

プライベートクラウドの ASA Virtual のクラスタを展開する

最終更新：2021年12月1日

プライベートクラウド用のASA 仮想のASA 仮想のクラスタを展開する

クラスタリングを利用すると、複数の ASA 仮想をグループ化して1つの論理デバイスとすることができます。クラスタは、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）を備える一方で、複数デバイスによって高いスループットおよび冗長性を達成します。VMware と KVM を使用して ASA 仮想クラスタを導入できます。ルーテッドファイアウォールモードのみがサポートされます。



(注) クラスタリングを使用する場合、一部の機能はサポートされません。[クラスタリングでサポートされない機能 \(56 ページ\)](#) を参照してください。

ASA 仮想クラスタリングについて

ここでは、クラスタリングアーキテクチャとその動作について説明します。

クラスタをネットワークに適合させる方法

クラスタは、複数のファイアウォールで構成され、これらは1つのデバイスとして機能します。ファイアウォールをクラスタとして機能させるには、次のインフラストラクチャが必要です。

- クラスタ内通信用の、隔離されたネットワーク。VXLAN インターフェイスを使用したクラスタ制御リンクと呼ばれます。レイヤ3物理ネットワーク上でレイヤ2仮想ネットワークとして機能する VXLAN により、ASA Virtual はクラスタ制御リンクを介してブロードキャスト/マルチキャストメッセージを送信できます。
- 各ファイアウォールへの管理アクセス（コンフィギュレーションおよびモニタリングのため）。ASA Virtual 導入には、クラスタノードの管理に使用する Management 0/0 インターフェイスが含まれています。

クラスタをネットワーク内に配置するときは、アップストリームおよびダウンストリームのルータは、レイヤ3の個別インターフェイスおよび次のいずれかの方法を使用して、クラスタとの間で送受信されるデータをロードバランシングできる必要があります。

- ポリシーベースルーティング：アップストリームとダウンストリームのルータが、ルートマップと ACL を使用してノード間のロードバランシングを実行します。
- 等コスト マルチパス ルーティング：アップストリームとダウンストリームのルータが、等コストのスタティックまたはダイナミックルートを使用してノード間のロードバランシングを実行します。



(注) レイヤ 2 スパンド EtherChannels はサポートされません。

クラスタ ノード

クラスタノードは連携して動作し、セキュリティポリシーおよびトラフィックフローの共有を達成します。ここでは、各ノードのロールの特長について説明します。

ブートストラップ コンフィギュレーション

各デバイスで、最小限のブートストラップ コンフィギュレーション（クラスタ名、クラスタ制御リンクインターフェイスなどのクラスタ設定）を設定します。通常、クラスタリングを有効にする最初のノードが制御ノードになります。以降のノードに対してクラスタリングをイネーブルにすると、そのノードはデータノードとしてクラスタに参加します。

制御ノードとデータノードの役割

クラスタ内のメンバーの1つが制御ノードになります。複数のクラスタノードが同時にオンラインになる場合、制御ノードは、ブートストラップ コンフィギュレーション内のプライオリティ設定によって決まります。プライオリティは1～100の範囲内で設定され、1が最高のプライオリティです。他のすべてのメンバーはデータノードです。一般的には、クラスタを作成した後で最初に追加したノードが制御ノードとなります。これは単に、その時点でクラスタに存在する唯一のノードであるからです。

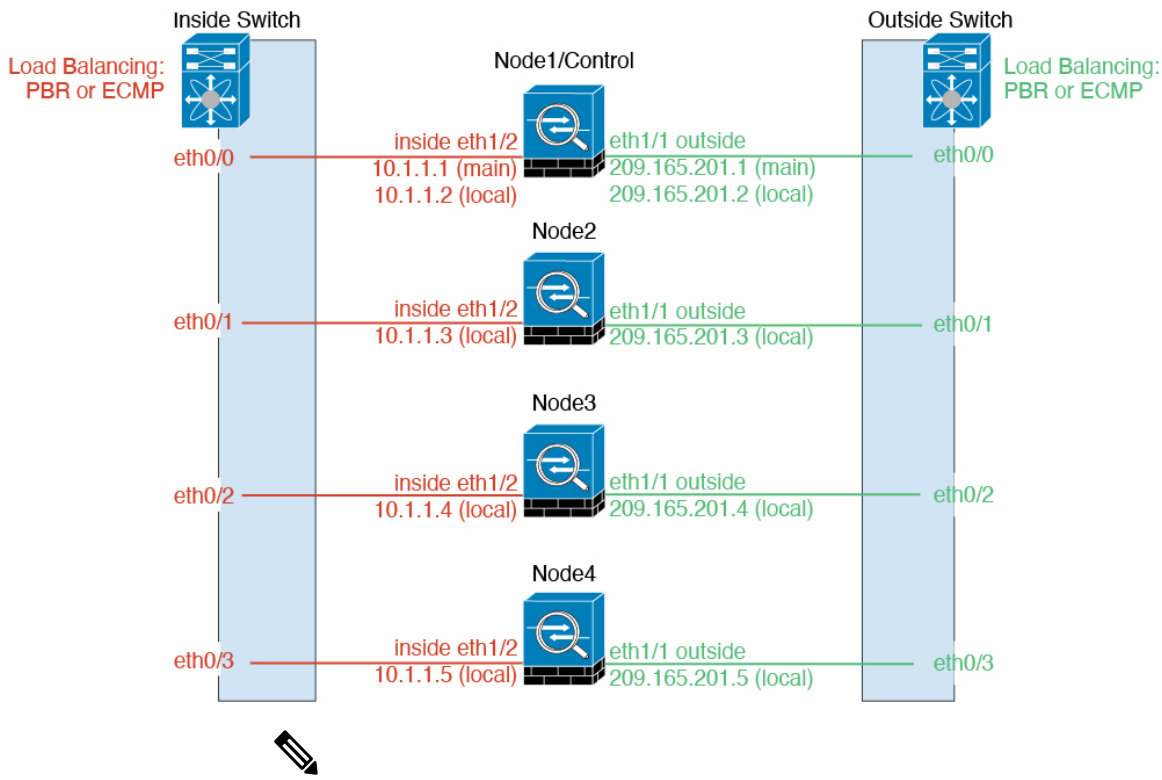
すべてのコンフィギュレーション作業（ブートストラップ コンフィギュレーションを除く）は、制御ノード上のみで実行する必要があります。コンフィギュレーションは、データノードに複製されます。物理的アセット（たとえばインターフェイス）の場合は、制御ノードのコンフィギュレーションがすべてのデータノード上でミラーリングされます。たとえば、内部インターフェイスとしてイーサネット1/2を設定し、外部インターフェイスとしてイーサネット1/1を設定した場合、これらのインターフェイスは内部および外部インターフェイスとしてデータノードでも使用されます。

機能によっては、クラスタ内でスケーリングしないものがあり、そのような機能については制御ノードがすべてのトラフィックを処理します。

個々のインターフェイス

クラスターフェイスを個々のインターフェイスとして設定できます。

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、それぞれが専用のローカル IP アドレスを持ちます。インターフェイス コンフィギュレーションは制御ノード上だけで行う必要があるため、このインターフェイス コンフィギュレーションの中で IP アドレスプールを設定して、このプールのアドレスをクラスタノード（制御ノード用を含む）のインターフェイスに使用させることができます。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、常に現在の制御ノードに属します。ローカル IP アドレスは、常にルーティングの制御ノードアドレスです。このメインクラスタ IP アドレスによって、管理アクセスのアドレスが一本化されます。制御ノードが変更されると、メインクラスタ IP アドレスは新しい制御ノードに移動するので、クラスタの管理をシームレスに続行できます。ただし、ロードバランシングを別途する必要があります（この場合はアップストリームスイッチ上）。



(注) レイヤ 2 スパンド EtherChannels はサポートされません。

ポリシーベース ルーティング（ルーテッド ファイアウォール モードのみ）

個別インターフェイスを使用するときは、各 ASA インターフェイスが専用の IP アドレスと MAC アドレスを維持します。ロードバランシング方法の 1 つが、ポリシーベースルーティング（PBR）です。

この方法が推奨されるのは、すでに PBR を使用しており、既存のインフラストラクチャを活用したい場合です。

PBR は、ルート マップおよび ACL に基づいて、ルーティングの決定を行います。管理者は、手動でトラフィックをクラスタ内のすべての ASA に分ける必要があります。PBR は静的であるため、常に最適なロードバランシング結果を実現できないこともあります。最高のパフォーマンスを達成するには、PBR ポリシーを設定するときに、同じ接続のフォワードとリターンのパケットが同じ ASA に送信されるように指定することを推奨します。たとえば、Cisco ルータがある場合は、冗長性を実現するには Cisco IOS PBR をオブジェクトトラッキングとともに使用します。Cisco IOS オブジェクトトラッキングは、ICMP ping を使用して各 ASA をモニタします。これで、PBR は、特定の ASA の到達可能性に基づいてルートマップをイネーブルまたはディセーブルにできます。詳細については、次の URL を参照してください。

<http://www.cisco.com/c/en/us/solutions/data-center-virtualization/intelligent-traffic-director/index.html>

http://www.cisco.com/en/US/products/ps6599/products_white_paper09186a00800a4409.shtml

等コスト マルチパス ルーティング (ルーテッド ファイアウォール モードのみ)

個別インターフェイスを使用するときは、各 ASA インターフェイスが専用の IP アドレスと MAC アドレスを維持します。ロードバランシング方法の1つが、等コストマルチパス (ECMP) ルーティングです。

この方法が推奨されるのは、すでに ECMP を使用しており、既存のインフラストラクチャを活用したい場合です。

ECMP ルーティングでは、ルーティングメトリックが同値で最高である複数の「最適パス」を介してパケットを転送できます。EtherChannel のように、送信元および宛先の IP アドレスや送信元および宛先のポートのハッシュを使用してネクストホップの1つにパケットを送信できます。ECMP ルーティングにスタティックルートを使用する場合は、ASA の障害発生時に問題が起きることがあります。ルートは引き続き使用されるため、障害が発生した ASA へのトラフィックが失われるからです。スタティック ルートを使用する場合は必ず、オブジェクトトラッキングなどのスタティック ルート モニタリング機能を使用してください。ダイナミックルーティングプロトコルを使用してルートの追加と削除を行うことを推奨します。この場合は、ダイナミック ルーティングに参加するように各 ASA を設定する必要があります。

クラスタ制御リンク

ノードごとに1つのインターフェイスをクラスタ制御リンク専用の VXLAN (VTEP) インターフェイスにする必要があります。

VXLAN トンネル エンドポイント

VXLAN トンネルエンドポイント (VTEP) デバイスは、VXLAN のカプセル化およびカプセル化解除を実行します。各 VTEP には2つのインターフェイスタイプ (VXLAN Network Identifier (VNI) インターフェイスと呼ばれる1つ以上の仮想インターフェイスと、VTEP 間に VNI をトンネリングする VTEP 送信元インターフェイスと呼ばれる通常のインターフェイス) があります。VTEP 送信元インターフェイスは、VTEP 間通信のトランスポート IP ネットワークに接続されます。

VTEP 送信元インターフェイス

VTEP 送信元インターフェイスは、VNI インターフェイスに関連付けられる予定の標準の ASA Virtual インターフェイスです。1 つの VTEP ソースインターフェイスをクラスタ制御リンクとして機能するように設定できます。ソースインターフェイスは、クラスタ制御リンクの使用専用に予約されています。各 VTEP ソースインターフェイスには、同じサブネット上の IP アドレスがあります。このサブネットは、他のすべてのトラフィックからは隔離し、クラスタ制御リンクインターフェイスだけが含まれるようにしてください。

VNI インターフェイス

VNI インターフェイスは VLAN インターフェイスに似ています。VNI インターフェイスは、タギングを使用して特定の物理インターフェイスでのネットワークトラフィックの分割を維持する仮想インターフェイスです。設定できる VNI インターフェイスは 1 つだけです。各 VNI インターフェイスは、同じサブネット上の IP アドレスを持ちます。

ピア VTEP

単一の VTEP ピアを許可するデータインターフェイス用の通常の VXLAN とは異なり、ASA Virtual クラスタリングでは複数のピアを設定できます。

クラスタ制御リンク トラフィックの概要

クラスタ制御リンク トラフィックには、制御とデータの両方のトラフィックが含まれます。

制御トラフィックには次のものが含まれます。

- 制御ノードの選択。
- 設定の複製。
- ヘルス モニタリング。

データ トラフィックには次のものが含まれます。

- 状態の複製。
- 接続所有権クエリおよびデータ パケット転送。

クラスタ制御リンクの障害

ユニットのクラスタ制御リンク回線プロトコルがダウンした場合、クラスタリングはディセーブルになります。データ インターフェイスはシャットダウンされます。クラスタ制御リンクの修復後、クラスタリングを再度イネーブルにして手動でクラスタに再参加する必要があります。



- (注) ASA 仮想 が非アクティブになると、すべてのデータ インターフェイスがシャットダウンされます。管理専用インターフェイスのみがトラフィックを送受信できます。管理インターフェイスは、そのユニットが DHCP またはクラスタ IP プールから受け取った IP アドレスを使用して引き続き稼働状態となります。クラスタ IP プールを使用している場合、リロードしてもクラスタでユニットがまだ非アクティブになっていると、管理インターフェイスはアクセスできません（制御ノードと同じメイン IP アドレスを使用するため）。さらに設定を行う場合は、コンソールポート（使用可能な場合）を使用する必要があります。

コンフィギュレーションの複製

クラスタ内のすべてのノードは、単一の設定を共有します。設定の変更は制御ノードでのみ可能（ブートストラップ設定は除く）で、変更はクラスタに含まれる他のすべてのノードに自動的に同期されます。

