  Interface  Status Track_id

```



```

-----
test_itd_pool          Eth1/11      UP      2
ACL Name
-----
acl1

Node  IP          Cfg-S  WGT Probe Port      Probe-IP  STS Trk#  Sla_id
-----
1      1.1.1.1  Active  1 ICMP                OK    3    10003

Bucket List
-----
test_itd_bucket_1

Node  IP          Cfg-S  WGT Probe Port      Probe-IP  STS Trk#  Sla_id
-----
2      2.1.1.1  Active  1 ICMP                OK    4    10004

Bucket List
-----
test_itd_bucket_2

Node  IP          Cfg-S  WGT Probe Port      Probe-IP  STS Trk#  Sla_id
-----
3      3.1.1.1  Active  1 ICMP                OK    5    10005

Bucket List
-----
test_itd_bucket_3, 4

```

```
switch(config)# show run service
```

```
!Command: show running-config services
!Time: Tue Sep 20 20:41:07 2016
```

```
version 7.0(3)I5(1)
feature itd
itd device-group dgl
  probe icmp
  node ip 1.1.1.1
  node ip 2.1.1.1
  node ip 3.1.1.1
```

```
itd test
  device-group dgl
  ingress interface Eth1/11
  load-balance method dst ip
  access-list acl1
  no shut
```

```
switch(config)# sh itd test
```

```
Legend:
```

```
ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive
```

```
Name          LB Scheme  Status  Buckets
-----
test          src-ip    ACTIVE  n/a
```

```
Source Interface
```

```

-----
Device Group                                     Probe Port
-----
Pool                                             Interface Status Track_id
-----
                                         Eth1/3     UP      1

ACL Name                                         Buckets
-----
APP1                                             8

Device Group
-----
dg1

Node IP          Cluster-id Cfg-S   WGT Probe Port      Probe-IP  STS Trk#
Sla_id
-----
1              1.1.1.3          Active  1 ICMP              OK      3
10003

Bucket List
-----
test_itd_bucket_2, 1

Node IP          Cluster-id Cfg-S   WGT Probe Port      Probe-IP  STS Trk#
Sla_id
-----
2              1.1.1.4          Active  1 ICMP              OK      4
10004

Bucket List
-----
test_itd_bucket_3, 6

Node IP          Cluster-id Cfg-S   WGT Probe Port      Probe-IP  STS Trk#
Sla_id
-----
3              1.1.1.5          Active  1 ICMP              OK      5
10005

Bucket List
-----
test_itd_bucket_4, 5

Node IP          Cluster-id Cfg-S   WGT Probe Port      Probe-IP  STS Trk#
Sla_id
-----
4              1.1.1.2          Active  1 ICMP              OK      2
10010

Bucket List
-----
test_itd_bucket_8, 7

ACL Name                                         Buckets
-----
APP2                                             8

```

```

Device Group
-----
dg2

Node IP                Cluster-id Cfg-S   WGT Probe Port    Probe-IP  STS Trk#
Sla_id
-----
1          2.1.1.1            Active   1 ICMP                OK    6
10006

Bucket List
-----
test_itd_acl_1_bucket_1, 6

Node IP                Cluster-id Cfg-S   WGT Probe Port    Probe-IP  STS Trk#
Sla_id
-----
2          2.1.1.2            Active   1 ICMP                OK    7
10007

Bucket List
-----
test_itd_acl_1_bucket_2, 7

Node IP                Cluster-id Cfg-S   WGT Probe Port    Probe-IP  STS Trk#
Sla_id
-----
3          2.1.1.3            Active   1 ICMP                OK    8
10008

Bucket List
-----
test_itd_acl_1_bucket_3, 8

Node IP                Cluster-id Cfg-S   WGT Probe Port    Probe-IP  STS Trk#
Sla_id
-----
4          2.1.1.4            Active   1 ICMP                OK    9
10009

Bucket List
-----
test_itd_acl_1_bucket_4, 5

switch(config)# show run services

!Command: show running-config services
!Running configuration last done at: Sun Nov 15 12:09:30 2020
!Time: Sun Nov 15 12:15:10 2020

version 9.4(1) Bios:version N/A
feature itd

itd device-group dg1
  probe icmp frequency 1 timeout 1
  node ip 1.1.1.3
  node ip 1.1.1.4
  node ip 1.1.1.5

```

```

node ip 1.1.1.2

itd device-group dg2
  probe icmp frequency 1 timeout 1
  node ip 2.1.1.1
  node ip 2.1.1.2
  node ip 2.1.1.3
  node ip 2.1.1.4

itd test
  ingress interface Eth1/3
  failaction node least-bucket
  load-balance method src ip
  access-list APP1 device-group dg1
  access-list APP2 device-group dg2
  no shut

switch(config)# itd session device-group dg1
switch(config-session-device-group)# node ip 1.1.1.5
switch(config-session-dg-node)# weight 2
switch(config-session-dg-node)# node ip 1.1.1.4
switch(config-session-dg-node)# weight 3
switch(config-session-dg-node)# node ip 1.1.1.6
switch(config-session-dg-node)# weight 2
switch(config-session-dg-node)# no node ip 1.1.1.2
switch(config-session-device-group)# commit
switch(config)# sh itd test

```

Legend:

ST(Status): ST-Standby, LF-Link Failed, PF-Probe Failed, PD-Peer Down, IA-Inactive

| Name | LB Scheme | Status | Buckets |
|------|-----------|--------|---------|
| test | src-ip | ACTIVE | n/a |

Source Interface

| Device Group | Probe | Port |
|--------------|-------|------|
| | | |

| Pool | Interface | Status | Track_id |
|------|-----------|--------|----------|
| | Eth1/3 | UP | 1 |

| ACL Name | Buckets |
|----------|---------|
| APP1 | 8 |

Device Group

dg1

| Node | IP | Cluster-id | Cfg-S | WGT | Probe | Port | Probe-IP | STS | Trk# |
|-------|---------|------------|--------|-----|-------|------|----------|-----|------|
| 1 | 1.1.1.3 | | Active | 1 | ICMP | | | OK | 3 |
| 10003 | | | | | | | | | |

Bucket List

```

-----
test_itd_bucket_2
Node IP                Cluster-id Cfg-S   WGT Probe Port      Probe-IP  STS Trk#
Sla_id
-----
  2                1.1.1.4            Active   3 ICMP                OK   4
10004

Bucket List
-----
test_itd_bucket_3, 6, 7
Node IP                Cluster-id Cfg-S   WGT Probe Port      Probe-IP  STS Trk#
Sla_id
-----
  3                1.1.1.5            Active   2 ICMP                OK   5
10005

Bucket List
-----
test_itd_bucket_4, 5
Node IP                Cluster-id Cfg-S   WGT Probe Port      Probe-IP  STS Trk#
Sla_id
-----
  4                1.1.1.6            Active   2 ICMP                PF  10
10011

Bucket List
-----
test_itd_bucket_8, 1
ACL Name                Buckets
-----
APP2                    8

Device Group
-----
dg2

Node IP                Cluster-id Cfg-S   WGT Probe Port      Probe-IP  STS Trk#
Sla_id
-----
  1                2.1.1.1            Active   1 ICMP                OK   6
10006

Bucket List
-----
test_itd_acl_1_bucket_1, 6
Node IP                Cluster-id Cfg-S   WGT Probe Port      Probe-IP  STS Trk#
Sla_id
-----
  2                2.1.1.2            Active   1 ICMP                OK   7
10007

Bucket List
-----

```

```

test_itd_acl_1_bucket_2, 7

Node IP Cluster-id Cfg-S WGT Probe Port Probe-IP STS Trk#
Sla_id
-----
3 2.1.1.3 Active 1 ICMP OK 8
10008

Bucket List
-----
test_itd_acl_1_bucket_3, 8

Node IP Cluster-id Cfg-S WGT Probe Port Probe-IP STS Trk#
Sla_id
-----
4 2.1.1.4 Active 1 ICMP OK 9
10009

Bucket List
-----
test_itd_acl_1_bucket_4, 5

switch(config)# sh run services

!Command: show running-config services
!Running configuration last done at: Sun Nov 15 12:17:19 2020
!Time: Sun Nov 15 12:18:16 2020

version 9.4(1) Bios:version N/A
feature itd

itd device-group dg1
probe icmp frequency 1 timeout 1
node ip 1.1.1.3
weight 1
node ip 1.1.1.4
weight 3
node ip 1.1.1.5
weight 2
node ip 1.1.1.6
weight 2

itd device-group dg2
probe icmp frequency 1 timeout 1
node ip 2.1.1.1
node ip 2.1.1.2
node ip 2.1.1.3
node ip 2.1.1.4

itd test
ingress interface Eth1/3
failaction node least-bucket
load-balance method src ip
access-list APP1 device-group dg1
access-list APP2 device-group dg2
no shut

```

次の例は、ACE をインクルード ACL に中断することなく追加する方法を示しています。

```

switch(config)#
switch(config-acl)# ip access-list acl1
switch(config-acl)# 1010 permit tcp any 10.220.0.0/16
switch(config-acl)# 1020 permit tcp any 20.1.1.0/24

```

```
switch(config)# show ip access-lists acl1

IP access list acl1
  1010 permit tcp any 10.220.0.0/16
  1020 permit tcp any 20.1.1.0/24

switch(config)# itd device-group dg1
switch(config-device-group)# probe icmp
switch(config-device-group)# node ip 1.1.1.1
switch(config-dg-node)# node ip 2.1.1.1
switch(config-dg-node)# node ip 3.1.1.1
switch(config-dg-node)# node ip 4.1.1.1

switch(config-dg-node)# itd test
switch(config-itd)# device-group dg1
switch(config-itd)# ingress interface Eth1/11
switch(config-itd)# load-balance method dst ip
Note: Configure buckets equal or more than the total number of nodes.

switch(config-itd)# access-list acl1
switch(config-itd)# no shut

switch(config)# show run service

!Command: show running-config services
!Time: Tue Sep 20 20:44:17 2016

version 7.0(3)I5(1)
feature itd

itd device-group dg1
  probe icmp
  node ip 1.1.1.1
  node ip 2.1.1.1
  node ip 3.1.1.1
  node ip 4.1.1.1

itd test
  device-group dg1
  ingress interface Eth1/11
  load-balance method dst ip
  access-list acl1
  no shut

switch(config-itd)# ip access-list acl1
switch(config-acl)# 1030 permit tcp any 30.1.1.0/24
switch(config-acl)# exit
switch(config)# itd session access-list acl1 refresh
switch(config)# sh ip access-lists | grep n 4 itd_
IP access list test_itd_bucket_1
  1010 permit tcp any 10.220.0.0 0.0.63.255
  1020 permit tcp any 20.1.1.0 0.0.0.63
  1030 permit tcp any 30.1.1.0/26
IP access list test_itd_bucket_2
  1010 permit tcp any 10.220.64.0 0.0.63.255
  1020 permit tcp any 20.1.1.64 0.0.0.63
  1030 permit tcp any 30.1.1.64/26
IP access list test_itd_bucket_3
  1010 permit tcp any 10.220.128.0 0.0.63.255
  1020 permit tcp any 20.1.1.128 0.0.0.63
  1030 permit tcp any 30.1.1.128/26
```

```

IP access list test_itd_bucket_4
    1010 permit tcp any 10.220.192.0 0.0.63.255
    1020 permit tcp any 20.1.1.192 0.0.0.63
    1030 permit tcp any 30.1.1.192/26
switch(config)# sh run rpm
interface Ethernet1/11
    ip policy route-map test_itd_pool

```

この例では、アクセスリストが適切に生成され、予想される ip 一致条件があることを確認します。Cisco Nexus リリース 9.3 (3) F 以降では、**show ip access-list dynamic** コマンドを使用してシステム内の ACL を検索できます。

```
Nexus# show ip access-lists CiscoService_itd_vip_1_bucket_1 dynamic
```

```

IP access list CiscoService_itd_vip_1_bucket_1
    10 permit ip 1.1.1.0 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_2 dynamic

IP access list CiscoService_itd_vip_1_bucket_2
    10 permit ip 1.1.1.32 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_3 dynamic

IP access list CiscoService_itd_vip_1_bucket_3
    10 permit ip 1.1.1.64 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_4 dynamic

IP access list CiscoService_itd_vip_1_bucket_4
    10 permit ip 1.1.1.96 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_5 dynamic

IP access list CiscoService_itd_vip_1_bucket_5
    10 permit ip 1.1.1.128 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_6 dynamic

IP access list CiscoService_itd_vip_1_bucket_6
    10 permit ip 1.1.1.160 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_7 dynamic

IP access list CiscoService_itd_vip_1_bucket_7
    10 permit ip 1.1.1.192 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_8 dynamic

IP access list CiscoService_itd_vip_1_bucket_8
    10 permit ip 1.1.1.224 255.255.255.31 192.168.255.1/32

```

次の例は、インクルード ACL から ACE を中断なく削除する方法を示しています。

```

switch(config)# feature itd

switch(config-acl)# ip access-list acl1
switch(config-acl)# 1010 permit tcp any 10.220.0.0/16
switch(config-acl)# 1020 permit tcp any 20.1.1.0/24
switch(config-acl)# 1030 permit tcp any 30.1.1.0/24

switch(config)# itd device-group dgl
switch(config-device-group)# probe icmp
switch(config-device-group)# node ip 1.1.1.1
switch(config-dg-node)# node ip 2.1.1.1
switch(config-dg-node)# node ip 3.1.1.1
switch(config-dg-node)# node ip 4.1.1.1
switch(config-dg-node)#
switch(config-dg-node)# itd test
switch(config-itd)# device-group dgl
switch(config-itd)# ingress interface Eth1/11

```



```

switch(config-itd)# load-balance method dst ip
Note: Configure buckets equal or more than the total number of nodes.

switch(config-itd)# access-list acl1
switch(config-itd)# no shut

switch(config-acl)# sh itd test

Legend:
ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name          LB Scheme  Status  Buckets
-----
test          dst-ip    ACTIVE  4

Exclude ACL
-----
Device Group                                Probe  Port
-----
dg1                                                ICMP

Pool          Interface  Status  Track_id
-----
test_itd_pool Eth1/11    UP      2

ACL Name
-----
acl1

Node  IP          Cfg-S  WGT  Probe Port  Probe-IP  STS  Trk#  Sla_id
-----
1     1.1.1.1  Active  1    ICMP                OK    3    10003

Bucket List
-----
test_itd_bucket_1
Node  IP          Cfg-S  WGT  Probe Port  Probe-IP  STS  Trk#  Sla_id
-----
2     2.1.1.1  Active  1    ICMP                OK    4    10004

Bucket List
-----
test_itd_bucket_2

Node  IP          Cfg-S  WGT  Probe Port  Probe-IP  STS  Trk#  Sla_id
-----
3     3.1.1.1  Active  1    ICMP                OK    5    10005

Bucket List
-----
test_itd_bucket_3

Node  IP          Cfg-S  WGT  Probe Port  Probe-IP  STS  Trk#  Sla_id
-----
4     4.1.1.1  Active  1    ICMP                OK    6    10006

Bucket List
-----
test_itd_bucket_4

switch(config)# show itd test

```

```

Legend:
  ST (Status): ST-Standby, LF-Link Failed, PF-Probe Failed, PD-Peer Down, IA-Inactive
Name          LB Scheme  Status  Buckets
-----
test          dst-ip    ACTIVE  4

Exclude ACL
-----

Device Group                                Probe  Port
-----
dgl                                                ICMP

Pool          Interface  Status  Track_id
-----
test_itd_pool Eth1/11    UP      2

ACL Name
-----
acl1
Node  IP          Cfg-S  WGT  Probe Port  Probe-IP  STS  Trk#  Sla_id
-----
1          1.1.1.1  Active  1  ICMP                OK    3    10003

  Bucket List
  -----
  test_itd_bucket_1

Node  IP          Cfg-S  WGT  Probe Port  Probe-IP  STS  Trk#  Sla_id
-----
2          2.1.1.1  Active  1  ICMP                OK    4    10004

  Bucket List
  -----
  test_itd_bucket_2

Node  IP          Cfg-S  WGT  Probe Port  Probe-IP  STS  Trk#  Sla_id
-----
3          3.1.1.1  Active  1  ICMP                OK    5    10005

  Bucket List
  -----
test_itd_bucket_3

Node  IP          Cfg-S  WGT  Probe Port  Probe-IP  STS  Trk#  Sla_id
-----
4          4.1.1.1  Active  1  ICMP                OK    6    10006

  Bucket List
  -----
  test_itd_bucket_4

switch(config)# sh run rpm

```

次の例は、ITD over VXLAN を構成する方法を示しています。

```
switch(config)# sh itd brief
```

```

Legend:
C-S (Config-State): A-Active, S-Standby, F-Failed
  ST (Status): ST-Standby, LF-Link Failed, PF-Probe Failed, PD-Peer Down, IA-Inactive
Name          LB Scheme  Status  Buckets  Interface

```

```

-----
ser1          src-ip    ACTIVE  256    VLAN100,Eth1/1

Source Interface
-----
loopback9
VRF-Name
-----Org1:vrf1
Device Group                                Probe  Port
-----
sf
Virtual IP                                Netmask/Prefix  Protocol  Port
-----
6.6.6.1 / 255.255.255.0                    IP          0

Node      IP              Cfg-S      WGT      Probe  Port  Probe-IP  STS
-----
1         10.200.1.2     Active     1        -----
2         10.200.6.2     Active     1        -----
                                         OK
                                         OK

```

次の例は、バケット配布を使用して ITD ノード レベル スタンバイを構成する方法を示しています。

```

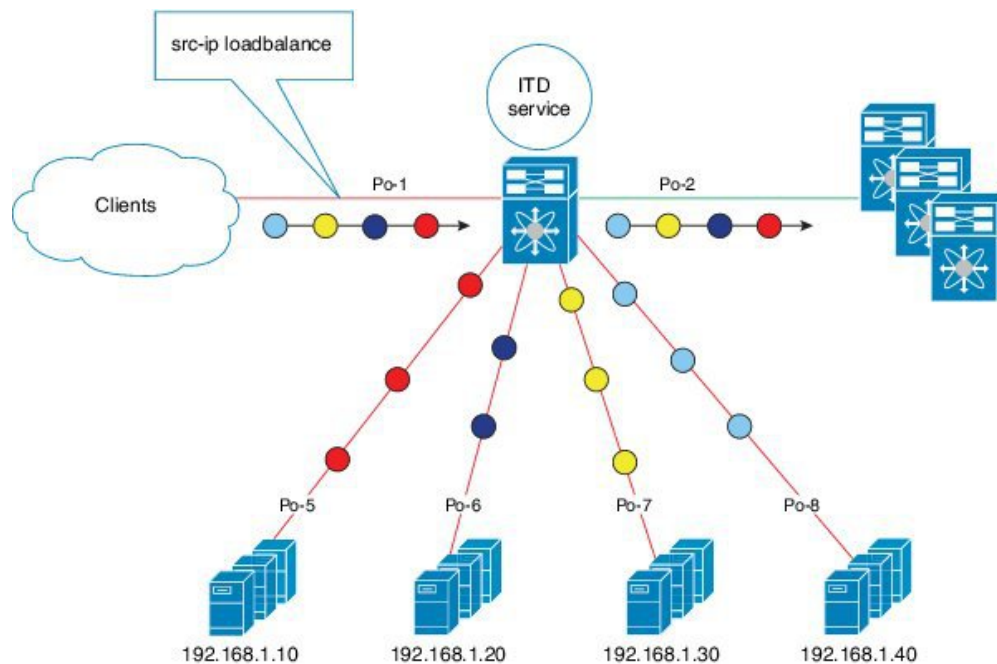
itd device-group dg
probe icmp
node ip 10.10.10.2
standby ip 13.13.13.2
node ip 11.11.11.2
standby ip 12.12.12.2
node ip 12.12.12.2
standby ip 11.11.11.2
node ip 13.13.13.2
standby ip 10.10.10.2
itd test
device-group dg
virtual ip 20.20.20.20.255.255.255.255 tcp 80 advertise enable
ingress interface Eth1/9
failaction bucket distribute
load-balance buckets 16
no shut

```

構成例 : ワンアーム展開モード

以下の構成は次の図のトポロジを使用します。

図 13: ワンアーム展開モード



3815161

ステップ 1 : デバイス グループを定義します。

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 210.10.10.11
switch(config-device-group)# node ip 210.10.10.12
switch(config-device-group)# node ip 210.10.10.13
switch(config-device-group)# node ip 210.10.10.14
switch(config-device-group)# probe icmp
```