ASA 仮想 クラスタの管理

ASA 仮想 クラスタリングを使用することの利点の1つは、管理のしやすさです。ここでは、クラスタを管理する方法について説明します。

管理ネットワーク

すべてのノードを単一の管理ネットワークに接続することを推奨します。このネットワークは、クラスタ制御リンクとは別のものです。

管理インターフェイス

管理用に、管理 0/0 インターフェイスを使用します。



- (注) 管理インターフェイスの動的ルーティングを有効にすることはできません。スタティックルートを使用する必要があります。

管理 IP アドレスには、静的アドレスまたは DHCP を使用できます。

静的 IP アドレスを使用する場合は、常に現在の制御ノードに属するクラスタの固定アドレスであるメインクラスタ IP アドレスを使用できます。インターフェイスごとに、管理者はアドレス範囲も設定します。これで、各ノード（現在の制御ノードも含まれます）がその範囲内のローカルアドレスを使用できるようになります。このメインクラスタ IP アドレスによって、管理アクセスのアドレスが一本化されます。制御ノードが変更されると、メインクラスタ IP アドレスは新しい制御ノードに移動するので、クラスタの管理をシームレスに続行できます。ローカル IP アドレスは、ルーティングに使用され、トラブルシューティングにも役立ちます。たとえば、クラスタを管理するにはメインクラスタ IP アドレスに接続します。このアドレスは常に、現在の制御ノードに関連付けられています。個々のメンバを管理するには、ローカル

IP アドレスに接続します。TFTP や syslog などの発信管理トラフィックの場合、制御ノードを含む各ノードは、ローカル IP アドレスを使用してサーバーに接続します。

DHCP を使用する場合、ローカルアドレスのプールを使用したり、メインクラスタの IP アドレスを使用したりしません。

制御ノードの管理対データノードの管理

すべての管理とモニタリングは制御ノードで実行できます。制御ノードから、すべてのノードのランタイム統計情報、リソース使用状況、その他のモニタリング情報を確認できます。また、クラスタ内のすべてのノードに対してコマンドを発行したり、コンソールメッセージをデータノードから制御ノードに複製したりできます。

必要に応じて、データノードを直接モニタできます。制御ノードからも可能ですが、ファイル管理（設定のバックアップやイメージの更新など）をデータノード上で実行できます。次の機能は、制御ノードからは使用できません。

- ノードごとのクラスタ固有統計情報のモニタリング。
- ノードごとの Syslog モニタリング（コンソールレプリケーションが有効な場合にコンソールに送信される Syslog を除く）。
- SNMP
- NetFlow

暗号キー複製

制御ノード上で暗号キーを作成すると、そのキーはすべてのデータノードに複製されます。メインクラスタ IP アドレスへの SSH セッションがある場合、制御ノードで障害が発生すると接続が切断されます。新しい制御ノードでは、SSH 接続に対して同じキーが使用されるため、新しい制御ノードに再接続するときに、キャッシュ済みの SSH ホストキーを更新する必要はありません。

ASDM 接続証明書 IP アドレス不一致

デフォルトでは、自己署名証明書は、ローカル IP アドレスに基づいて ASDM 接続に使用されます。ASDM を使用してメインクラスタ IP アドレスに接続すると、IP アドレス不一致に関する警告メッセージが表示される場合があります。これは、証明書で使用されているのがローカル IP アドレスであり、メインクラスタ IP アドレスではないためです。このメッセージは無視して、ASDM 接続を確立できます。ただし、この種の警告を回避するには、新しい証明書を登録し、この中でメインクラスタ IP アドレスと、IP アドレスプールからのすべてのローカル IP アドレスを指定します。この証明書を各クラスタメンバに使用します。詳細については、

「<https://www.cisco.com/c/en/us/td/docs/security/asdm/identity-cert/cert-install.html>」を参照してください。

サイト間クラスタリング

サイト間インストールの場合、次の推奨ガイドラインに従う限り、ASA 仮想クラスタリングを利用できます。

各クラスタ シャーシを、個別のサイト ID に属するように設定できます。サイト ID は、LISP インスペクションを使用するフローモビリティ、データセンターのサイト間クラスタリングのパフォーマンスを向上し、ラウンドトリップ時間の遅延を減少させるためのディレクタ ローカリゼーション、およびトラフィックフローのバックアップオーナーが常にオーナーとは異なるサイトにある接続のサイト冗長性を有効にするために使用されます。

サイト間クラスタリングの詳細については、以下の項を参照してください。

- Data Center Interconnect のサイジング : [ASA 仮想クラスタリングの要件と前提条件 \(8 ページ\)](#)
- サイト間のガイドライン : [ASA 仮想クラスタリングに関するガイドライン \(9 ページ\)](#)
- クラスタ フロー モビリティの設定 : [クラスタ フロー モビリティの設定 \(34 ページ\)](#)
- ディレクタ ローカリゼーションの有効化 : [ディレクタ ローカリゼーションの有効化 \(32 ページ\)](#)
- サイト冗長性の有効化 : [ディレクタ ローカリゼーションの有効化 \(32 ページ\)](#)
- サイト間での例 : [個別インターフェイス ルーテッド モード ノースサウス サイト間の例 \(55 ページ\)](#)

ASA 仮想クラスタリングのライセンス

各クラスタノードには、同じモデルライセンスが必要です。すべてのノードに同じ数の CPU とメモリを使用することをお勧めします。そうしないと、パフォーマンスが最小能力のメンバーに一致するようにすべてのノードで制限されます。スループットレベルは、一致するように制御ノードから各データノードに複製されます。



- (注) ASA 仮想 を登録解除してライセンスを解除した場合、ASA 仮想 をリロードすると、重大なレート制限状態に戻ります。ライセンスのない、パフォーマンスの低いクラスタノードは、クラスタ全体のパフォーマンスに悪影響を及ぼします。すべてのクラスタノードのライセンスを保持するか、ライセンスのないノードを削除してください。

ASA 仮想クラスタリングの要件と前提条件

モデルの要件

- ASAv30、ASAv50、ASAv100
- VMware または KVM
- 最大 16 ノード

ASA 仮想プラットフォームおよびソフトウェア要件

クラスタ内のすべてのノード：

- 同じモデルである必要があります。すべてのノードに同じ数の CPU とメモリを使用することをお勧めします。そうしないと、パフォーマンスが最小能力のノードに一致するようにすべてのノードで制限されます。
- イメージアップグレード時を除き、同じソフトウェアを実行する必要があります。ヒットレス アップグレードがサポートされます。
- コンフィギュレーション複製前の初期クラスタ制御リンク通信のために、新しいクラスタメンバーは、制御ノードと同じ SSL 暗号化設定 (**ssl encryption** コマンド) を使用する必要があります。

ASA 仮想クラスタリングに関するガイドライン

フェールオーバー

フェールオーバーは、クラスタリングではサポートされません。

IPv6

クラスタ制御リンクは、IPv4 のみを使用してサポートされます。

その他のガイドライン

- 大々的なトポロジ変更が発生する場合（ASA 上でのインターフェイスまたはスイッチの有効化または無効化、VSS または vPC を形成するための追加スイッチの追加など）、ヘルスチェック機能を無効にし、無効化したインターフェイスのインターフェイスモニタリングも無効にする必要があります。トポロジの変更が完了して、設定の変更がすべてのノードに同期されたら、インターフェイスヘルスチェック機能を再度有効にできます。
- ノードを既存のクラスタに追加したときや、ノードをリロードしたときは、一時的に、限定的なパケット/接続ドロップが発生します。これは予定どおりの動作です。場合によっては、ドロップされたパケットが原因で接続がハングすることがあります。たとえば、FTP 接続の FIN/ACK パケットがドロップされると、FTP クライアントがハングします。この場合は、FTP 接続を再確立する必要があります。
- データインターフェイスの VXLAN はサポートしていません。クラスタ制御リンクのみが VXLAN をサポートします。
- クラスタ内のすべてのノードに変更が複製されるまでには時間がかかります。たとえば、オブジェクトグループを使用するアクセスコントロールルール（展開時に複数のルールに分割される）を追加するなどの大きな変更を行うと、変更の完了に必要な時間がクラスタノードが成功メッセージで応答できるタイムアウトを超える可能性があります。この場合、「failed to replicate command」というメッセージが表示されることがあります。このメッセージは無視できます。

ASA 仮想クラスタリングのデフォルト

- クラスタのヘルスチェック機能は、デフォルトで有効になり、ホールド時間は3秒です。デフォルトでは、すべてのインターフェイスでインターネットヘルスマonitoringが有効になっています。
- 失敗したクラスタ制御リンクのクラスタ再結合機能が5分おきに無制限に試行されます。
- 失敗したデータインターフェイスのクラスタ自動再結合機能は、5分後と、2に設定された増加間隔で合計で3回試行されます。
- 接続再分散は、デフォルトでは無効になっています。接続再分散を有効にした場合の、デフォルトの負荷情報交換間隔は5秒です。
- HTTPトラフィックでは、5秒間の接続複製遅延がデフォルトで有効になっています。

Day0 設定を使用した ASA 仮想 クラスタリングの設定

制御ノード Day0 設定

制御ノードの次の Day0 設定には、ブートストラップ設定と、それに続くデータノードに複製されるインターフェイス設定が含まれています。太字のテキストは、データノードの Day0 設定で変更する必要がある値を示しています。



- (注) この設定には、クラスタ中心の設定のみが含まれます。Day0 設定には、ライセンス、SSH アクセス、ASDM アクセスなどの他の設定も含める必要があります。Day0 設定の詳細については、スタートアップガイドを参照してください。

```
!BOOTSTRAP
! Cluster interface mode
cluster interface mode individual
!
! VXLAN peer group
object-group network cluster-peers
network-object host 10.6.6.51
network-object host 10.6.6.52
network-object host 10.6.6.53
network-object host 10.6.6.54
!
! Alternate object group representation
! object-network xyz
! range 10.6.6.51 10.6.6.54
! object-group network cluster-peers
! network-object object xyz
!
! Cluster control link physical interface (VXLAN tunnel endpoint (VTEP) src interface)
interface gigabitethernet 0/7
description CCL VTEP src ifc
nve-only cluster
nameif ccl
security-level 0
```

```
ip address 10.6.6.51 255.255.255.0
no shutdown
!
! VXLAN Network Identifier (VNI) interface
interface vni1
segment-id 1
vtep-nve 1
!
! Set the CCL MTU
mtu ccl 1664
!
! Network Virtualization Endpoint (NVE) association with VTEP src interface
nve 1
encapsulation vxlan
source-interface ccl
peer-group cluster-peers
!
! Management Interface Using DHCP
interface management 0/0
nameif management
ip address dhcp setroute
no shutdown
!
! Alternate Management Using Static IP
! ip local pool mgmt_pool 10.1.1.1 10.10.10.4
! interface management 0/0
! nameif management
! ip address 10.1.1.25 255.255.255.0 cluster-pool mgmt_pool
! no shutdown
!
! Cluster Config
cluster group cluster1
local-unit A
cluster-interface vni1 ip 10.2.2.1 255.255.255.0
priority 1
enable noconfirm
!
! INTERFACES
!
ip local pool inside_pool 10.10.10.11 10.10.10.14
ip local pool outside_pool 10.11.11.11 10.11.11.14
!
interface GigabitEthernet0/1
nameif inside
security-level 100
ip address 10.10.10.10 255.255.255.0 cluster-pool inside_pool
!
interface GigabitEthernet0/0
nameif outside
security-level 0
ip address 10.11.11.10 255.255.255.0 cluster-pool outside_pool
!
!JUMBO FRAME RESERVATION for CCL MTU
jumbo-frame reservation
```

データノード Day0 設定

データノードの次の Day0 設定には、ブートストラップ設定のみが含まれています。太字のテキストは、制御ノードの Day0 設定から変更する必要がある値を示しています。



- (注) この設定には、クラスタ中心の設定のみが含まれます。Day0 設定には、ライセンス、SSH アクセス、ASDM アクセスなどの他の設定も含める必要があります。Day0 設定の詳細については、スタートアップガイドを参照してください。

```

!BOOTSTRAP
! Cluster interface mode
cluster interface mode individual
!
! VXLAN peer group
object-group network cluster-peers
network-object host 10.6.6.51
network-object host 10.6.6.52
network-object host 10.6.6.53
network-object host 10.6.6.54
!
! Alternate object group representation
! object-network xyz
! range 10.6.6.51 10.6.6.54
! object-group network cluster-peers
! network-object object xyz
!
! Cluster control link physical interface (VXLAN tunnel endpoint (VTEP) src interface)
interface gigabitethernet 0/7
description CCL VTEP src ifc
nve-only cluster
nameif ccl
security-level 0
ip address 10.6.6.52 255.255.255.0
no shutdown
!
! VXLAN Network Identifier (VNI) interface
interface vni1
segment-id 1
vtep-nve 1
!
! Set the CCL MTU
mtu ccl 1664
!
! Network Virtualization Endpoint (NVE) association with VTEP src interface
nve 1
encapsulation vxlan
source-interface ccl
peer-group cluster-peers
!
! Management Interface Using DHCP
interface management 0/0
nameif management
ip address dhcp setroute
no shutdown
!
! Alternate Management Using Static IP
! ip local pool mgmt_pool 10.1.1.1 10.10.10.4
! interface management 0/0
! nameif management
! ip address 10.1.1.25 255.255.255.0 cluster-pool mgmt_pool
! no shutdown
!
! Cluster Config
cluster group cluster1

```

```
local-unit B
cluster-interface vni1 ip 10.2.2.2 255.255.255.0
priority 2
enable noconfirm
!
! INTERFACES
!
ip local pool inside_pool 10.10.10.11 10.10.10.14
ip local pool outside_pool 10.11.11.11 10.11.11.14
!
interface GigabitEthernet0/1
nameif inside
security-level 100
ip address 10.10.10.10 255.255.255.0 cluster-pool inside_pool
!
interface GigabitEthernet0/0
nameif outside
security-level 0
ip address 10.11.11.10 255.255.255.0 cluster-pool outside_pool
!
!JUMBO FRAME RESERVATION for CCL MTU
jumbo-frame reservation
```