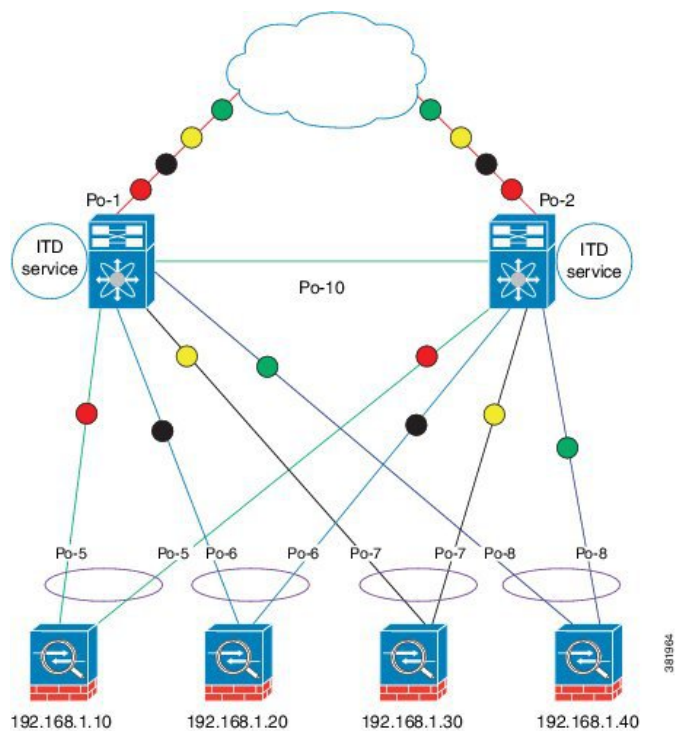
ステップ 2 : ITD サービスを定義します。

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 1
switch(config-itd)# device-group DG
switch(config-itd)# no shutdown
```

構成例 : vPC でのワンアーム展開モード

以下の構成は次の図のトポロジを使用します。

図 14: VPC でのワンアーム展開モード



デバイス 1

ステップ 1 : デバイス グループを定義します。

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 210.10.10.11
switch(config-device-group)# node ip 210.10.10.12
switch(config-device-group)# node ip 210.10.10.13
switch(config-device-group)# node ip 210.10.10.14
switch(config-device-group)# probe icmp
```

ステップ 2 : ITD サービスを定義します。

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 1
switch(config-itd)# device-group DG
switch(config-itd)# no shutdown
```

デバイス 2

ステップ 1 : デバイス グループを定義します。

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 210.10.10.11
switch(config-device-group)# node ip 210.10.10.12
switch(config-device-group)# node ip 210.10.10.13
```

```
switch(config-device-group)# node ip 210.10.10.14
switch(config-device-group)# probe icmp
```

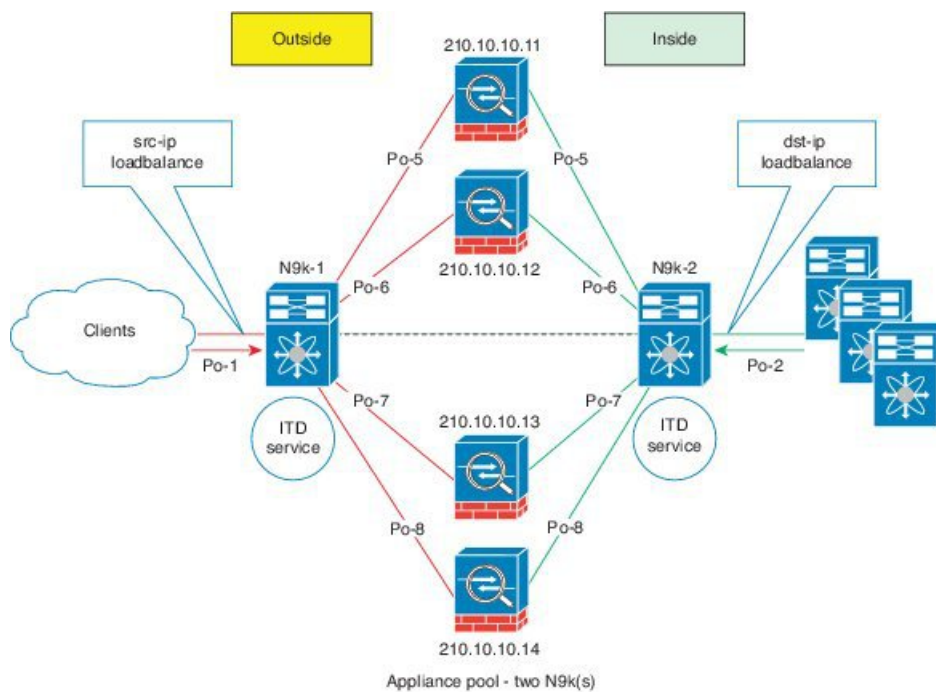
ステップ 2：ITD サービスを定義します。

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 2
switch(config-itd)# device-group DG
switch(config-itd)# no shutdown
```

構成例：サンドイッチ展開モード

以下の構成は次の図のトポロジを使用します。

図 15: サンドイッチ展開モード



3-40385

デバイス 1

ステップ 1：デバイス グループを定義します。

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 210.10.10.11
switch(config-device-group)# node ip 210.10.10.12
switch(config-device-group)# node ip 210.10.10.13
switch(config-device-group)# node ip 210.10.10.14
switch(config-device-group)# probe icmp
```

ステップ 2：ITD サービスを定義します。

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 1
switch(config-itd)# device-group DG
switch(config-itd)# load-balance method src ip
switch(config-itd)# no shutdown
```

デバイス2

ステップ 1：デバイス グループを定義します。

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 220.10.10.11
switch(config-device-group)# node ip 220.10.10.12
switch(config-device-group)# node ip 220.10.10.13
switch(config-device-group)# node ip 220.10.10.14
switch(config-device-group)# probe icmp
```

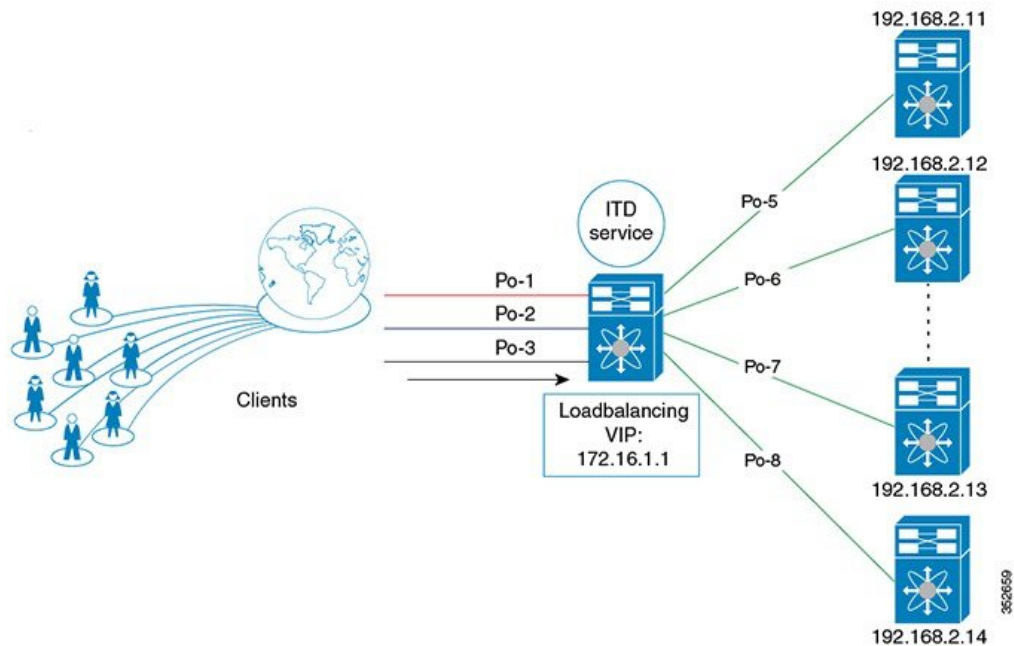
ステップ 2：ITD サービスを定義します。

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 2
switch(config-itd)# device-group DG
switch(config-itd)# load-balance method dst ip
switch(config-itd)# no shutdown
```

構成例：サーバー ロードバランシング展開モード

以下の構成は次の図のトポロジを使用します。

図 16: VIP を使用した ITD 負荷分散



ステップ 1 : デバイス グループを定義します。

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 192.168.2.11
switch(config-device-group)# node ip 192.168.2.12
switch(config-device-group)# node ip 192.168.2.13
switch(config-device-group)# node ip 192.168.2.14
switch(config-device-group)# probe icmp
```