展開後のASA 仮想クラスタリングの設定

ASA 仮想の展開後にクラスタリングを設定するには、次のタスクを実行します。

インターフェイスの設定

各ノードのクラスタ インターフェイス モードと、制御ノードのインターフェイスを設定します。インターフェイス構成は、クラスタに参加するときにデータノードに複製されます。クラスタ制御リンクの構成は、ブートストラップコンフィギュレーション手順で説明されていることに注意してください。

各ノードのでクラスタ インターフェイス モードを設定する

クラスタリングを有効にする前に、個々のインターフェイスを使用するようにファイアウォールを変換する必要があります。クラスタリングによって使用できるインターフェイスの種類が制限されるため、このプロセスでは、既存の設定に互換性のないインターフェイスがあるかどうかを確認し、サポートされていないインターフェイスを設定できないようにします。

始める前に

- モードの設定は、クラスタに追加する各 ASA 仮想 で個別に行う必要があります。
- コンソールポート（使用可能な場合）またはSSH（設定されている場合）のいずれかを使用して、ASA 仮想 CLI に接続します。これらのオプションのいずれも使用できない場合は、ASDM を使用してクラスタリングを設定できます。

手順

ステップ1 互換性のないコンフィギュレーションを表示し、強制的にインターフェイスモードにして後でコンフィギュレーションを修正できるようにします。このコマンドではモードは変更されません。

cluster interface-mode individual check-details

例：

```
ciscoasa(config)# cluster interface-mode individual check-details
```

注意 インターフェイス モードを設定した後は、SSH を使用して常にインターフェイスに接続できるようになります。ただし、クラスタリング要件に適合するように管理インターフェイスを設定する前に ASA をリロードすると（たとえば、クラスタ IP プールを追加するため、または DHCP から IP アドレスを取得するため）、クラスタと互換性のないインターフェイスコンフィギュレーションが削除されるため、再接続できなくなります。その場合は、可能であればコンソールポートに接続してインターフェイス コンフィギュレーションを修正する必要があります。

ステップ2 クラスタリング用にインターフェイス モードを設定します。

cluster interface-mode individual force

例：

```
ciscoasa(config)# cluster interface-mode individual force
```

デフォルト設定はありません。明示的にモードを選択する必要があります。モードを設定していない場合は、クラスタリングをイネーブルにできません。

force オプションを指定すると、互換性のないコンフィギュレーションの検査は行わずにモードが変更されます。コンフィギュレーションの問題がある場合は、モードを変更した後に手動で解決する必要があります。インターフェイス コンフィギュレーションの修正ができるのはモードの設定後に限られるので、**force** オプションを使用することを推奨します。このようにすれば、最低でも、既存のコンフィギュレーションの状態から開始できます。さらにガイドランスが必要な場合は、モードを設定した後で **check-details** オプションを再実行します。

force オプションを指定しないと、互換性のないコンフィギュレーションがある場合は、コンフィギュレーションをクリアしてリロードするように求められるので、コンソールポート（可能な場合）に接続して管理アクセスを再設定する必要があります。コンフィギュレーションに互換性の問題がない場合は（まれなケース）、モードが変更され、コンフィギュレーションは維持されます。コンフィギュレーションをクリアしたくない場合は、**n** を入力してコマンドを終了します。

インターフェイス モードを解除するには、**no cluster interface-mode** コマンドを入力します。

個々のインターフェイスの設定

クラスタリングを有効にする前に、現在 IP アドレスが設定されているインターフェイスをクラスタ対応に変更する必要があります。管理に静的 IP アドレスを使用する場合は、少なくとも、SSH が現在接続されている管理インターフェイスを変更する必要がある場合があります。他のインターフェイスについては、クラスタリングを有効化する前またはその後に設定できません。完全なコンフィギュレーションが新しいクラスタノードと同期するように、すべてのインターフェイスを事前に設定することを推奨します。

ここでは、個々のインターフェイスがクラスタリング互換となるようにインターフェイスを設定する方法について説明します。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、常に現在の制御ノードに属します。すべてのデータインターフェイスは個別インターフェイスである必要があります。

管理インターフェイスでは、IP アドレスプールを設定するか、DHCP を使用できます。管理インターフェイスのみが DHCP からのアドレスの取得をサポートしています。DHCP を使用する場合は、この手順を使用しないでください。代わりに、通常どおりに設定します。

始める前に

- (オプション) サブインターフェイスを設定します。
- 管理インターフェイスには、静的アドレスを使用するか、DHCP を使用できます。静的 IP アドレスを使用しており、SSH を使用して管理インターフェイスにリモートに接続している場合は、将来のデータノードの現在の IP アドレスは一時的なものです。
 - 各メンバには、制御ノードで定義されたクラスタ IP プールから IP アドレスが割り当てられます。
 - クラスタ IP プールには、将来のセカンダリ IP アドレスを含む、ネットワークですでに使用中のアドレスを含めることはできません。

次に例を示します。

1. 制御ノードに 10.1.1.1 を設定します。
2. 他のノードには、10.1.1.2、10.1.1.3、10.1.1.4 を使用します。
3. 制御ノードのクラスタの IP プールを設定する場合、使用中であるために .2、.3、.4 のアドレスをプールに含めることはできません。
4. 代わりに、.5、.6、.7、.8 のような、ネットワークの他の IP アドレスを使用する必要があります。



(注) プールには、制御ノードを含むクラスタのメンバ数分のアドレスが必要です。元の .1 アドレスはメインクラスタ IP アドレスであり、現在の制御ノードのものであります。

5. クラスタに参加すると古い一時的なアドレスは放棄され、他の場所で使用できません。

手順

ステップ 1 ローカル IP アドレス (IPv4 と IPv6 の一方または両方) のプールを設定します。このアドレスの 1 つが、このインターフェイス用に各クラスタノードに割り当てられます。

(IPv4)

ip local pool *poolname first-address — last-address [mask mask]*

(IPv6)

ipv6 local pool *poolname ipv6-address/prefix-length number_of_addresses*

例 :

```
ciscoasa(config)# ip local pool ins 192.168.1.2-192.168.1.9
ciscoasa(config-if)# ipv6 local pool insipv6 2001:DB8:45:1003/64 8
```

少なくともクラスタ内のノードと同じ数のアドレスを含めます。クラスタを拡張する予定の場合は、アドレスを増やします。現在の制御ノードに属するメインクラスタ IP アドレスは、このプールの一部ではありません。必ず、同じネットワークの IP アドレスの 1 つをメインクラスタ IP アドレス用に確保してください。

各ノードに割り当てられるローカルアドレスを、事前に正確に特定することはできません。各ノードで使用されているアドレスを表示するには、**show ip[v6] local pool poolname** コマンドを入力します。各クラスタ メンバには、クラスタに参加したときにメンバ ID が割り当てられます。この ID によって、プールから使用されるローカル IP が決定します。

ステップ 2 インターフェイス コンフィギュレーション モードを開始します。

interface *interface_id*

例 :

```
ciscoasa(config)# interface gigabitethernet 0/1
```

ステップ 3 インターフェイスの名前を指定します。

nameif *name*

例 :

```
ciscoasa(config-if)# nameif inside
```

name は最大 48 文字のテキスト文字列です。大文字と小文字は区別されません。名前を変更するには、このコマンドで新しい値を再入力します。

ステップ 4 メインクラスタの IP アドレスを設定し、クラスタ プールを指定します。

(IPv4)

ip address *ip_address* [*mask*] **cluster-pool** *poolname*

(IPv6)

ipv6 address *ipv6-address/prefix-length* **cluster-pool** *poolname*

例 :

```
ciscoasa(config-if)# ip address 192.168.1.1 255.255.255.0 cluster-pool ins
ciscoasa(config-if)# ipv6 address 2001:DB8:45:1003::99/64 cluster-pool insipv6
```

この IP アドレスは、クラスタ プールアドレスと同じネットワーク上に存在している必要がありますが、プールに含まれてはなりません。IPv4 アドレスと IPv6 アドレスの一方または両方を設定できます。

DHCP、PPPoE、および IPv6 自動設定はサポートされません。IP アドレスを手動で設定する必要があります。

ステップ 5 セキュリティ レベルを設定します。 *number* には、0 (最低) ~ 100 (最高) の整数を指定します。

security-level *number*

例 :

```
ciscoasa(config-if)# security-level 100
```

ステップ 6 インターフェイスをイネーブルにします。

no shutdown

例

次の例では、管理 0/0、GigabitEthernet 0/0、および GigabitEthernet 0/1 インターフェイスを個別のインターフェイスとして設定します。

```
ip local pool mgmt 10.1.1.2-10.1.1.9
ipv6 local pool mgmtipv6 2001:DB8:45:1002/64 8

interface management 0/0
 nameif management
 ip address 10.1.1.1 255.255.255.0 cluster-pool mgmt
 ipv6 address 2001:DB8:45:1001::99/64 cluster-pool mgmtipv6
 security-level 100
 no shutdown

ip local pool out 209.165.200.225-209.165.200.232
ipv6 local pool outipv6 2001:DB8:45:1002/64 8

interface gigabitethernet 0/0
 nameif outside
 ip address 209.165.200.233 255.255.255.224 cluster-pool out
```

```

ipv6 address 2001:DB8:45:1002::99/64 cluster-pool outipv6
security-level 0
no shutdown

ip local pool ins 192.168.1.2-192.168.1.9
ipv6 local pool insipv6 2001:DB8:45:1003/64 8

interface gigabitethernet 0/1
nameif inside
ip address 192.168.1.1 255.255.255.0 cluster-pool ins
ipv6 address 2001:DB8:45:1003::99/64 cluster-pool insipv6
security-level 100
no shutdown

```

ブートストラップコンフィギュレーションの作成

クラスタ内の各ノードがクラスタに参加するには、ブートストラップ設定が必要です。

制御ノードのブートストラップの設定

クラスタ内の各ノードがクラスタに参加するには、ブートストラップ設定が必要です。一般的には、クラスタに参加するように最初に設定したノードが制御ノードとなります。クラスタリングをイネーブルにした後で、選定期間が経過すると、クラスタの制御ノードが選定されます。最初はクラスタ内に1つのノードしかないため、そのノードが制御ノードになります。クラスタに追加する後続のノードはデータノードになります。

始める前に

- コンフィギュレーションをバックアップします。後でクラスタから脱退する必要があるときに備えて、コンフィギュレーションを復元できるようにしておくためです。
- クラスタ制御リンクと管理インターフェイス（オプションでDHCPを使用可能）を除いて、コンフィギュレーション内のインターフェイスはすべて、クラスタ IP プールを指定して設定されている必要があります。この設定は、クラスタリングを有効化する前に行います。既存のインターフェイスコンフィギュレーションがある場合は、そのインターフェイスコンフィギュレーションをクリアすることも（**clear configure interface**）、インターフェイスをクラスタインターフェイスに変換することもできます。これは、クラスタリングをイネーブルにする前に行います。
- 稼働中のクラスタにノードを追加すると、一時的に、限定的なパケット/接続ドロップが発生することがありますが、これは想定内の動作です。
- クラスタ制御リンクで使用するジャンボフレーム予約を有効にして、クラスタ制御リンクの MTU を推奨値に設定できるようにします。**jumbo-frame reservation** コマンドを参照してください。ジャンボフレームを有効にすると ASA がリロードされるため、この手順を進める前に実行しておく必要があります。

手順

ステップ 1 クラスタに参加する前に、クラスタ制御リンクインターフェイスのVXLANインターフェイスを設定します。

後でクラスタリングを有効化するときに、このインターフェイスをクラスタ制御リンクとして識別します。

クラスタ制御リンクインターフェイスコンフィギュレーションは、制御ノードからデータノードには複製されませんが、同じコンフィギュレーションを各ノードで使用する必要があります。このコンフィギュレーションは複製されないため、クラスタ制御リンクインターフェイスの設定は各ノードで個別に行う必要があります。

a) ネットワークオブジェクトグループを作成して、VTEPピアのIPアドレスを識別します。

ネットワークオブジェクトグループの詳細については、ASA ファイアウォール コンフィギュレーションガイドの「Objects for Access Control」の章を参照してください。

VTEP間の基礎となるIPネットワークは、VNIインターフェイスが使用するクラスタ制御リンクネットワークから独立しています。各VTEPソースインターフェイスには、同じサブネット上のIPアドレスがあります。このサブネットは、他のすべてのトラフィックからは隔離し、クラスタ制御リンクインターフェイスだけが含まれるようにしてください。

例：

次に、インラインで定義されたホストを含むネットワークオブジェクトグループを作成する例を示します。

```
ciscoasa(config)# object-group network cluster-peers
ciscoasa(network-object-group)# network-object host 10.6.6.51
ciscoasa(network-object-group)# network-object host 10.6.6.52
ciscoasa(network-object-group)# network-object host 10.6.6.53
ciscoasa(network-object-group)# network-object host 10.6.6.54
```

次の例では、スタンドアロンネットワークオブジェクトを参照するネットワークオブジェクトグループを作成します。

```
ciscoasa(config)# object network xyz
ciscoasa(config-network-object)# range 10.6.6.51 10.6.6.54

ciscoasa(config)# object-group network cluster-peers
ciscoasa(network-object-group)# network-object object xyz
```

b) VTEP送信元インターフェイスを設定します。

```
interface interface_id
nve-only cluster
nameif name
ip address ip_address subnet_mask
no shutdown
```

IP アドレスは、ネットワーク オブジェクト グループのピアの 1 つとして含める必要があります。

例 :

```
ciscoasa(config)# interface gigabitethernet 0/7
ciscoasa(config-if)# nve-only cluster
ciscoasa(config-if)# nameif ccl
ciscoasa(config-if)# ip address 10.6.6.51 255.255.255.0
ciscoasa(config-if)# no shutdown
```

- c) VTEP ソースインターフェイスを NVE インスタンスに関連付けます。

nve 1

source-interface *interface-name*

peer-group *network_object_name*

ID 1 で NVE インスタンスを 1 つだけ指定できます。

encapsulation vxlan コマンドが NVE インスタンスのデフォルトにより追加されます。明示的に追加する必要はありません。

例 :

```
ciscoasa(config)# nve 1
ciscoasa(cfg-nve)# source-interface ccl
ciscoasa(cfg-nve)# peer-group cluster-peers
```

- d) VTEP ソースインターフェイスの最大伝送ユニットを指定します。データインターフェイスの最大 MTU より少なくとも 154 バイト高い値を指定します。

mtu *interface_name bytes*

クラスタ制御リンクのトラフィックにはデータパケット転送が含まれるため、クラスタ制御リンクはデータパケット全体のサイズに加えてクラスタトラフィックのオーバーヘッド (100 バイト) および VXLAN のオーバーヘッド (54 バイト) にも対応する必要があります。MTU を 1554 ~ 9198 バイトの間で設定します。デフォルトの MTU は 1554 バイトです。データインターフェイスが 1500 に設定されている場合は、クラスタ制御リンクの MTU を 1654 に設定することをお勧めします。この値にはジャンボ フレームの予約が必要です (**jumbo-frame reservation** コマンドを参照)。

たとえばジャンボ フレームを使用している場合、最大 MTU は 9198 バイトであるため、データインターフェイスの最大 MTU は 9044 になり、クラスタ制御リンクは 9198 に設定できます。

このコマンドはデータノードに複製されますが、ブートストラップ設定とともにこの設定を構成することをお勧めします。

例 :

```
ciscoasa(config)# mtu ccl 1654
```

- e) (任意) VXLAN UDP ポートを設定します。

vxlan port number

デフォルトでは、VTEP 送信元インターフェイスは UDP ポート 4789 への VXLAN トラフィックを受け入れます。ネットワークで標準以外のポートを使用する場合は、それを変更できます。

例 :

```
ciscoasa(config)# vxlan port 5678
```

- f) VNI インターフェイスを作成します。

interface vni vni_num

segment-id id

vtep-nve 1

例 :

```
ciscoasa(config)# interface vni 1
ciscoasa(config-if)# segment-id 1000
ciscoasa(config-if)# vtep-nve 1
```

- 1 ~ 10000 の範囲で VNI 番号を設定します。この ID は内部インターフェイス識別子です。
- 1 ~ 16777215 の範囲でセグメント ID を設定します。セグメント ID は VXLAN タギングに使用されます。

インターフェイスの名前などのパラメータを設定しないでください。

- ステップ 2** クラスタに名前を付け、クラスタ コンフィギュレーションモードにします。

cluster group name

例 :

```
ciscoasa(config)# cluster group pod1
```

名前は 1 ~ 38 文字の ASCII 文字列であることが必要です。ノードごとに設定できるクラスタグループは1つだけです。クラスタのすべてのメンバが同じ名前を使用する必要があります。

- ステップ 3** クラスタのこのメンバの名前を指定します。

local-unit node_name

1 ~ 38 文字の一意的 ASCII 文字列を使用します。各ノードには一意の名前が必要です。クラスタ内の他のノードと同じ名前を付けることはできません。

例 :

```
ciscoasa(cfg-cluster)# local-unit node1
```

ステップ4 クラスタ制御リンク VNI インターフェイスを指定します。

cluster-interface vni_interface_id ip ip_address mask

例：

```
ciscoasa(cfg-cluster)# cluster-interface vni1 ip 192.168.1.1 255.255.255.0
INFO: Non-cluster interface config is cleared on VNI1
```

IP アドレスには IPv4 アドレスを指定します。IPv6 は、このインターフェイスではサポートされません。ノードごとに、同じネットワーク上の異なる IP アドレスを指定します。VNI ネットワークは、物理 VTEP ネットワーク上で稼働する暗号化された仮想ネットワークです。

ステップ5 制御ノードの選択に対するこのノードのプライオリティを設定します。

priority priority_number

例：

```
ciscoasa(cfg-cluster)# priority 1
```

プライオリティは 1 ~ 100 であり、1 が最高のプライオリティです。

ステップ6 (オプション) クラスタ制御リンクの制御トラフィックの認証キーを設定します。

key shared_secret

例：

```
ciscoasa(cfg-cluster)# key chuntheunavoidable
```

共有秘密は、1 ~ 63 文字の ASCII 文字列です。共有秘密は、キーを生成するために使用されます。このコマンドは、データパストラフィック（接続状態アップデートや転送されるパケットなど）には影響しません。データパストラフィックは、常にクリアテキストとして送信されます。

ステップ7 クラスタリングをイネーブルにします。

enable [noconfirm]

例：

```
ciscoasa(cfg-cluster)# enable
INFO: Clustering is not compatible with following commands:
policy-map global_policy
  class inspection_default
    inspect skinny
policy-map global_policy
  class inspection_default
    inspect sip
Would you like to remove these commands? [Y]es/[N]o:Y
```

```
INFO: Removing incompatible commands from running configuration...
Cryptochecksum (changed): f16b7fc2 a742727e e40bc0b0 cd169999
INFO: Done
```

enable コマンドが入力されると、ASA は実行コンフィギュレーションをスキャンして、クラスタリングに対応していない機能の非互換コマンドの有無を調べます。デフォルト コンフィギュレーションにあるコマンドも、これに該当することがあります。互換性のないコマンドを削除するように求められます。応答として **No** を入力した場合は、クラスタリングはイネーブルになりません。確認を省略し、互換性のないコマンドを自動的に削除するには、**noconfirm** キーワードを使用します。

最初にイネーブルにしたノードについては、制御ノード選定が発生します。これまでは最初のノードがクラスタの唯一のメンバーである必要があるため、これが制御ノードになります。この期間中にコンフィギュレーション変更を実行しないでください。

クラスタリングをディセーブルにするには、**no enable** コマンドを入力します。

(注) クラスタリングをディセーブルにした場合は、すべてのデータインターフェイスがシャットダウンされ、管理インターフェイスだけがアクティブになります。

例

次の例では、管理インターフェイス、内部インターフェイス、外部インターフェイス、および VXLAN クラスタ制御リンクを設定し、その後で、「node1」という名前の ASA のクラスタリングを有効化します。これは最初にクラスタに追加されるノードであるため、制御ノードになります。

```
ip local pool mgmt 10.1.1.2-10.1.1.9
ipv6 local pool mgmtipv6 2001:DB8:45:1002/64 8

interface management 0/0
  nameif management
  ip address 10.1.1.1 255.255.255.0 cluster-pool mgmt
  ipv6 address 2001:DB8:45:1001::99/64 cluster-pool mgmtipv6
  security-level 100
  no shutdown

ip local pool out 209.165.200.225-209.165.200.232
ipv6 local pool outipv6 2001:DB8:45:1002/64 8

interface gigabitethernet 0/0
  nameif outside
  ip address 209.165.200.233 255.255.255.224 cluster-pool out
  ipv6 address 2001:DB8:45:1002::99/64 cluster-pool outipv6
  security-level 0
  no shutdown

ip local pool ins 192.168.1.2-192.168.1.9
ipv6 local pool insipv6 2001:DB8:45:1003/64 8

interface gigabitethernet 0/1
  nameif inside
  ip address 192.168.1.1 255.255.255.0 cluster-pool ins
```

データノードのブートストラップの設定

```

ipv6 address 2001:DB8:45:1003::99/64 cluster-pool insipv6
security-level 100
no shutdown

object-group network cluster-peers
network-object host 10.6.6.51
network-object host 10.6.6.52
network-object host 10.6.6.53
network-object host 10.6.6.54

interface gigabitethernet 0/7
nve-only cluster
nameif ccl
ip address 10.6.6.51 255.255.255.0
no shutdown

nve 1
source-interface ccl
peer-group cluster-peers

mtu ccl 1654

interface vni 1
segment-id 1000
vtep-nve 1

cluster group pod1
local-unit node1
cluster-interface vni1 ip 192.168.1.1 255.255.255.0
priority 1
key 67impala
enable noconfirm

```

データノードのブートストラップの設定

データノードを設定するには、次の手順に従います。

始める前に

- コンフィギュレーションをバックアップします。後でクラスタから脱退する必要が生じたときに備えて、コンフィギュレーションを復元できるようにしておくためです。
- クラスタ制御リンクと管理インターフェイス（オプションでDHCPを使用可能）を除いて、コンフィギュレーション内のインターフェイスはすべて、クラスタ IP プールを指定して設定されている必要があります。この設定は、クラスタリングを有効化する前に行います。既存のインターフェイスコンフィギュレーションがある場合は、そのインターフェイスコンフィギュレーションをクリアすることも（**clear configure interface**）、インターフェイスをクラスタインターフェイスに変換することもできます。これは、クラスタリングをイネーブルにする前に行います。
- 稼働中のクラスタにノードを追加すると、一時的に、限定的なパケット/接続ドロップが発生することがありますが、これは想定内の動作です。
- クラスタ制御リンクで使用するジャンボフレーム予約を有効にして、クラスタ制御リンクの MTU を推奨値に設定できるようにします。**jumbo-frame reservation** コマンドを参照し

てください。ジャンボフレームを有効にすると ASA がリロードされるため、この手順を進める前に実行しておく必要があります。

手順

ステップ 1 制御ノードに設定したものと同じクラスタ制御リンクインターフェイスを設定します。VTEP ソースインターフェイスに別の IP アドレスを指定してください (**太字**で表示)。

例 :

```
ciscoasa(config)# object-group network cluster-peers
ciscoasa(network-object-group)# network-object host 10.6.6.51
ciscoasa(network-object-group)# network-object host 10.6.6.52
ciscoasa(network-object-group)# network-object host 10.6.6.53
ciscoasa(network-object-group)# network-object host 10.6.6.54
ciscoasa(config)# interface gigabitethernet 0/7
ciscoasa(config-if)# nve-only cluster
ciscoasa(config-if)# nameif ccl
ciscoasa(config-if)# ip address 10.6.6.52 255.255.255.0
ciscoasa(config-if)# no shutdown
ciscoasa(config)# nve 1
ciscoasa(cfg-nve)# source-interface ccl
ciscoasa(cfg-nve)# peer-group cluster-peers
ciscoasa(config)# mtu ccl 1654
ciscoasa(config)# interface vni 1
ciscoasa(config-if)# segment-id 1000
ciscoasa(config-if)# vtep-nve 1
```

ステップ 2 制御ノードに設定したものと同じクラスタ名を指定します。

例 :

```
ciscoasa(config)# cluster group pod1
```

ステップ 3 クラスタのこのメンバに一意の文字列で名前を指定します。

local-unit node_name

例 :

```
ciscoasa(cfg-cluster)# local-unit node2
```

1 ~ 38 文字の ASCII 文字列を指定します。

各ノードには一意の名前が必要です。クラスタ内の他のノードと同じ名前を付けることはできません。

ステップ 4 制御ノードに設定したものと同じクラスタ制御リンクインターフェイスを指定しますが、ノードごとに同じネットワーク上の異なる IP アドレスを指定します。

cluster-interface vni_interface_id ip ip_address mask

例 :

```
ciscoasa(cfg-cluster)# cluster-interface vni1 ip 192.168.1.2 255.255.255.0
INFO: Non-cluster interface config is cleared on VNI1
```

IP アドレスには IPv4 アドレスを指定します。IPv6 は、このインターフェイスではサポートされません。このインターフェイスには、**nameif** を設定することはできません。

ステップ 5 サイト間クラスタリングを使用している場合、このノードのサイト ID を設定し、サイト固有の MAC アドレスが使用されるようにします。

site-id number

例 :

```
ciscoasa(cfg-cluster)# site-id 2
```

number は 1 ~ 8 です。

ステップ 6 制御ノードの選定に対するこのノードのプライオリティを設定します。通常は、制御ノードより高い値にします。

priority priority_number

例 :

```
ciscoasa(cfg-cluster)# priority 2
```

プライオリティを 1 ~ 100 に設定します。1 が最高のプライオリティです。

ステップ 7 制御ノードに設定したものと同一認証キーを設定します。

例 :

```
ciscoasa(cfg-cluster)# key chuntheunavoidable
```

ステップ 8 クラスタリングをイネーブルにします。

enable as-data-node

enable as-data-node コマンドを使用することによって、設定に関するすべての非互換性（主にまだクラスタリング用に設定されていないインターフェイスの存在）を回避できます。このコマンドを実行すると、クラスタに参加させるデータノードが現在の選定において制御ノードとなる可能性をなくすことができます。データノードのコンフィギュレーションは、制御ノードから同期されたコンフィギュレーションによって上書きされます。

クラスタリングをディセーブルにするには、**no enable** コマンドを入力します。

(注) クラスタリングをディセーブルにした場合は、すべてのデータインターフェイスがシャットダウンされ、管理インターフェイスだけがアクティブになります。

例

次の例には、データノード **node2** の設定が含まれています。

```
object-group network cluster-peers
  network-object host 10.6.6.51
  network-object host 10.6.6.52
  network-object host 10.6.6.53
  network-object host 10.6.6.54

interface gigabitethernet 0/7
  nve-only cluster
  nameif ccl
  ip address 10.6.6.52 255.255.255.0
  no shutdown

nve 1
  source-interface ccl
  peer-group cluster-peers

mtu ccl 1654

interface vni 1
  segment-id 1000
  vtep-nve 1

cluster group pod1
  local-unit node2
  cluster-interface vni1 ip 192.168.1.2 255.255.255.0
  priority 2
  key 67impala
  enable noconfirm
```