ステップ 2 : ITD サービスを定義します。

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 1
switch(config-itd)# ingress interface port-channel 2
switch(config-itd)# ingress interface port-channel 3
switch(config-itd)# device-group DG
Switch(config-itd)# virtual ip 172.16.1.1 255.255.255.255
switch(config-itd)# no shutdown
```

構成例 : WCCP として ITD を再配置する (Web プロキシ展開モード)

プロキシサーバーは、他のサーバーからのリソースを求めるクライアントからの要求の仲介として機能します。Web プロキシサーバーは、特にローカル ネットワークとインターネット間の仲介役として機能します。通常、Web プロキシサーバーでは、ネットワーク デバイスがインターネットに向かう Web トラフィックを自分にリダイレクトする必要があります (転送フロー)。ただし、後続の packets 転送では、ネットワーク デバイスが packets を定期的に転送するだけで済みます。

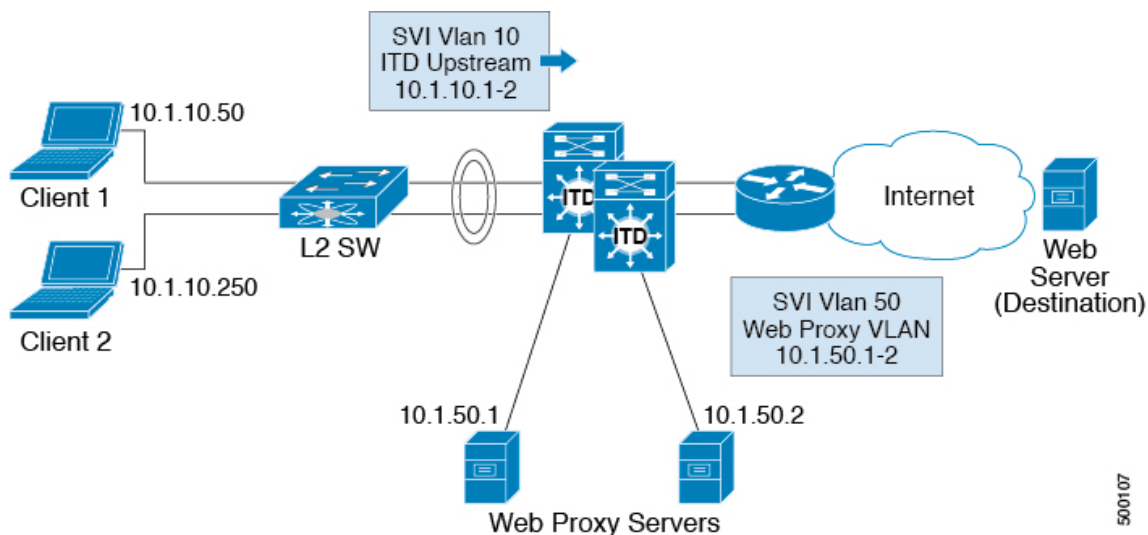
ITD を使用した Web プロキシ展開では、スイッチはインターネットに向かう Web トラフィックを照合し、プロキシサーバーに向けて負荷を分散します。プロキシサーバーは自律モード（WCCP から独立してアクティブ-アクティブ）で動作し、プロキシサーバーにリダイレクトされるトラフィックを処理します。ITD を介して実行されるノードヘルス プローブは、ノードの状態を追跡し、可用性に基づいて適切にノードを削除または追加するという目的を果たします。スタンバイサーバーは、冗長性のためにグループレベルまたはノードレベルで構成することもできます。

ITD リダイレクションは、通常、クライアント側 VLAN の順方向でのみ必要です。その後、パケットは ITD リダイレクションまたは配布なしでルーティングまたは転送されます。このような Web プロキシ展開を使用する ITD には、順方向用に構成された 1 つの ITD サービスのみが必要です。ただし、送信元レイヤ 4 ポートに基づいてトラフィックを選択して、リバーストラフィックリダイレクションが必要です。LB パラメータを逆にして、フローの対称性も維持する必要があります。

Web プロキシ展開の ITD では、ITD プローブを使用して Web プロキシサーバーの可用性をチェックします。これは、障害が発生したプロキシサーバーに送信されたトラフィックが失われるため重要です。

以下の構成は次の図のトポロジを使用します。

図 17: Web プロキシ展開モード



この例では、インターネットへの宛先ポート 80/443（入力 VLAN 10）が Web プロキシサーバー 10.1.50.1 および 10.1.50.2 に配布されます。プライベートネットワーク（10.0.0.0/8、192.168.0.0/16、172.16.0.0/12）宛ての VLAN 10 上のトラフィックは、プロキシに送信されません。

ステップ 0 : アクセスリストの構成

```
ip access-list ACL1
 10 permit ip any any tcp 80
 20 permit ip any any tcp 443
```

ステップ 1 : ITD デバイス グループの Web プロキシサーバーを設定し、サーバーの IP アドレスを指定します。

```
itd device-group Web_Proxy_Servers
  probe icmp
  node ip 10.1.50.1
  node ip 10.1.50.2
```

ステップ 2 : プライベート IP アドレス宛てのすべてのトラフィックを除外するように除外 ACL を構成します。

```
ip access-list itd_exclude_ACL
  10 permit ip any 10.0.0.0/8
  20 permit ip any 192.168.0.0/16
  30 permit ip any 172.16.0.0/12
```

ステップ 3 : 除外 ACL を適用します。

```
Itd Web_proxy_SERVICE
  device-group Web_Proxy_Servers
  exclude access-list itd_exclude_ACL
  access-list ACL1
  ingress interface Vlan 10
  failaction node reassign
  load-balance method src ip
  no shutdown
```

なんらかの理由でリターントラフィックのリダイレクトも必要な場合は、次の追加の構成手順が必要です。



(注) レイヤ 4 範囲演算子を使用したポート フィルタリングのみが可能です。また、除外 ACL は許可エントリのみをサポートします。

ステップ 4 : ポート 80 と 443 を除くすべてを除外するように、リターン除外 ACL を構成します。

```
ip access-list itd_exclude_return
  10 permit tcp any range 0 79 any
  20 permit tcp any range 81 442 any
  30 permit tcp any range 444 65535 any
```

ステップ 5 : リターン トラフィックのリターン ITD サービスを構成し、除外 ACL を適用します。

```
Itd Web_proxy_SERVICE
  device-group Web_Proxy_Servers
  exclude access-list itd_exclude_return
  ingress interface Vlan 20 <- Internet-facing ingress interface on the Nexus switch
  failaction node reassign
  load-balance method dst ip <- Flow symmetry between forward/return flow achieved by
  flipping the LB parameter
  no shutdown
```

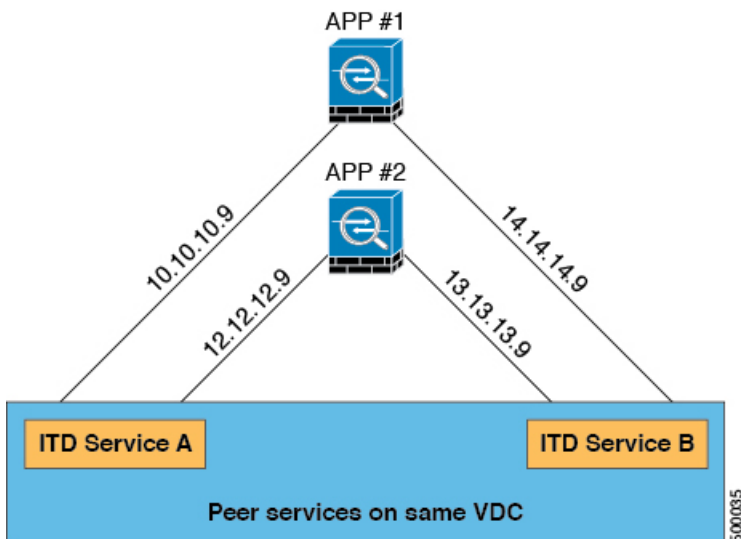
構成例：サンドイッチモード向けピア同期

ITD ピア サービス上のサンドイッチ アプライアンスへのリンクがダウンすると、サービスはノードへのリンクがダウンしていることを示す通知をピアに送信します。次に、ピアサービスはリンクをダウンさせ、トラフィックがそのリンクを通過しないようにします。

ピア同期なしで、ITD サービス A のアプライアンス APP #1 に接続されているリンクが次のトポロジでダウンし、ITD サービス B に通知されない場合、サービス B は引き続き APP #1 にトラフィックを送信し、トラフィックはドロップされます。

以下の構成では、このトポロジを使用します。

図 18: サンドイッチモードのピア同期



デバイス 1

ステップ 1：デバイス グループを定義します。

```
switch(config)# itd device-group dev-A
switch(config-device-group)# node ip 10.10.10.9 ---> Link to app #1
switch(config-device-group)# node ip 12.12.12.9 ---> Link to app #2
switch(config-device-group)# probe icmp
```

ステップ 2：ピア同期を有効にして ITD サービスを定義します。

```
switch(config)# itd service-A
switch(config-itd)# device-group dev-A
switch(config-itd)# ingress interface ethernet 7/4
switch(config-itd)# peer local service service-B
switch(config-itd)# no shutdown
```

```
switch(config-itd)# show itd
Name           Probe LB Scheme  Status  Buckets
-----
Service-A      ICMP  src-ip          ACTIVE  2
```

```

Device Group                                     VRF-Name
-----
Dev-A

Route Map                                     Interface   Status Track_id
-----
Service-A_itd_pool                           Eth7/45     UP        3

Node  IP                               Config-State Weight Status   Track_id Sla_id
-----
1     10.10.10.9                       Active      1       Peer Down 1       10001

IP Access List
-----
Service-A_itd_bucket_0

Node  IP                               Config-State Weight Status   Track_id Sla_id
-----
2     12.12.12.9                       Active      1       OK        2       10002