クラスタリング動作のカスタマイズ

Day0 設定の一環として、またはクラスタの展開後に、クラスタリングヘルスモニタリング、TCP 接続複製の遅延、フローのモビリティ、他の最適化をカスタマイズできます。

制御ノードで次の手順を実行します。

ASA クラスタの基本パラメータの設定

制御ノード上のクラスタ設定をカスタマイズできます。

手順

ステップ1 クラスタの設定モードを開始します。

```
cluster group name
```

ステップ2 (任意) データノードから制御ノードへのコンソール複製を有効にします。

console-replicate

この機能はデフォルトで無効に設定されています。ASAは、特定の重大イベントが発生したときに、メッセージを直接コンソールに出力します。コンソール複製を有効にすると、データノードから制御ノードにコンソールメッセージが送信されるので、モニタする必要があるのはクラスタのコンソールポート1つだけです。

ステップ3 クラスタリング イベントの最小トレース レベルを設定します。

trace-level level

必要に応じて最小レベルを設定します。

- **critical** : クリティカル イベント (重大度 = 1)
- **warning** : 警告 (重大度 = 2)
- **informational** : 情報イベント (重大度 = 3)
- **debug** : デバッグ イベント (重大度 = 4)

ヘルスマニタリングおよび自動再参加設定の設定

この手順では、ノードとインターフェイスのヘルスマニタリングを設定します。

たとえば、管理インターフェイスなど、必須以外のインターフェイスのヘルスマニタリングをディセーブルにすることができます。ヘルスマニタリングはVLANサブインターフェイスでは実行されません。クラスタ制御リンクのモニタリングは設定できません。このリンクは常にモニターされています。

手順

ステップ1 クラスタの設定モードを開始します。

cluster group name

例 :

```
ciscoasa(config)# cluster group test
ciscoasa(cfg-cluster)#
```

ステップ2 クラスタノードのヘルスチェック機能をカスタマイズします。

health-check [holdtime timeout]

ノードのヘルスを確認するため、ASAのクラスタノードはクラスタ制御リンクで他のノードにハートビートメッセージを送信します。ノードが保留時間内にピアノードからハートビートメッセージを受信しない場合、そのピアノードは応答不能またはデッド状態と見なされます。

- **holdtime timeout** : ノードのハートビートステータスメッセージの時間間隔を指定します。指定できる範囲は .3 ~ 45 秒で、デフォルトは 3 秒です。

何らかのトポロジ変更を行うとき（たとえば、データインターフェイスの追加または削除、ASA またはスイッチ上のインターフェイスの有効化または無効化）は、ヘルスチェック機能を無効にし、無効化したインターフェイスのインターフェイスモニタリングも無効にする必要があります（**no health-check monitor-interface**）。トポロジの変更が完了して、設定の変更がすべてのノードに同期されたら、ヘルスチェック機能を再度有効にできます。

例 :

```
ciscoasa(cfg-cluster)# health-check holdtime 5
```

ステップ 3 インターフェイスでインターフェイスヘルスチェックを無効化します。

no health-check monitor-interface interface_id

インターフェイスのヘルスチェックはリンク障害をモニターします。ASA がメンバーをクラスタから削除するまでの時間は、そのノードが確立済みメンバーであるか、またはクラスタに参加しようとしているかによって異なります。デフォルトでは、ヘルスチェックはすべてのインターフェイスでイネーブルになっています。このコマンドの **no** 形式を使用してディセーブル（無効）にすることができます。たとえば、管理インターフェイスなど、必須以外のインターフェイスのヘルスマニタリングをディセーブルにすることができます。

- **interface_id** : インターフェイスの監視を無効にします。ヘルスマニタリングは VLAN サブインターフェイスでは実行されません。クラスタ制御リンクのモニタリングは設定できません。このリンクは常にモニターされています。

何らかのトポロジ変更を行うとき（たとえば、データインターフェイスの追加または削除、ASA またはスイッチ上のインターフェイスの有効化または無効化）は、ヘルスチェック機能を無効（**no health-check**）にし、無効化したインターフェイスのインターフェイスモニタリングも無効にする必要があります。トポロジの変更が完了して、設定の変更がすべてのノードに同期されたら、ヘルスチェック機能を再度有効にできます。

例 :

```
ciscoasa(cfg-cluster)# no health-check monitor-interface management1/1
```

ステップ 4 ヘルスチェック失敗後の自動再結合クラスタ設定をカスタマイズします。

health-check {data-interface | cluster-interface | system} auto-rejoin [unlimited | auto_rejoin_max] auto_rejoin_interval auto_rejoin_interval_variation

- **system** : 内部エラー時の自動再結合の設定を行います。内部の障害には、アプリケーション同期のタイムアウト、矛盾したアプリケーションステータスなどがあります。
- **unlimited** : (**cluster-interface** のデフォルト) 再結合の試行回数を制限しません。
- **auto-rejoin-max** : 再結合の試行回数を 0 ~ 65535 の範囲の値に設定します。0 は自動再結合を無効化します。**data-interface** と **system** のデフォルトは 3 です。

- **auto_rejoin_interval** : 再結合試行の間隔を 2 ~ 60 の範囲の分単位で定義します。デフォルト値は 5 分です。クラスタへの再参加をノードが試行する最大合計時間は、最後の障害発生時から 14400 分 (10 日) に制限されます。
- **auto_rejoin_interval_variation** : 間隔を増加させるかどうかを定義します。1 ~ 3 の範囲で値を設定します (1 : 変更なし、2 : 直前の間隔の 2 倍、3 : 直前の間隔の 3 倍)。たとえば、間隔を 5 分に設定し、変分を 2 に設定した場合は、最初の試行が 5 分後、2 回目の試行が 10 分後 (2 x 5)、3 階目の試行が 20 分後 (2 x 10) となります。デフォルト値は、クラスタインターフェイスの場合は 1、データインターフェイスおよびシステムの場合は 2 です。

例 :

```
ciscoasa(cfg-cluster)# health-check data-interface auto-rejoin 10 3 3
```

ステップ 5 ASA がインターフェイスを障害が発生していると思なし、クラスタからノードが削除されるまでのデバウンス時間を設定します。

health-check monitor-interface debounce-time ms

例 :

```
ciscoasa(cfg-cluster)# health-check monitor-interface debounce-time 300
```

デバウンス時間は 300 ~ 9000 ms の範囲の値を設定します。デフォルトは 500 ms です。値を小さくすると、インターフェイスの障害をより迅速に検出できます。デバウンス時間を短くすると、誤検出の可能性が高くなることに注意してください。インターフェイスのステータス更新が発生すると、ASA はインターフェイスを障害としてマークし、クラスタからノードを削除するまで指定されたミリ秒数待機します。

ステップ 6 (任意) トラフィック負荷のモニタリングを設定します。

load-monitor [frequency seconds] [intervals intervals]

- **seconds**: モニタリングメッセージ間の時間を、10 ~ 360 秒の範囲で設定します。 **frequency** デフォルトは 20 秒です。
- 間隔 (**interval**) : ASA がデータを保持する間隔の数を 1 ~ 60 の範囲で設定します。 **intervals** デフォルトは 30 です。

クラスタメンバのトラフィック負荷をモニターできます。対象には、合計接続数、CPU とメモリの使用率、バッファドロップなどが含まれます。負荷が高すぎる場合、残りのノードが負荷を処理できる場合は、ノードのクラスタリングを手動で無効にするか、外部スイッチのロードバランシングを調整するかを選択できます。この機能は、デフォルトでイネーブルにされています。トラフィックの負荷を定期的にモニターできます。負荷が高すぎる場合は、ノードでクラスタリングを手動で無効にすることを選択できます。

トラフィック負荷を表示するには、**show cluster info load-monitor** コマンドを使用します。

例 :

```
ciscoasa(cfg-cluster)# load-monitor frequency 50 intervals 25
```

```

ciscoasa(cfg-cluster)# show cluster info load-monitor
ID Unit Name
0 B
1 A_1
Information from all units with 50 second interval:
Unit Connections Buffer Drops Memory Used CPU Used
Average from last 1 interval:
0 0 0 14 25
1 0 0 16 20
Average from last 25 interval:
0 0 0 12 28
1 0 0 13 27

```

例

次の例では、ヘルスチェック保留時間を .3 秒に設定し、管理 0/0 インターフェイスのモニタリングを無効にし、データインターフェイスの自動再結合の試行回数を 2 分から開始して前回の間隔の 3 倍増加させる計 4 回に設定し、クラスタ制御リンクの自動再結合の試行回数を 2 分おきの計 6 回に設定しています。

```

ciscoasa(config)# cluster group test
ciscoasa(cfg-cluster)# health-check holdtime .3
ciscoasa(cfg-cluster)# no health-check monitor-interface management0/0
ciscoasa(cfg-cluster)# health-check data-interface auto-rejoin 4 2 3
ciscoasa(cfg-cluster)# health-check cluster-interface auto-rejoin 6 2 1

```

接続リバランスおよびクラスタ TCP 複製遅延の設定

接続の再分散を設定できます。アップストリームまたはダウンストリームルータによるロードバランシングの結果として、フロー分散に偏りが生じた場合は、新しい TCP フローを過負荷のノードから他のノードにリダイレクトするように設定できます。既存のフローは他のノードには移動されません。

TCP 接続のクラスタ複製の遅延を有効化して、ディレクタ/バックアップフロー作成の遅延による存続期間が短いフローに関連する「不要な作業」を排除できます。ディレクタ/バックアップフローが作成される前にノードが失敗する場合は、それらのフローを回復することはできません。同様に、フローを作成する前にトラフィックが別のノードに再調整される場合、流れを回復することはできません。TCP のランダム化を無効化するトラフィックの TCP の複製の遅延を有効化しないようにする必要があります。

手順

ステップ 1 TCP 接続のクラスタ複製の遅延を有効化します。

```

cluster replication delay seconds { http | match tcp { host ip_address | ip_address mask | any | any4 | any6 } [{ eq | lt | gt } port] { host ip_address | ip_address mask | any | any4 | any6 } [{ eq | lt | gt } port] }

```

例 :

```
ciscoasa(config)# cluster replication delay 15 match tcp any any eq ftp
ciscoasa(config)# cluster replication delay 15 http
```

1 ～ 15 の範囲で秒数を設定します。**http** 遅延はデフォルトで 5 秒間有効になります。

ステップ 2 クラスタの設定モードを開始します。

```
cluster group name
```

ステップ 3 (オプション) TCP トラフィックの接続の再分散を有効化します。

```
conn-rebalance [frequency seconds]
```

例 :

```
ciscoasa(cfg-cluster)# conn-rebalance frequency 60
```

このコマンドは、デフォルトでディセーブルになっています。有効化されている場合は、ASA は負荷情報を定期的に交換し、新しい接続の負荷を高負荷のデバイスから低負荷のデバイスに移動します。負荷情報を交換する間隔を、1 ～ 360 秒の範囲内で指定します。デフォルトは 5 秒です。

サイト間トポロジに対しては接続の再分散を設定しないでください。異なるサイトのクラスタメンバには接続を再分散できません。

サイト間機能の設定

サイト間クラスタリングの場合、冗長性と安定性を高めるために、設定をカスタマイズできます。

ディレクタ ローカリゼーションの有効化

データセンターのサイト間クラスタリングのパフォーマンスを向上させ、ラウンドトリップ時間を短縮するために、ディレクターローカリゼーションをイネーブルにすることができます。通常、新しい接続は特定のサイト内のクラスタメンバーによってロードバランスされ、所有されています。しかし、ASA は任意のサイトのメンバーにディレクターロールを割り当てます。ディレクターローカリゼーションにより、所有者と同じサイトのローカルディレクタ、どのサイトにも存在可能なグローバルディレクタという追加のディレクターロールが有効になります。所有者とディレクタが同一サイトに存在すると、パフォーマンスが向上します。また、元の所有者が失敗した場合、ローカルなディレクタは同じサイトで新しい接続の所有者を選択します。グローバルなディレクタは、クラスタメンバーが別のサイトで所有される接続のパケットを受信する場合に使用されます。

始める前に

- ブートストラップ設定でクラスタメンバーのサイト ID を設定します。

- 次のトラフィック タイプは、ローカリゼーションをサポートしていません：NAT および PAT トラフィック、SCTP 検査されたトラフィック、フラグメンテーション所有クエリ。

手順

ステップ 1 クラスタの設定モードを開始します。

cluster group name

例：

```
ciscoasa(config)# cluster group cluster1
ciscoasa(cfg-cluster)#
```

ステップ 2 ディレクタ ローカリゼーションをイネーブルにします。

director-localization

サイト冗長性の有効化

サイトの障害からフローを保護するために、サイトの冗長性を有効にできます。接続バックアップオーナーがオーナーと同じサイトにある場合は、サイトの障害からフローを保護するために、追加のバックアップ オーナーが別のサイトから選択されます。

始める前に

- ブートストラップ設定でクラスタ メンバーのサイト ID を設定します。

手順

ステップ 1 クラスタの設定モードを開始します。

cluster group name

例：

```
ciscoasa(config)# cluster group cluster1
ciscoasa(cfg-cluster)#
```

ステップ 2 サイトの冗長性を有効にします。

site-redundancy

クラスタ フロー モビリティの設定

LISP のトラフィックを検査して、サーバーがサイト間を移動する時にフロー モビリティを有効にできます。

LISP インспекションについて

LISP トラフィックを検査することで、サイト間のフローのモビリティを有効にできます。

LISP について

VMware vMotion などのデータセンター仮想マシンのモビリティによって、サーバーはクライアントへの接続を維持すると同時に、データセンター間を移動できます。このようなデータセンター サーバモビリティをサポートするには、サーバーの移動時にサーバーへの入力ルートが更新できる必要があります。Cisco Locator/ID Separation Protocol (LISP) のアーキテクチャは、デバイス ID、つまりエンドポイント ID (EID) をその場所、つまりルーティング ロケータ (RLOC) から2つの異なるナンバリングスペースに分離し、サーバーの移行をクライアントに対して透過的にします。たとえば、サーバーが新しい場所に移動し、クライアントがサーバーにトラフィックを送信すると、ルータは新しい場所にトラフィックをリダイレクトします。

LISP では、LISP の出力トンネルルータ (ETR)、入力トンネルルータ (ITR)、ファーストホップルータ、マップリゾルバ (MR)、およびマップサーバ (MS) などのある一定のロールにおいてルータとサーバが必要です。サーバが別のルータに接続されていることをサーバのファーストホップルータが感知すると、そのルータは他のすべてのルータとデータベースを更新し、クライアントに接続されている ITR がトラフィックを代行受信してカプセル化し、新しいサーバの場所に送信できるようにします。

ASA LISP のサポート

ASA は LISP 自体を実行しませんが、場所の変更に関する LISP トラフィックを検査し、シームレスなクラスタリング操作のためにこの情報を使用できます。LISP の統合を行わない場合、サーバが新しいサイトに移動すると、トラフィックは元のフローオーナーの代わりに、新しいサイトで ASA クラスタメンバーになります。新しい ASA が古いサイトの ASA にトラフィックを転送した後、古い ASA は、サーバに到達するためにトラフィックを新しいサイトに送り返す必要があります。このトラフィックフローは最適ではなく、「トロンボーンング」または「ヘアピニング」と呼ばれます。

LISP 統合により、ASA クラスタメンバーは、最初のホップルータと ETR または ITR 間でやり取りされる LISP トラフィックを検査し、フローの所有者を新しいサイトに変更できます。

LISP のガイドライン

- ASA クラスタメンバーは、サイトのファーストホップルータと ITR または ETR の間に存在している必要があります。ASA クラスタ自体を拡張セグメントのファーストホップルータにすることはできません。
- 完全分散されたフローのみがサポートされます。一元化されたフロー、半分散されたフロー、または個々のノードに属しているフローは新しいオーナーには移動されません。半分散されたフローには SIP などのアプリケーションが含まれており、親フローとそのすべての子フローが同じ ASA によって所有されます。

- クラスタはレイヤ3および4のフロー状態を移動させるだけです。一部のアプリケーションデータが失われる可能性があります。
- 短時間のフローまたはビジネスに不可欠でないフローの場合、オーナーの移動は有用でない可能性があります。インスペクションポリシーを設定するときに、この機能でサポートされるトラフィックのタイプを制御できます。また、フロー モビリティを不可欠なトラフィックに制限する必要があります。

ASA LISP の実装

この機能には、複数の相互に関係する設定が含まれています（それらについてはすべてこの章で説明します）。

1. （任意）ホストまたはサーバ IP アドレスに基づく検査対象 EID の制限：ファースト ホップ ルータは、ASA クラスタが関与していないホストまたはネットワークに EID 通知メッセージを送信する場合があります。このため、クラスタに関連するサーバまたはネットワークのみに EID を制限できます。たとえば、クラスタが2つのサイトのみに関与しているが、LISP が3つのサイトで実行されている場合は、クラスタに関与している2つのサイトに対してのみ EID を含める必要があります。
2. LISP トラフィック インスペクション：ASA は、ファーストホップルータと ITR または ETR の間で送信される EID 通知メッセージにおいて、UDP ポート 4342 上の LISP トラフィックを検査します。ASA は、EID とサイト ID を関連付ける EID テーブルを保持します。たとえば、最初のホップルータの送信元 IP アドレスと ITR または ETR の宛先アドレスをもつ LISP トラフィックを検査する必要があります。LISP トラフィックにはディレクタが割り当てられておらず、LISP トラフィック自体はクラスタ状態の共有に参加しないことに注意してください。
3. 指定されたトラフィックでのフロー モビリティを有効にするサービス ポリシー：ビジネスクリティカルなトラフィックでフロー モビリティを有効にする必要があります。たとえば、フロー モビリティを、HTTPS トラフィックのみに制限したり、特定のサーバとの間でやり取りされるトラフィックのみに制限したりできます。
4. サイト ID：ASA は、各クラスタノードのサイト ID を使用して新しいオーナーを特定します。
5. フロー モビリティを有効にするクラスタレベルの設定：クラスタ レベルでもフロー モビリティを有効にする必要があります。このオン/オフの切り替えを使用することで、特定のクラスのトラフィックまたはアプリケーションに対してフロー モビリティを簡単に有効または無効にできます。

LISP インスペクションの設定

LISP のトラフィックを検査して、サーバーがサイト間を移動する時にフロー モビリティを有効にできます。

始める前に

- 制御ノードのブートストラップの設定 (18 ページ) および データノードのブートストラップの設定 (24 ページ) に従って、各クラスタユニットをサイトIDに割り当てます。
- LISP のトラフィックはデフォルトインспекショントラフィック クラスに含まれないため、この手順の一部として LISP のトラフィック用に別のクラスを設定する必要があります。

手順

ステップ 1 (任意) LISP インспекションマップを設定して、IP アドレスに基づいて検査済みの EID を制限し、LISP の事前共有キーを設定します。

- a) 拡張 ACL を作成します。宛先 IP アドレスのみが EID 組み込みアドレスと照合されます。

```
access list eid_acl_name extended permit ip source_address mask destination_address mask
```

IPv4 ACL および IPv6 ACL のどちらにも対応しています。厳密な **access-list extended** の構文については、コマンドリファレンスを参照してください。

- b) LISP インспекションマップを作成し、パラメータ モードに移行します。

```
policy-map type inspect lisp inspect_map_name  
parameters
```

- c) 作成した ACL を識別して、許可された EID を定義します。

```
allowed-eid access-list eid_acl_name
```

ファースト ホップ ルータまたは ITR/ETR は、ASA クラスタが関与していないホストまたはネットワークに EID 通知メッセージを送信することがあります。このため、クラスタに関連するサーバーまたはネットワークのみに EID を制限できます。たとえば、クラスタが 2 つのサイトのみに関与しているが、LISP が 3 つのサイトで実行されている場合は、クラスタに関与している 2 つのサイトに対してのみ EID を含める必要があります。

- d) 必要に応じて、事前共有キーを入力します。

```
validate-key key
```

例 :

```
ciscoasa(config)# access-list TRACKED_EID_LISP extended permit ip any 10.10.10.0  
255.255.255.0  
ciscoasa(config)# policy-map type inspect lisp LISP_EID_INSPECT  
ciscoasa(config-pmap)# parameters  
ciscoasa(config-pmap-p)# allowed-eid access-list TRACKED_EID_LISP  
ciscoasa(config-pmap-p)# validate-key MadMaxShinyandChrome
```

ステップ 2 ファースト ホップ ルータとポート 4342 の ITR または ETR の間の UDP トラフィック の LISP インспекションの設定。

- a) 拡張 ACL を設定して LISP のトラフィックを特定します。

```
access list inspect_acl_name extended permit udp source_address mask destination_address mask eq 4342
```

UDP ポート 4342 を指定する必要があります。IPv4 ACL および IPv6 ACL のどちらにも対応しています。厳密な **access-list extended** の構文については、コマンドリファレンスを参照してください。

- b) ACL のクラス マップを作成します。

```
class-map inspect_class_name
```

```
match access-list inspect_acl_name
```

- c) ポリシーマップ、クラスマップを指定し、オプションの LISP インспекションマップを使用してインспекションを有効化し、サービスポリシーをインターフェイスに適用します（新規であれば）。

```
policy-map policy_map_name
```

```
class inspect_class_name
```

```
inspect lisp [inspect_map_name]
```

```
service-policy policy_map_name {global | interface ifc_name}
```

既存のサービスポリシーある場合は、既存のポリシーマップ名を指定します。デフォルトで、ASA には **global_policy** と呼ばれるグローバルポリシーが含まれているため、グローバルポリシーの名前を指定します。ポリシーをグローバルに適用しない場合は、インターフェイスごとに1つのサービスポリシーを作成することもできます。LISP インспекションは、双方向にトラフィックに適用するため、送信元と宛先の両方のインターフェイスにサービスポリシーを適用する必要はありません。トラフィックが両方向のクラスマップに一致する場合、ポリシーマップを適用するインターフェイスに入るまたは存在するトラフィックのすべてが影響を受けます。

例：

```
ciscoasa(config)# access-list LISP_ACL extended permit udp host 192.168.50.89 host 192.168.10.8 eq 4342
ciscoasa(config)# class-map LISP_CLASS
ciscoasa(config-cmap)# match access-list LISP_ACL
ciscoasa(config-cmap)# policy-map INSIDE_POLICY
ciscoasa(config-pmap)# class LISP_CLASS
ciscoasa(config-pmap-c)# inspect lisp LISP_EID_INSPECT
ciscoasa(config)# service-policy INSIDE_POLICY interface inside
```

ASAは、ファーストホップルータと ITR または ETR の間で送信される EID 通知メッセージの LISP トラフィックを検査します。ASA は、EID とサイト ID を関連付ける EID テーブルを保持します。

ステップ 3 トラフィック クラスのフローモビリティを有効化します。

- a) 拡張 ACL を設定して、サーバーがサイトを変更するときに、最適なサイトに再割り当てするビジネスクリティカルなトラフィックを特定します。

```
access list flow_acl_name extended permit udp source_address mask destination_address mask eq port
```

IPv4 ACL および IPv6 ACL のどちらにも対応しています。厳密な **access-list extended** の構文については、コマンドリファレンスを参照してください。フローモビリティは、ビジネスクリティカルなトラフィックに対してイネーブルにする必要があります。たとえば、フローモビリティを HTTPS トラフィックのみ、または特定のサーバーへのトラフィックのみに制限できます。

- b) ACL のクラスマップを作成します。

```
class-map flow_map_name
match access-list flow_acl_name
```

- c) LISP インспекションを有効化した同じポリシーマップ、フロークラスマップを指定して、フローモビリティを有効にします。

```
policy-map policy_map_name
class flow_map_name
cluster flow-mobility lisp
```

例：

```
ciscoasa(config)# access-list IMPORTANT-FLOWS extended permit tcp any 10.10.10.0
255.255.255.0 eq https
ciscoasa(config)# class-map IMPORTANT-FLOWS-MAP
ciscoasa(config)# match access-list IMPORTANT-FLOWS
ciscoasa(config-cmap)# policy-map INSIDE_POLICY
ciscoasa(config-pmap)# class IMPORTANT-FLOWS-MAP
ciscoasa(config-pmap-c)# cluster flow-mobility lisp
```

- ステップ 4** クラスタグループコンフィギュレーションモードに移行し、クラスタのフローのモビリティを有効にします。

```
cluster group name
flow-mobility lisp
```

このオン/オフの切り替えにより、フローモビリティの有効化や無効化を簡単に行えます。

例

次に例を示します。

- EID を 10.10.10.0/24 ネットワーク上の EID に制限します。
- 192.168.50.89 (内部) にある LISP ルータと 192.168.10.8 (別の ASA インターフェイス上) にある ITR または ETR ルータの間の LISP トラフィック (UDP 4342) を検査します。
- HTTPS を使用して 10.10.10.0/24 のサーバーに送信されるすべての内部トラフィックに対してフローモビリティを有効にします。

- クラスタに対してフロー モビリティをイネーブルにします。

```
access-list TRACKED_EID_LISP extended permit ip any 10.10.10.0 255.255.255.0
policy-map type inspect lisp LISP_EID_INSPECT
  parameters
    allowed-eid access-list TRACKED_EID_LISP
    validate-key MadMaxShinyandChrome
!
access-list LISP_ACL extended permit udp host 192.168.50.89 host 192.168.10.8 eq 4342
class-map LISP_CLASS
  match access-list LISP_ACL
policy-map INSIDE_POLICY
  class LISP_CLASS
    inspect lisp LISP_EID_INSPECT
service-policy INSIDE_POLICY interface inside
!
access-list IMPORTANT-FLOWS extended permit tcp any 10.10.10.0 255.255.255.0 eq https
class-map IMPORTANT-FLOWS-MAP
  match access-list IMPORTANT-FLOWS
policy-map INSIDE_POLICY
  class IMPORTANT-FLOWS-MAP
    cluster flow-mobility lisp
!
cluster group cluster1
  flow-mobility lisp
```

クラスタノードの管理

クラスタを導入した後は、コンフィギュレーションを変更し、クラスタノードを管理できます。

非アクティブノードになる

クラスタの非アクティブなメンバーになるには、クラスタリングコンフィギュレーションは変更せずに、そのノード上でクラスタリングをディセーブルにします。



- (注) ASAが（手動で、またはヘルスチェックエラーにより）非アクティブになると、すべてのデータインターフェイスがシャットダウンされます。管理専用インターフェイスのみがトラフィックを送受信できます。トラフィックフローを再開させるには、クラスタリングを再びイネーブルにします。または、そのノードをクラスタから完全に削除します。管理インターフェイスは、そのノードがクラスタ IP プールから受け取った IP アドレスを使用して引き続き稼働状態となります。ただし、リロードしてもノードがクラスタ内でまだアクティブではない場合（クラスタリングが無効な状態で設定を保存した場合など）、管理インターフェイスは無効になります。それ以降のコンフィギュレーション作業には、コンソールポートを使用する必要があります。