IP Access List
-----
Service-A_itd_bucket_1

```

デバイス2

ステップ1：デバイス グループを定義します。

```

switch(config)# itd device-group dev-B
switch(config-device-group)# node ip 14.14.14.9 ---> Link to app #1
switch(config-device-group)# node ip 13.13.13.9 ---> Link to app #2
switch(config-device-group)# probe icmp

```

ステップ2：ピア同期を有効にして ITD サービスを定義します。

```

switch(config)# itd service-B
switch(config-itd)# device-group dev-B
switch(config-itd)# ingress interface ethernet 7/45
switch(config-itd)# peer local service service-A
switch(config-itd)# no shutdown

switch(config-itd)# show itd
Name          Probe LB Scheme  Status  Buckets
-----
Service-B     ICMP  src-ip          ACTIVE  2

Device Group                                     VRF-Name
-----
Dev-B

Route Map                                     Interface   Status Track_id
-----
Service-B_itd_pool                           Eth7/45     UP        3

Node  IP                               Config-State Weight Status   Track_id Sla_id
-----

```

```

1      14.14.14.9      Active      1      Probe Failed      3      10003

IP Access List
-----
Service-B_itd_bucket_0

Node  IP                Config-State  Weight  Status      Track_id  Sla_id
-----
2      13.13.13.9      Active      1      OK          4          10004

IP Access List
-----
Service-B_itd_bucket_1

```

構成例：スティックのファイアーウォール

ITD サービス

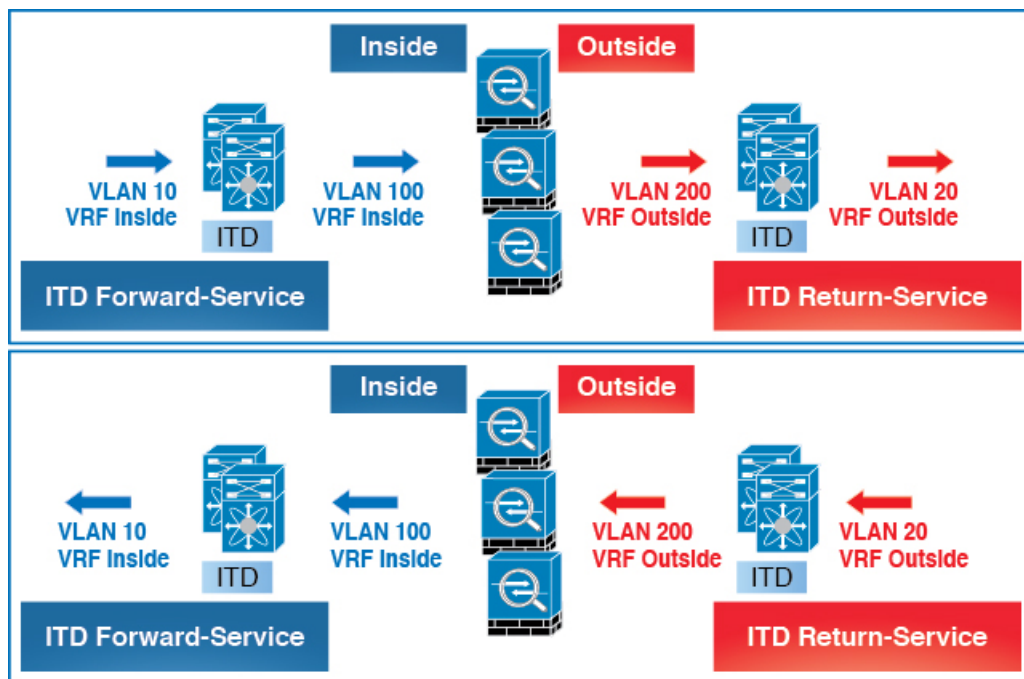
ITD サービス構成は、トラフィック フローの特定の方向に対する ITD トラフィック分散を定義します。フローの両方向をリダイレクトする必要がある場合は、2つの ITD サービスを設定する必要があります。1つは転送トラフィック フロー用、もう1つはリターントラフィック フロー用です。ASA には異なる内部インターフェイスと外部インターフェイスの IP アドレスがあるため、2つの異なるデバイス グループも、対応する内部および外部 IP アドレスを指すように構成する必要があります。

ASA VLAN

ITD 転送およびリターンサービスは、Nexus スイッチの内部および外部 VLAN SVI に接続されます。ファイアウォールなどのセキュリティアプリケーションはすべてのトラフィックを検査する必要があるため、サービスでトラフィックフィルタリングは構成されません。その結果、SVI に到達するトラフィックはすべて、対応する ASA インターフェイスにリダイレクトされます。

ASA インターフェイスがスイッチの VLAN と同じ VLAN で構成されている場合、ファイアウォールからスイッチに向かうトラフィックは、スイッチの別の VLAN に ITD サービスが存在するため、ASA にリダイレクトされます。したがって、ファイアウォールと Nexus スイッチ間のトラフィックループを防止するには、個別の VLAN のペアが必要です。

図 19: ITD ASA の展開



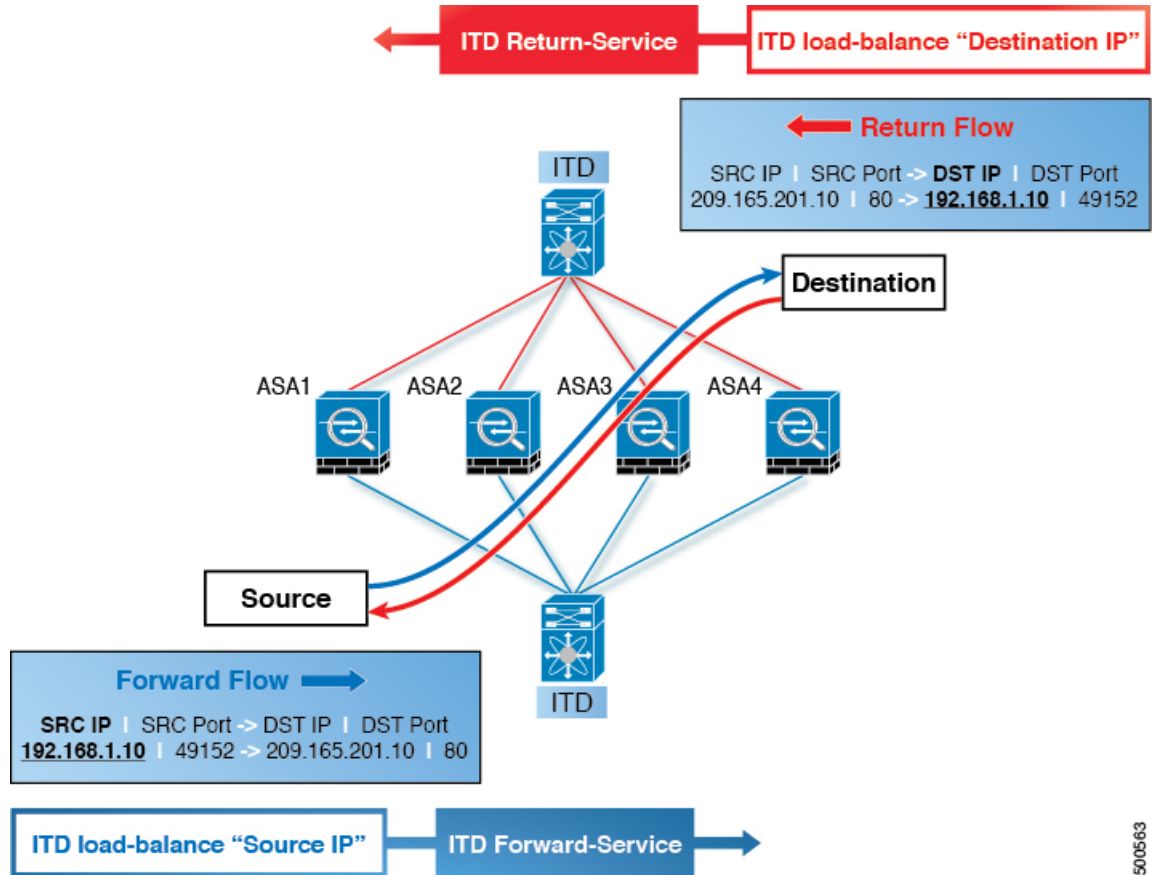
この図は、VLAN 10 および 20 を、ネットワーク上の送信元および接続先への内部および外部インターフェイスとして示しています。VLAN 100 および 200 は、ループのないトラフィックを確保するために ASA に対して使用されます。

フローの対称性

ファイアウォールは通常、順方向と戻り方向の両方のトラフィックフローを検査します。インスペクションのステートフルな性質により、通常、クラスタ化されていないファイアウォールの通常の操作中にフローの対称性を維持する必要があります。クラスタ化されたファイアウォールの場合でも、トラフィックフローの非対称性により、クラスタ制御リンクを介したフローのリダイレクトが増加します。非対称フローが増えると、ファイアウォールに不要なオーバーヘッドが追加され、パフォーマンスが低下します。

フローの対称性は、固有の IP 永続性と ITD アルゴリズムの決定論的性質を使用して実現できます。ファイアウォールの一般的な ITD 構成では、転送フローに 1 つの ITD サービスを使用し、リターンフローに 1 つの ITD サービスを使用します。ロードバランスパラメータの値が両方のサービスで同じになるようにこれら 2 つの ITD サービスを設定すると、フローの対称性が確実に維持されます。

図 20: ITD ASA 展開におけるフローの対称性

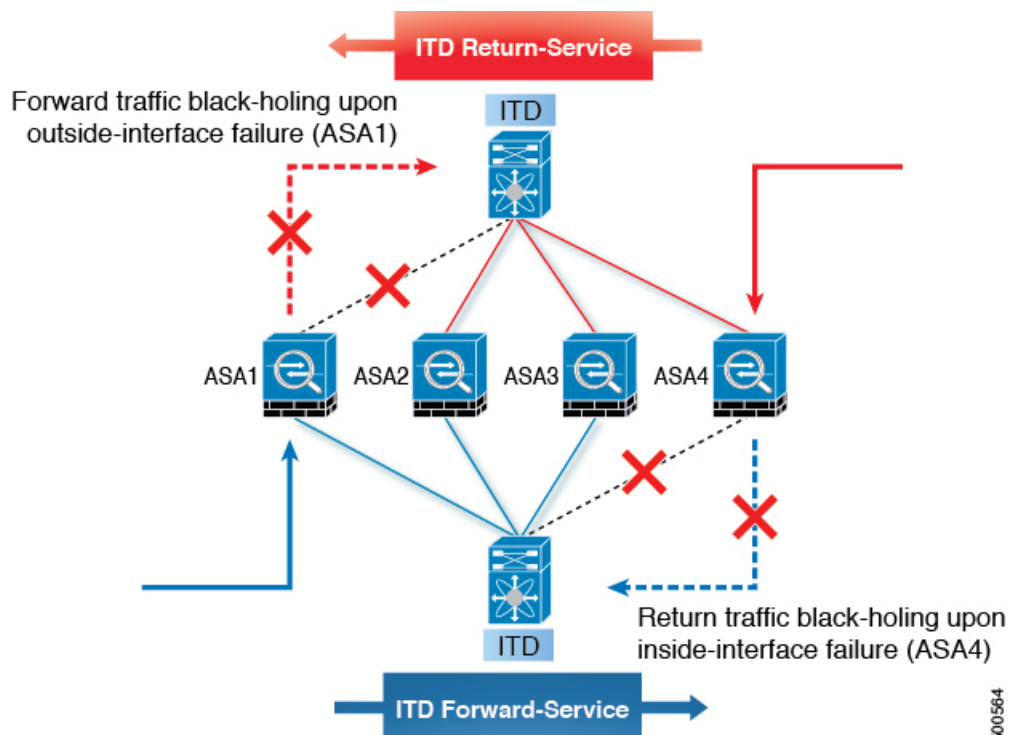


この図は、順方向フローの送信元 IP アドレスと逆方向フローの宛先 IP アドレスがどのように一定であることを示しています。各 ITD サービスに適切なパラメータを選択すると、ITD IP の永続性によるフローの対称性が保証されます。

Link Failures

ASA の内部または外部インターフェイスに障害が発生すると、トラフィックの出力インターフェイスがダウンしているため、その ASA の反対側に着信するトラフィックが失われる可能性があります。ITD ピア スイッチ ノード状態同期機能は、ASA のリモート側を ITD から削除し、スイッチ間でノード状態を同期することにより、この問題を解決できます。

図 21: ASA 障害シナリオ

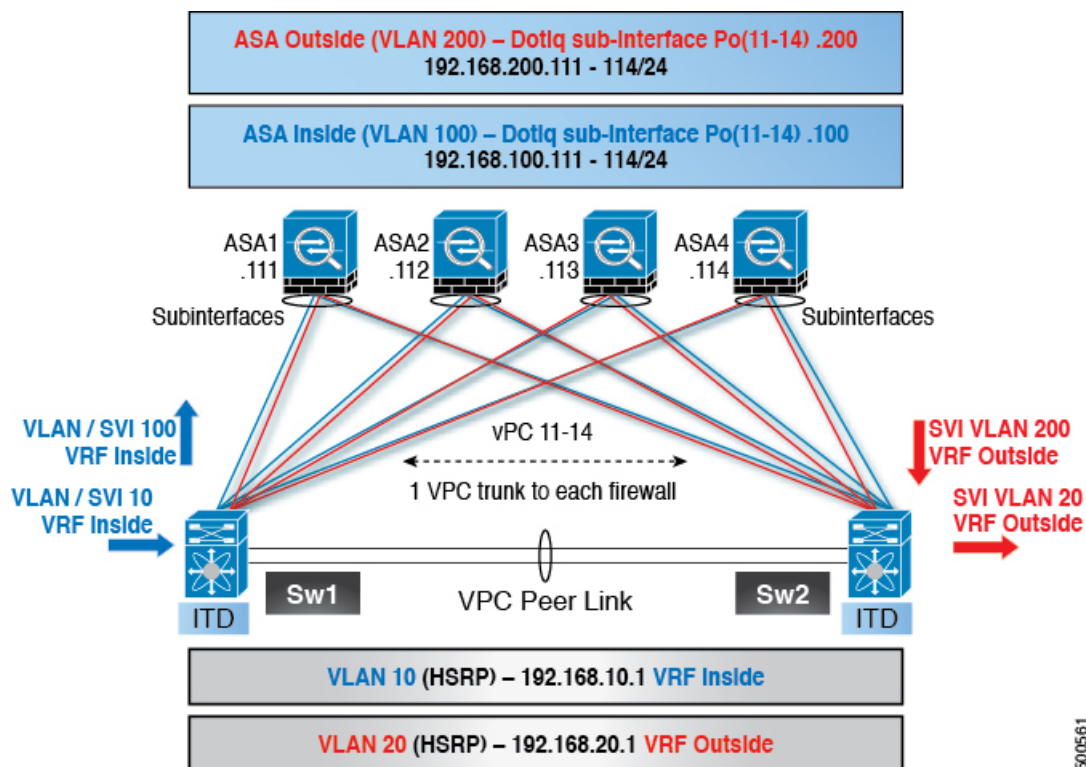


ITD ピア スイッチ ノード状態同期機能は、デュアル スイッチの非 vPC（またはシングル スイッチ）トポロジでのみサポートされます。ASA クラスタリングは、このような障害が発生した場合に ASA が完全に停止することを保証するため、この問題も解決します。ファイアウォール オン スティックの実装（シングル リンクまたは vPC）では、この問題に対処できません。これは、ASA の内部インターフェイスと外部インターフェイスが同じ物理（または仮想）インターフェイスに属しているためです。

設定例

スティック展開のファイアウォールでは、通常、vPC ポートチャネル（または単一ポート）トランクを使用して ASA をスイッチに接続します。この設定では、内部インターフェイスと外部インターフェイスは dot1q サブインターフェイス（VLAN 100 および 200）であり、スイッチには内部および外部コンテキストにそれぞれ 2 つの VLAN または SVI があり、それらの間で物理ポートが分離されていません。

図 22:スティック (vPC を使用) 展開のファイアウォール



500561

ステップ 1 : スイッチの構成

(注) この例は、スイッチ Sw1 の構成の一部を示しています。構成は、同様にすべての ASA に向けて適切に拡張する必要があります。他の機能は、すでに構成されていると想定されます。