手順

ステップ1 クラスタの設定モードを開始します。

cluster group name

例：

```
ciscoasa(config)# cluster group pod1
```

ステップ2 クラスタリングをディセーブルにします。

no enable

このノードが制御ノードであった場合は、新しい制御ノードの選定が実行され、別のメンバーが制御ノードになります。

クラスタコンフィギュレーションは維持されるので、後でクラスタリングを再度イネーブルにできます。

制御ノードからのデータノードの非アクティブ化

ログインしているノード以外のメンバを非アクティブにするには、次のステップを実行します。



- (注) ASAが非アクティブになると、すべてのデータインターフェイスがシャットダウンされます。管理専用インターフェイスのみがトラフィックを送受信できます。トラフィックフローを再開するには、クラスタリングを再度有効にします。管理インターフェイスは、そのノードがクラスタIPプールから受け取ったIPアドレスを使用して引き続き稼働状態となります。ただし、リロードしてもノードがクラスタ内でまだアクティブではない場合（クラスタリングが無効な状態で設定を保存した場合など）、管理インターフェイスは無効になります。それ以降のコンフィギュレーション作業には、コンソールポートを使用する必要があります。
-

手順

クラスタからノードを削除します。

cluster remove unit node_name

ブートストラップコンフィギュレーションは変更されず、制御ノードから最後に同期されたコンフィギュレーションもそのままであるので、コンフィギュレーションを失わずに後でそのノードを再度追加できます。制御ノードを削除するためにデータノードでこのコマンドを入力した場合は、新しい制御ノードが選定されます。

メンバ名を一覧表示するには、**cluster remove unit ?** と入力するか、**show cluster info** コマンドを入力します。

例：

```
ciscoasa(config)# cluster remove unit ?

Current active units in the cluster:
asa2

ciscoasa(config)# cluster remove unit asa2
WARNING: Clustering will be disabled on unit asa2. To bring it back
to the cluster please logon to that unit and re-enable clustering
```

クラスタへの再参加

ノードがクラスタから削除された場合（たとえば、障害が発生したインターフェイスの場合、またはメンバーを手動で非アクティブにした場合）は、クラスタに手動で再参加する必要があります。

手順

ステップ1 コンソールで、クラスタ コンフィギュレーション モードを開始します。

cluster group name

例：

```
ciscoasa(config)# cluster group pod1
```

ステップ2 クラスタリングをイネーブルにします。

enable

クラスタからの脱退

クラスタから完全に脱退するには、クラスタ ブートストラップ コンフィギュレーション全体を削除する必要があります。各ノードの現在のコンフィギュレーションは（アクティブユニットから同期されて）同じであるため、クラスタから脱退すると、クラスタリング前のコンフィギュレーションをバックアップから復元するか、IPアドレスの競合を避けるためコンフィギュレーションを消去して初めからやり直すことも必要になります。

手順

ステップ1 データノードの場合、クラスタリングを次のように無効化します。

cluster group *cluster_name* no enable

例：

```
ciscoasa(config)# cluster group cluster1
ciscoasa(cfg-cluster)# no enable
```

クラスタリングがデータノード上でイネーブルになっている間は、コンフィギュレーション変更を行うことはできません。

ステップ2 クラスタ コンフィギュレーションをクリアします。

clear configure cluster

ASAは、管理インターフェイスとクラスタ制御リンクを含むすべてのインターフェイスをシャットダウンします。

ステップ3 クラスタ インターフェイス モードをディセーブルにします。

no cluster interface-mode

モードはコンフィギュレーションには保存されないため、手動でリセットする必要があります。

ステップ4 バックアップコンフィギュレーションがある場合、実行コンフィギュレーションにバックアップコンフィギュレーションをコピーします。

copy *backup_cfg* running-config

例：

```
ciscoasa(config)# copy backup_cluster.cfg running-config

Source filename [backup_cluster.cfg]?

Destination filename [running-config]?
ciscoasa(config)#
```

ステップ5 コンフィギュレーションをスタートアップに保存します。

write memory

ステップ6 バックアップコンフィギュレーションがない場合は、管理アクセスを再設定します。たとえば、インターフェイス IP アドレスを変更し、正しいホスト名を復元します。

制御ノードの変更



注意 制御ノードを変更する最良の方法は、制御ノードでクラスタリングを無効にし、新しい制御ユニットの選択を待ってから、クラスタリングを再度有効にする方法です。制御ノードにするノードを厳密に指定する必要がある場合は、この項の手順を使用します。ただし、中央集中型機能の場合は、この手順を使用して制御ノード変更を強制するとすべての接続がドロップされるので、新しい制御ノード上で接続を再確立する必要があります。

制御ノードを変更するには、次の手順を実行します。

手順

新しいノードを制御ノードとして設定します。

cluster control-node unitnode_name

例：

```
ciscoasa(config)# cluster control-node unit asa2
```

メインクラスタ IP アドレスへの再接続が必要になります。

メンバー名を表示するには、**cluster control-node unit ?**（現在のノードを除くすべての名前が表示される）と入力するか、**show cluster info** コマンドを入力します。

クラスタ全体でのコマンドの実行

コマンドをクラスタ内のすべてのノードに、または特定のノードに送信するには、次の手順を実行します。**show** コマンドをすべてのノードに送信すると、すべての出力が収集されて現在のノードのコンソールに表示されます。その他のコマンド、たとえば **capture** や **copy** も、クラスタ全体での実行を活用できます。

手順

すべてのノードにコマンドを送信します。ノード名を指定した場合は、特定のノードに送信します。

cluster exec [unit node_name] コマンド

例：

```
ciscoasa# cluster exec show xlate
```

ノード名を表示するには、**cluster exec unit ?**（現在のノードを除くすべての名前が表示される）と入力するか、**show cluster info** コマンドを入力します。

例

同じキャプチャファイルをクラスタ内のすべてのノードから同時に TFTP サーバにコピーするには、制御ノードで次のコマンドを入力します。

```
ciscoasa# cluster exec copy /pcap capture: tftp://10.1.1.56/capture1.pcap
```

複数の PCAP ファイル（各ノードから 1 つずつ）が TFTP サーバにコピーされます。宛先のキャプチャファイル名には自動的にノード名が付加され、**capture1_asa1.pcap**、**capture1_asa2.pcap** などとなります。この例では、**asa1** と **asa2** はクラスタノード名です。

ASA 仮想クラスタのモニタリング

クラスタの状態と接続をモニターおよびトラブルシューティングできます。

クラスタ ステータスのモニタリング

クラスタの状態のモニタリングについては、次のコマンドを参照してください。

- **show cluster info [health [details]]**

キーワードを指定しないで **show cluster info** コマンドを実行すると、クラスタ内のすべてのメンバのステータスが表示されます。

show cluster info health コマンドは、インターフェイス、ノードおよびクラスタ全体の現在の状態を表示します。**details** キーワードは、ハートビート メッセージの失敗数を表示します。

show cluster info コマンドについては次の出力を参照してください。

```
ciscoasa# show cluster info
Cluster stbu: On
  This is "C" in state DATA_NODE
    ID      : 0
    Site ID : 1
    Version : 9.4(1)
    Serial No.: P3000000025
    CCL IP   : 10.0.0.3
    CCL MAC  : 000b.fcf8.c192
    Last join : 17:08:59 UTC Sep 26 2011
    Last leave: N/A
  Other members in the cluster:
    Unit "D" in state DATA_NODE
      ID      : 1
      Site ID : 1
```

```

        Version   : 9.4(1)
Serial No.: P3000000001
CCL IP    : 10.0.0.4
CCL MAC   : 000b.fcfc8.c162
Last join : 19:13:11 UTC Sep 23 2011
Last leave: N/A
Unit "A" in state CONTROL_NODE
  ID      : 2
  Site ID : 2
    Version   : 9.4(1)
Serial No.: JAB0815R0JY
CCL IP    : 10.0.0.1
CCL MAC   : 000f.f775.541e
Last join : 19:13:20 UTC Sep 23 2011
Last leave: N/A
Unit "B" in state DATA_NODE
  ID      : 3
  Site ID : 2
    Version   : 9.4(1)
Serial No.: P3000000191
CCL IP    : 10.0.0.2
CCL MAC   : 000b.fcfc8.c61e
Last join : 19:13:50 UTC Sep 23 2011
Last leave: 19:13:36 UTC Sep 23 2011

```

• show cluster info auto-join

時間遅延後にクラスタノードがクラスタに自動的に再参加するかどうか、および障害状態（ライセンスの待機やシャーシのヘルスチェック障害など）がクリアされたかどうかを示します。ノードが永続的に無効になっている場合、またはノードがすでにクラスタ内にある場合、このコマンドでは出力が表示されません。

show cluster info auto-join コマンドについては次の出力を参照してください。

```

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit will try to join cluster in 253 seconds.
Quit reason: Received control message DISABLE

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit will try to join cluster when quit reason is cleared.
Quit reason: Control node has application down that data node has up.

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit will try to join cluster when quit reason is cleared.
Quit reason: Chassis-blade health check failed.

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit will try to join cluster when quit reason is cleared.
Quit reason: Service chain application became down.

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit will try to join cluster when quit reason is cleared.
Quit reason: Unit is kicked out from cluster because of Application health check failure.

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit join is pending (waiting for the smart license entitlement: ent1)

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit join is pending (waiting for the smart license export control flag)

```

• **show cluster info transport {asp |cp[detail]}**

次のトランスポート関連の統計情報を表示します。

- **asp** : データ プレーンのトランスポート統計情報。
- **cp** : コントロールプレーンのトランスポート統計情報。

detail キーワードを入力すると、クラスタで信頼性の高いトランスポートプロトコルの使用状況が表示され、バッファがコントロールプレーンでいっぱいになったときにパケット ドロップの問題を特定できます。 **show cluster info transport cp detail** コマンドについては次の出力を参照してください。

```
ciscoasa# show cluster info transport cp detail
Member ID to name mapping:
  0 - unit-1-1   2 - unit-4-1   3 - unit-2-1
```

Legend:

```
U   - unreliable messages
UE  - unreliable messages error
SN  - sequence number
ESN - expecting sequence number
R   - reliable messages
RE  - reliable messages error
RDC - reliable message deliveries confirmed
RA  - reliable ack packets received
RFR - reliable fast retransmits
RTR - reliable timer-based retransmits
RDP - reliable message dropped
RDPR - reliable message drops reported
RI  - reliable message with old sequence number
RO  - reliable message with out of order sequence number
ROW - reliable message with out of window sequence number
ROB - out of order reliable messages buffered
RAS - reliable ack packets sent
```

This unit as a sender

```
-----
      all      0      2      3
U   123301  3867966  3230662  3850381
UE   0        0        0        0
SN  1656a4ce  acb26fe  5f839f76  7b680831
R   733840   1042168  852285   867311
RE   0        0        0        0
RDC 699789   934969   740874   756490
RA  385525   281198   204021   205384
RFR 27626    56397    0        0
RTR 34051   107199   111411   110821
RDP 0        0        0        0
RDPR 0      0        0        0
```

This unit as a receiver of broadcast messages

```
-----
      0      2      3
U   111847  121862  120029
R   7503   665700  749288
ESN 5d75b4b3  6d81d23  365ddd50
RI  630    34278   40291
RO  0      582     850
ROW 0      566     850
ROB 0      16      0
RAS 1571   123289  142256
```

This unit as a receiver of unicast messages

```
-----
      0          2          3
U      1          3308122  4370233
R     513846      879979   1009492
ESN   4458903a  6d841a84  7b4e7fa7
RI    66024      108924   102114
RO     0          0          0
ROW   0          0          0
ROB   0          0          0
RAS   130258     218924   228303
```

Gated Tx Buffered Message Statistics

```
-----
current sequence number: 0

total:                    0
current:                  0
high watermark:          0

delivered:                0
deliver failures:        0

buffer full drops:        0
message truncate drops:  0

gate close ref count:    0

num of supported clients:45
```

MRT Tx of broadcast messages

=====

Message high watermark: 3%
 Total messages buffered at high watermark: 5677
 [Per-client message usage at high watermark]

```
-----
Client name                Total messages  Percentage
Cluster Redirect Client    4153           73%
Route Cluster Client       419            7%
RRI Cluster Client         1105           19%
```

Current MRT buffer usage: 0%

Total messages buffered in real-time: 1
 [Per-client message usage in real-time]
 Legend:
 F - MRT messages sending when buffer is full
 L - MRT messages sending when cluster node leave
 R - MRT messages sending in Rx thread

```
-----
Client name                Total messages  Percentage  F  L  R
VPN Clustering HA Client    1             100%      0  0  0
```

MRT Tx of unitcast messages(to member_id:0)

=====

Message high watermark: 31%
 Total messages buffered at high watermark: 4059
 [Per-client message usage at high watermark]

```
-----
Client name                Total messages  Percentage
Cluster Redirect Client    3731           91%
RRI Cluster Client         328            8%
```

Current MRT buffer usage: 29%

```

Total messages buffered in real-time: 3924
[Per-client message usage in real-time]
Legend:
  F - MRT messages sending when buffer is full
  L - MRT messages sending when cluster node leave
  R - MRT messages sending in Rx thread
-----
Client name                Total messages  Percentage  F  L  R
Cluster Redirect Client    3607            91%        0  0  0
RRI Cluster Client         317             8%         0  0  0

MRT Tx of unitcast messages(to member_id:2)
=====
Message high watermark: 14%
Total messages buffered at high watermark: 578
[Per-client message usage at high watermark]
-----
Client name                Total messages  Percentage
VPN Clustering HA Client   578            100%

Current MRT buffer usage: 0%
Total messages buffered in real-time: 0

MRT Tx of unitcast messages(to member_id:3)
=====
Message high watermark: 12%
Total messages buffered at high watermark: 573
[Per-client message usage at high watermark]
-----
Client name                Total messages  Percentage
VPN Clustering HA Client   572            99%
Cluster VPN Unique ID Client 1                0%

Current MRT buffer usage: 0%
Total messages buffered in real-time: 0