```
interface vlan 10
  description Inside_Vlan_to_Network
  vrf member INSIDE
  ip address 192.168.10.10/24
  hsrp 10
  ip address 192.168.10.1

interface vlan 20
  description Outside_Vlan_to_Network
  vrf member OUTSIDE
  ip address 192.168.20.10/24
  hsrp 20
  ip address 192.168.20.1

interface vlan 100
  description Inside_Vlan_to_ASA
  vrf member INSIDE
  ip address 192.168.100.10/24
  hsrp 100
  ip address 192.168.100.1

interface vlan 200
```

```

description Outside_Vlan_to_ASA
vrf member OUTSIDE
ip address 192.168.200.10/24
hsrp 200
  ip address 192.168.200.1

interface port-channel 11
description VPC_TO_ASA1
switchport mode trunk
switchport trunk allowed vlan 100,200
vpc 11
no shutdown

interface ethernet 4/25
description Link_To_ITD-ASA-1
switchport
switchport mode trunk
switchport trunk allowed vlan 100,200
channel-group 11 mode active
no shutdown

interface port-channel 41
description Downstream_vPC_to_network
switchport mode trunk
switchport trunk allowed vlan 10,20
vpc 41
no shutdown

interface ethernet 5/1-4
description Downstream_vPC_member
switchport
switchport mode trunk
switchport trunk allowed vlan 10,20
channel-group 41
no shutdown

itd device-group FW_INSIDE
  #Config Firewall Inside interfaces as nodes
  node ip 192.168.100.111
  node ip 192.168.100.112
  node ip 192.168.100.113
  node ip 192.168.100.114
  probe icmp frequency 5 timeout 5 retry-count 1

itd device-group FW_OUTSIDE
  #Config Firewall Outside interfaces as nodes
  node ip 192.168.200.111
  node ip 192.168.200.112
  node ip 192.168.200.113
  node ip 192.168.200.114
  probe icmp frequency 5 timeout 5 retry-count 1

itd INSIDE
vrf INSIDE
  #applies ITD service to VRF 'INSIDE'
device-group FW_INSIDE
  #FW inside interfaces attached to service.
ingress interface vlan 10
  #applies ITD route map to vlan 1101 interface
failaction node reassign
  #To use the next available Active FW if an FW goes offline
load-balance method src ip buckets 16
  #distributes traffic into 16 buckets

```

```
        #load balances traffic based on Source IP.
        #OUTSIDE service uses Dest IP.
no shut

itd OUTSIDE
vrf OUTSIDE
    #applies ITD service to VRF 'OUTSIDE'
device-group FW_OUTSIDE
ingress interface vlan 20
failaction node reassign
load-balance method dst ip buckets 16
    #load balances traffic based on Dest IP.
    #INSIDE service uses Src IP.
no shut
```

ステップ 2 : ASA の構成。

```
interface port-channel 11
    nameif aggregate
    security-level 100
    no ip address

interface port-channel 11.100
    description INSIDE
    vlan 100
    nameif inside
    security-level 100
    ip address 192.168.100.111 255.255.255.0

interface port-channel 11.200
    description OUTSIDE
    vlan 200
    nameif outside
    security-level 100
    ip address 192.168.200.111 255.255.255.0

same-security-traffic permit inter-interface

interface TenGigabitEthernet 0/6
    description CONNECTED_TO_SWITCH-A-VPC
    channel-group 11 mode active
    no nameif
    no security-level

interface TenGigabitEthernet 0/7
    description CONNECTED_TO_SWITCH-B-VPC
    channel-group 11 mode active
    no nameif
    no security-level
```

このトポロジ例には、次の点が当てはまります。

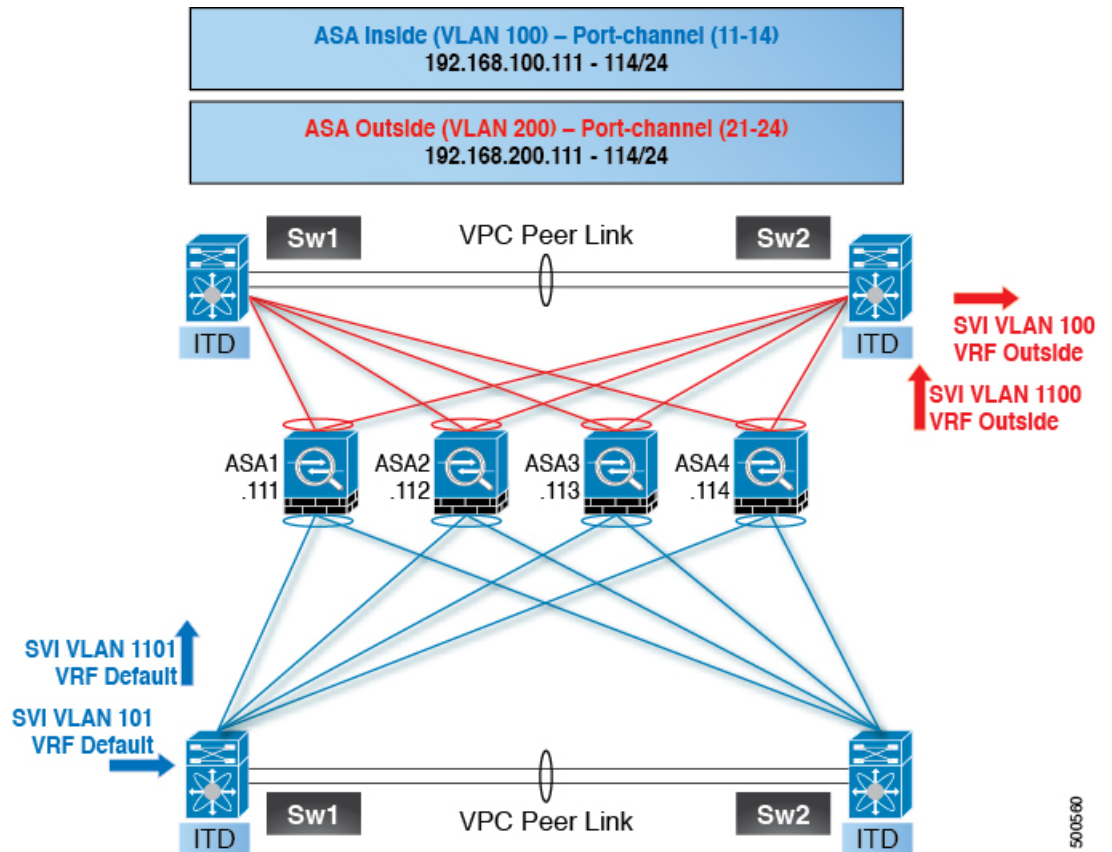
- VLAN 10、20、100、および 200 とそれらの SVI は、適切な VRF にマッピングされます。
- この例では、ITD ロードバランシング設定を使用してフローの対称性を実現しています。
- vPC シナリオでは、vPC の 1 つのメンバーが稼働している限り、ITD に変更はありません。vPC レッグに障害が発生したスイッチの ITD リダイレクションは、通常の vPC 配置の場合と同様に、ピアリンクを介してピアスイッチを通過します。

- このトポロジでは、内部インターフェイスと外部インターフェイスが ASA の同じ物理インターフェイスまたは仮想インターフェイス（dot1q サブインターフェイス）に結び付けられているため、物理リンクの障害時にトラフィックが失われることはありません。
- vPC 上のルーティングプロトコルネイバーをサポートするには、`layer3 peer-router` コマンドを vPC ドメイン内で構成する必要があります。
- レイヤ3インターフェイスはファイアウォールの内側と外側の両方のインターフェイスに接続するために使用されるため、VRF が必要です。VRF は、特定の場合にトラフィックがファイアウォールを迂回して（VLAN間）ルーティングされるのを防ぐために配置されます。
- トラフィックはポリシーベースルーティングを使用して ASA に向けられるため、ルートは必要ありません。

構成例：vPC を使用したデュアルスイッチ サンドイッチ モードのファイアウォール

vPC を使用したサンドイッチ モードの場合、内部および外部 ASA インターフェイスはそれぞれ別のポート チャネルバンドルに割り当てられます。vPC の結果として、単一のリンク障害がトラフィック フローを妨げることはなく、ITD は引き続きピア スイッチのリンクを介して ASA に転送します。

図 23: vPC を使用したデュアルスイッチ サンドイッチ モード



ステップ 1 : 2つのスイッチを構成します。

```
switch #1:
interface vlan 10
  description INSIDE_VLAN
  ip address 192.168.10.10/24

interface vlan 100
  description FW_INSIDE_VLAN
  ip address 192.168.100.10/24

interface port-channel 11
  description To_ASA-1_INSIDE
  switchport mode access
  switchport access vlan 100
  vpc 11

interface ethernet 4/1
  description To_ASA-1_INSIDE
  switchport mode access
  switchport access vlan 100
  channel-group 11 mode active

switch #2:
interface vlan 20
  description OUTSIDE_VLAN
  ip address 192.168.20.10/24
```

```
interface vlan 200
  description FW_OUTSIDE_VLAN
  ip address 192.168.200.10/24

interface port-channel 21
  description To_ASA-1_OUTSIDE
  switchport mode access
  switchport access vlan 200
  vpc 11

interface ethernet 4/25
  description To_ASA-1_OUTSIDE
  switchport mode access
  switchport access vlan 200
  channel-group 21 mode active
```

ステップ 2 : ASA の構成。

```
interface port-channel 11
  description INSIDE
  vlan 100
  nameif inside
  security-level 100
  ip address 192.168.100.111 255.255.255.0

interface port-channel 21
  description OUTSIDE
  vlan 100
  nameif outside
  security-level 100
  ip address 192.168.200.111 255.255.255.0

same-security-traffic permit inter-interface

interface TenGigabitEthernet 0/6
  description CONNECTED_TO_SWITCH-A-VPC
  channel-group 11 mode active
  no nameif
  no security-level

interface TenGigabitEthernet 0/7
  description CONNECTED_TO_SWITCH-B-VPC
  channel-group 11 mode active
  no nameif
  no security-level

interface TenGigabitEthernet 0/8
  description CONNECTED_TO_SWITCH-A-VPC
  channel-group 21 mode active
  no nameif
  no security-level

interface TenGigabitEthernet 0/9
  description CONNECTED_TO_SWITCH-B-VPC
  channel-group 21 mode active
  no nameif
  no security-level
```

このトポロジ例には、次の点が当てはまります。

- この例では、ITD ロードバランシング設定を使用してフローの対称性を実現しています。

- vPC シナリオでは、vPC の 1 つのメンバーが稼働している限り、ITD に変更はありません。vPC レッグに障害が発生したスイッチの ITD リダイレクションは、通常の vPC 配置の場合と同様に、ピア リンクを介してピア スイッチを通過します。
- このトポロジでは、ASA のポート チャネルの 1 つ（または非 vPC トポロジの単一の物理リンク）に障害が発生すると、トラフィック損失が発生する可能性があります。
- vPC 上のルーティングプロトコル ネイバーをサポートするには、layer3 peer-router コマンドを vPC ドメイン内で構成する必要があります。
- トラフィックはポリシーベース ルーティングを使用して ASA に向けられるため、ルートは必要ありません。

構成例：レイヤ3クラスタリングのファイアウォール

ASA クラスタは、1 つのユニットとして機能する複数の ASA から構成されます。複数の ASA を単一論理デバイスとしてグループ化することで、単一デバイスのすべての利便性（管理、ネットワークへの統合）を提供し、同時に複数デバイスによる高いスループットと冗長性を実現できます。

ITD は、個々のモードのレイヤ3 ASA クラスタにロードバランシングできます。ITD はクラスタリングを補完するものであり、ITD は各ファイアウォールによってどのフローが処理されるかを予測できるようにします。OSPF ECMP およびポートチャネルハッシュアルゴリズムに依存する代わりに、ITD バケットを使用してこれらのフローを決定できます。

レイヤ3 クラスタでは、バケット割り当てに基づいてフローの所有者を事前に決定できます。ITD およびレイヤ3 クラスタリングがない場合、所有者の最初の選択は通常、予測できません。ITD では、所有者を事前に決定できます。

ASA クラスタリングでは、バックアップフローの所有者も使用します。クラスタ内の特定のファイアウォールを通過するすべてのフローについて、別のファイアウォールがそのフローの状態と、フローを所有する ASA を保存します。実際のアクティブなフローの所有者が失敗した場合、ITD failaction の再割り当てにより、失敗した所有者の ASA からのすべてのフロー（バケット）が、デバイスグループにリストされている次のアクティブノードに移動します。このトラフィックを受信する新しいファイアウォールが、受信するフローのバックアップの所有者でない場合、バックアップの所有者からフロー状態情報を受信し、トラフィックをシームレスに処理する必要があります。

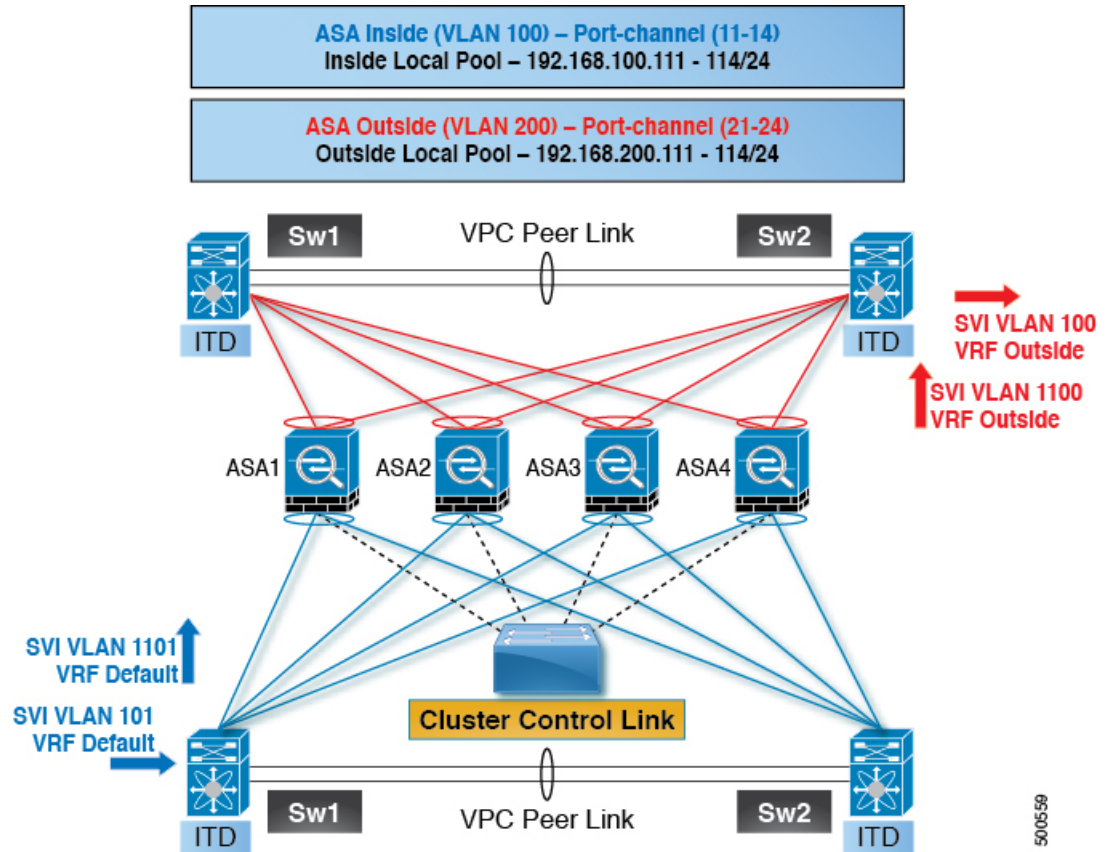
ITD で ASA クラスタリングを使用する場合の潜在的な欠点は、バックアップフローおよびその他のクラスタテーブル操作が、非クラスタ化ファイアウォールでは消費されないメモリと CPU リソースを消費することです。したがって、非クラスター化ファイアウォールを使用すると、ファイアウォールのパフォーマンスが向上する場合があります。

次の表は、ASA デバイスのステータスが変化したときに、ECMP と ITD で発生するクラスタ制御リンク（CCL）への影響の概要を比較したものです。

表 4: ECMP と ITD - CCL の影響の概要の比較

| ASA ステータス | ITD | ECMP |
|----------------|--|--|
| 定常状態 | <p>CCL 上の最小限のトラフィックと予想されるトラフィックタイプ。</p> <p>ラインカードとスイッチのタイプに関係なく、まったく同じ負荷分散。</p> | <p>同じラインカードタイプとスイッチモデルがすべての場所で使用されている場合、CCL 上の最小限のトラフィック。</p> <p>異なるハードウェアが使用されている場合、より高いレベルの非対称性が発生し、CCL ネットワークでトラフィックが発生する可能性があります。ハードウェアごとに異なるハッシュ関数があります。</p> <p>2つのスイッチ（たとえば、vPC 内）が同じフローを異なる ASA デバイスに送信し、CCL トラフィックが発生する可能性があります。</p> |
| 1つの ASA で障害が発生 | <p>CCL に追加のトラフィックはありません。</p> <p>ITD は、IP ステイッキ性と復元力のあるハッシュを提供します。</p> | <p>すべてのフローが再ハッシュされ、追加のトラフィックリダイレクションが CCL で発生します。クラスタ内のすべての ASA デバイスへのトラフィックが影響を受ける可能性があります。</p> |
| 単一 ASA のリカバリ | <p>トラフィックリダイレクションは、クラスタ内の2つの ASA デバイス間で CCL で発生する可能性があります。つまり、パケットを受信する回復された ASA と、以前にそのパケットにサービスを提供していた ASA です。</p> | <p>追加のトラフィックリダイレクションは、CCL で発生する可能性があります。クラスタ内のすべての ASA デバイスへのトラフィックが影響を受ける可能性があります。</p> |
| ASA 追加 | <p>CCL の最小限の追加トラフィック。</p> | <p>すべてのフローが再ハッシュされ、追加のトラフィックリダイレクションが CCL で発生します。クラスタ内のすべての ASA デバイスへのトラフィックが影響を受ける可能性があります。</p> |

図 24: vPC を使用したデュアルスイッチ サンドイッチを備えた ASA クラスタ



ステップ 1 : 2つのスイッチを構成します。



- (注) クラスタリングを導入しても、ITD 構成は変更されません。ITD の設定は、トポロジのタイプによって異なります。この例では、設定は vPC トポロジを使用したデュアルスイッチ サンドイッチと同じです。

```
switch #1:
interface vlan 10
description INSIDE_VLAN
ip address 192.168.10.10/24

interface vlan 100
description FW_INSIDE_VLAN
ip address 192.168.100.10/24

interface port-channel 11
description To_ASA-1_INSIDE
switchport mode access
switchport access vlan 100
vpc 11

interface ethernet 4/1
description To_ASA-1_INSIDE
```

```

switchport mode access
switchport access vlan 100
channel-group 11 mode active

switch #2:
interface vlan 20
  description OUTSIDE_VLAN
  ip address 192.168.20.10/24

interface vlan 200
  description FW_OUTSIDE_VLAN
  ip address 192.168.200.10/24

interface port-channel 21
  description To_ASA-1_OUTSIDE
  switchport mode access
  switchport access vlan 200
  vpc 11

interface ethernet 4/25
  description To_ASA-1_OUTSIDE
  switchport mode access
  switchport access vlan 200
  channel-group 21 mode active

```

ステップ2：ASAを構成します。

```

cluster group ASA-CLUSTER-L3
  local-unit ASA1
  cluster-interface port-channel 31
  ip address 192.168.250.100 255.255.255.0
  priority 1
  health-check holdtime 1.5
  clacp system-mac auto system-priority 1
  enable

mac-address pool MAC-INSIDE aaaa.0101.0001 - aaaa.0101.0008
mac-address pool MAC-OUTSIDE aaaa.0100.0001 - aaaa.0100.0008
ip local pool IP-OUTSIDE 192.168.200.111-192.168.200.114
ip local pool IP-INSIDE 192.168.100.111-192.168.100.114

interface port-channel 11
  description INSIDE
  lacp max-bundle 8
  mac-address cluster-pool MAC-INSIDE
  nameif inside
  security-level 100
  ip address 192.168.100.11 255.255.255.0 cluster-pool IP-INSIDE

interface port-channel 21
  description OUTSIDE
  lacp max-bundle 8
  mac-address cluster-pool MAC-OUTSIDE
  nameif outside
  security-level 100
  ip address 192.168.200.11 255.255.255.0 cluster-pool IP-OUTSIDE

interface port-channel 31
  description Clustering Interface
  lacp max-bundle 8

interface TenGigabitEthernet 0/6
  channel-group 11 mode active

```

```
no nameif
no security-level
no ip address

interface TenGigabitEthernet 0/7
channel-group 11 mode active
no nameif
no security-level
no ip address

interface TenGigabitEthernet 0/8
channel-group 21 mode active
no nameif
no security-level
no ip address

interface TenGigabitEthernet 0/9
channel-group 21 mode active
no nameif
no security-level
no ip address

interface TenGigabitEthernet 1/0
channel-group 31 mode on
no nameif
no security-level
no ip address

interface TenGigabitEthernet 1/1
channel-group 31 mode on
no nameif
no security-level
no ip address
```

この例では、ポートチャネル 11 および 21 が内部インターフェイスと外部インターフェイスに使用されています。ポートチャネル 31 はクラスタリングインターフェイスです。個別インターフェイスは通常のルーテッドインターフェイスであり、それぞれが専用の IP アドレスを IP アドレスプールから取得します。メインクラスタ IP アドレスは、そのクラスタのための固定アドレスであり、常に現在のプライマリユニットに属します。同様に、MAC アドレスプールも構成され、対応する内部または外部ポートチャネルの下で使用されます。

ITD レイヤ 2 の構成例

次の例は、ITD-L2 を構成する方法を示しています。

ITD レイヤ 2 機能を有効にします。

```
(config) feature itd
(config) itd Port-group 100
(config-port-group) int eth 1/11
(config-port-group) int eth 1/12
(config) itd SER3
(config-itd) port-group 100
(config-itd) source vlan 2010-2015
(config-itd) no shutdown
```

ITD-L2 構成を確認します。

```
s!Command: show running-config services
!Running configuration last done at: Thu Dec  5 00:04:35 2019
!Time: Thu Dec  5 20:44:06 2019

version 9.3(3u)I9(1u) Bios:version 08.36
feature itd

itd port-group PG100
  interface Eth1/11
  interface Eth1/12
  interface Eth1/13
  interface Eth1/14
  interface Eth1/15
  interface Eth1/16
  interface Eth1/17
  interface Eth1/18
  interface Eth1/19
  interface Eth1/20
  interface Eth1/21
  interface Eth1/22
  interface Eth1/23

itd SER1
  port-group PG100
  source vlan 10-15
  no shut

itd SER2
  port-group PG100
  source vlan 1010-1015
  no shut
```

関連資料

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。