```

• **show cluster history**

クラスタの履歴、およびクラスタノードが参加できなかった理由や、ノードがクラスタを離れた理由に関するエラーメッセージが表示されます。

クラスタ全体のパケットのキャプチャ

クラスタでのパケットのキャプチャについては、次のコマンドを参照してください。

cluster exec capture

クラスタ全体のトラブルシューティングをサポートするには、**cluster exec capture** コマンドを使用して制御ノード上でのクラスタ固有トラフィックのキャプチャを有効にします。これで、クラスタ内のすべてのデータノードでも自動的に有効になります。

クラスタリソースのモニタリング

クラスタリソースのモニタリングについては、次のコマンドを参照してください。

show cluster {cpu | memory | resource} [options]

クラスタ全体の集約データを表示します。使用可能な *options* はデータのタイプによって異なります。

クラスタ トラフィックのモニタリング

クラスタトラフィックのモニタリングについては、次のコマンドを参照してください。

- **show conn [detail]、cluster exec show conn**

show conn コマンドは、フローがディレクタ、バックアップ、またはフォワーダのどのフローであるかを示します。**cluster exec show conn** コマンドを任意のノードで使用すると、すべての接続が表示されます。このコマンドの表示からは、1つのフローのトラフィックがクラスタ内のさまざまなASAにどのように到達するかがわかります。クラスタのスループットは、ロードバランシングの効率とコンフィギュレーションによって異なります。このコマンドを利用すると、ある接続のトラフィックがクラスタ内をどのように流れるかが簡単にわかります。また、ロードバランサがフローのパフォーマンスにどのように影響を与えるかを理解するのに役立ちます。

また、**show conn detail** コマンドはフローモビリティの影響を受けるフローを表示します。

次に、**show conn detail** コマンドの出力例を示します。

```
ciscoasa/ASA2/data node# show conn detail
12 in use, 13 most used
Cluster stub connections: 0 in use, 46 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
      B - initial SYN from outside, b - TCP state-bypass or nailed,
      C - CTIQBE media, c - cluster centralized,
      D - DNS, d - dump, E - outside back connection, e - semi-distributed,
      F - outside FIN, f - inside FIN,
      G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
      i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
      k - Skinny media, L - LISP triggered flow owner mobility,
      M - SMTP data, m - SIP media, n - GUP
      O - outbound data, o - offloaded,
      P - inside back connection,
      Q - Diameter, q - SQL*Net data,
      R - outside acknowledged FIN,
      R - UDP SUNRPC, r - inside acknowledged FIN, S - awaiting inside SYN,
      s - awaiting outside SYN, T - SIP, t - SIP transient, U - up,
      V - VPN orphan, W - WAAS,
      w - secondary domain backup,
      X - inspected by service module,
      x - per session, Y - director stub flow, y - backup stub flow,
      Z - Scansafe redirection, z - forwarding stub flow
ESP outside: 10.1.227.1/53744 NP Identity Ifc: 10.1.226.1/30604, , flags c, idle 0s,
uptime
1m21s, timeout 30s, bytes 7544, cluster sent/rcvd bytes 0/0, owners (0,255) Traffic
received
at interface outside Locally received: 7544 (93 byte/s) Traffic received at interface
NP
Identity Ifc Locally received: 0 (0 byte/s) UDP outside: 10.1.227.1/500 NP Identity
Ifc:
10.1.226.1/500, flags -c, idle 1m22s, uptime 1m22s, timeout 2m0s, bytes 1580, cluster
sent/rcvd bytes 0/0, cluster sent/rcvd total bytes 0/0, owners (0,255) Traffic
received at
interface outside Locally received: 864 (10 byte/s) Traffic received at interface
NP Identity
```

```
Ifc Locally received: 716 (8 byte/s)
```

接続フローのトラブルシューティングを行うには、最初にすべてのノードの接続を一覧表示します。それには、任意のノードで **cluster exec show conn** コマンドを入力します。ディレクタ (Y)、バックアップ (y)、およびフォワーダ (z) のフラグを持つフローを探します。次の例には、3つのすべてのASAでの172.18.124.187:22から192.168.103.131:44727へのSSH接続が表示されています。ASA1にはzフラグがあり、この接続のフォワーダであることを表しています。ASA3にはYフラグがあり、この接続のディレクタであることを表しています。ASA2には特別なフラグはなく、これがオーナーであることを表しています。アウトバウンド方向では、この接続のパケットはASA2の内部インターフェイスに入り、外部インターフェイスから出ていきます。インバウンド方向では、この接続のパケットはASA1およびASA3の外部インターフェイスに入り、クラスタ制御リンクを介してASA2に転送され、次にASA2の内部インターフェイスから出ていきます。

```
ciscoasa/ASA1/control node# cluster exec show conn
ASA1 (LOCAL):*****
18 in use, 22 most used
Cluster stub connections: 0 in use, 5 most used
TCP outside 172.18.124.187:22 inside 192.168.103.131:44727, idle 0:00:00, bytes
37240828, flags z

ASA2:*****
12 in use, 13 most used
Cluster stub connections: 0 in use, 46 most used
TCP outside 172.18.124.187:22 inside 192.168.103.131:44727, idle 0:00:00, bytes
37240828, flags UIO

ASA3:*****
10 in use, 12 most used
Cluster stub connections: 2 in use, 29 most used
TCP outside 172.18.124.187:22 inside 192.168.103.131:44727, idle 0:00:03, bytes
0, flags Y
```

- **show cluster info [conn-distribution | packet-distribution | loadbalance | flow-mobility counters]**

show cluster info conn-distribution コマンドと **show cluster info packet-distribution** コマンドは、すべてのクラスタノードへのトラフィック分散を表示します。これらのコマンドは、外部ロードバランサを評価し、調整するのに役立ちます。

show cluster info loadbalance コマンドは、接続再分散の統計情報を表示します。

show cluster info flow-mobility counters コマンドは、EIDおよびフローの所有者の動作情報を表示します。**show cluster info flow-mobility counters** コマンドについては次の出力を参照してください。

```
ciscoasa# show cluster info flow-mobility counters
EID movement notification received : 4
EID movement notification processed : 4
Flow owner moving requested : 2
```

- **show cluster info load-monitor [details]**

この**show cluster info load-monitor**コマンドは、最後の間隔のクラスタメンバのトラフィック負荷と、設定された間隔の合計数（デフォルトでは30）を表示します。各間隔の各測定値を表示するには、**details** キーワードを使用します。

```
ciscoasa(cfg-cluster)# show cluster info load-monitor
ID Unit Name
0 B
1 A_1
Information from all units with 20 second interval:
Unit      Connections    Buffer Drops    Memory Used    CPU Used
Average from last 1 interval:
0          0              0              14             25
1          0              0              16             20
Average from last 30 interval:
0          0              0              12             28
1          0              0              13             27
```

```
ciscoasa(cfg-cluster)# show cluster info load-monitor details
```

```
ID Unit Name
0 B
1 A_1
Information from all units with 20 second interval
Connection count captured over 30 intervals:
Unit ID 0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
Unit ID 1
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
Buffer drops captured over 30 intervals:
Unit ID 0
```

```

0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
Unit ID 1
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0

```

Memory usage(%) captured over 30 intervals:

```

Unit ID 0
25     25     30     30     30     35
25     25     35     30     30     30
25     25     30     25     25     35
30     30     30     25     25     25
25     20     30     30     30     30
Unit ID 1
30     25     35     25     30     30
25     25     35     25     30     35
30     30     35     30     30     30
25     20     30     25     25     30
20     30     35     30     30     35

```

CPU usage(%) captured over 30 intervals:

```

Unit ID 0
25     25     30     30     30     35
25     25     35     30     30     30
25     25     30     25     25     35
30     30     30     25     25     25

```

	25	20	30	30	30	30
Unit ID 1						
	30	25	35	25	30	30
	25	25	35	25	30	35
	30	30	35	30	30	30
	25	20	30	25	25	30
	20	30	35	30	30	35

• **show cluster {access-list | conn | traffic | user-identity | xlate} [options]**

クラスタ全体の集約データを表示します。使用可能な *options* はデータのタイプによって異なります。

show cluster access-list コマンドについては次の出力を参照してください。

```
ciscoasa# show cluster access-list
hitcnt display order: cluster-wide aggregated result, unit-A, unit-B, unit-C, unit-D
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096) alert-interval
  300
access-list 101; 122 elements; name hash: 0xe7d586b5
access-list 101 line 1 extended permit tcp 192.168.143.0 255.255.255.0 any eq www
(hitcnt=0, 0, 0, 0, 0) 0x207a2b7d
access-list 101 line 2 extended permit tcp any 192.168.143.0 255.255.255.0 (hitcnt=0,
  0, 0, 0, 0) 0xfe4f4947
access-list 101 line 3 extended permit tcp host 192.168.1.183 host 192.168.43.238
(hitcnt=1, 0, 0, 0, 1) 0x7b521307
access-list 101 line 4 extended permit tcp host 192.168.1.116 host 192.168.43.238
(hitcnt=0, 0, 0, 0, 0) 0x5795c069
access-list 101 line 5 extended permit tcp host 192.168.1.177 host 192.168.43.238
(hitcnt=1, 0, 0, 1, 0) 0x51bde7ee
access-list 101 line 6 extended permit tcp host 192.168.1.177 host 192.168.43.13
(hitcnt=0, 0, 0, 0, 0) 0x1e68697c
access-list 101 line 7 extended permit tcp host 192.168.1.177 host 192.168.43.132
(hitcnt=2, 0, 0, 1, 1) 0xc1ce5c49
access-list 101 line 8 extended permit tcp host 192.168.1.177 host 192.168.43.192
(hitcnt=3, 0, 1, 1, 1) 0xb6f59512
access-list 101 line 9 extended permit tcp host 192.168.1.177 host 192.168.43.44
(hitcnt=0, 0, 0, 0, 0) 0xdc104200
access-list 101 line 10 extended permit tcp host 192.168.1.112 host 192.168.43.44
(hitcnt=429, 109, 107, 109, 104)
0xce4f281d
access-list 101 line 11 extended permit tcp host 192.168.1.170 host 192.168.43.238
(hitcnt=3, 1, 0, 0, 2) 0x4143a818
access-list 101 line 12 extended permit tcp host 192.168.1.170 host 192.168.43.169
(hitcnt=2, 0, 1, 0, 1) 0xb18dfea4
access-list 101 line 13 extended permit tcp host 192.168.1.170 host 192.168.43.229
(hitcnt=1, 1, 0, 0, 0) 0x21557d71
access-list 101 line 14 extended permit tcp host 192.168.1.170 host 192.168.43.106
(hitcnt=0, 0, 0, 0, 0) 0x7316e016
access-list 101 line 15 extended permit tcp host 192.168.1.170 host 192.168.43.196
(hitcnt=0, 0, 0, 0, 0) 0x013fd5b8
access-list 101 line 16 extended permit tcp host 192.168.1.170 host 192.168.43.75
(hitcnt=0, 0, 0, 0, 0) 0x2c7dba0d
```

使用中の接続の、すべてのノードでの合計数を表示するには、次のとおりに入力します。

```
ciscoasa# show cluster conn count
Usage Summary In Cluster:*****
  200 in use (cluster-wide aggregated)
  cl2 (LOCAL):*****
  100 in use, 100 most used

  cl1:*****
  100 in use, 100 most used
```

- **show asp cluster counter**

このコマンドは、データベースのトラブルシューティングに役立ちます。

クラスタのルーティングのモニタリング

クラスタのルーティングについては、次のコマンドを参照してください。

- **show route cluster**

- **debug route cluster**

クラスタのルーティング情報を表示します。

- **show lisp eid**

EIDs と サイト ID を示す ASA EID テーブルを表示します。

cluster exec show lisp eid コマンドからの、次の出力を参照してください。

```
ciscoasa# cluster exec show lisp eid
L1 (LOCAL):*****
  LISP EID      Site ID
  33.44.33.105  2
  33.44.33.201  2
  11.22.11.1    4
  11.22.11.2    4
L2:*****
  LISP EID      Site ID
  33.44.33.105  2
  33.44.33.201  2
  11.22.11.1    4
  11.22.11.2    4
```

- **show asp table classify domain inspect-lisp**

このコマンドは、トラブルシューティングに役立ちます。

クラスタリングのロギングの設定

クラスタリングのロギングの設定については、次のコマンドを参照してください。

- **logging device-id**

クラスタ内の各ノードは、syslogメッセージを個別に生成します。**logging device-id** コマンドを使用すると、同一または異なるデバイス ID 付きで syslog メッセージを生成することができ、クラスタ内の同一または異なるノードからのメッセージのように見せることができます。

クラスタのインターフェイスのモニタリング

クラスタのインターフェイスのモニタリングについては、次のコマンドを参照してください。

- **show cluster interface-mode**

クラスタ インターフェイスのモードを表示します。

クラスタリングのデバッグ

クラスタリングのデバッグについては、次のコマンドを参照してください。

- **debug cluster [ccp | datapath | fsm | general | hc | license | rpc | transport]**

クラスタリングのデバッグ メッセージを表示します。

- **debug cluster flow-mobility**

クラスタリング フロー モビリティ関連のイベントを表示します。

- **debug lisp eid-notify-intercept**

EID 通知メッセージ代行受信時のイベントを表示します。

- **show cluster info trace**

show cluster info trace コマンドは、トラブルシューティングのためのデバッグ情報を表示します。

show cluster info trace コマンドについては次の出力を参照してください。

```
ciscoasa# show cluster info trace
Feb 02 14:19:47.456 [DEBUG]Receive CCP message: CCP_MSG_LOAD_BALANCE
Feb 02 14:19:47.456 [DEBUG]Receive CCP message: CCP_MSG_LOAD_BALANCE
Feb 02 14:19:47.456 [DEBUG]Send CCP message to all: CCP_MSG_KEEPLIVE from 80-1 at
CONTROL_NODE
```

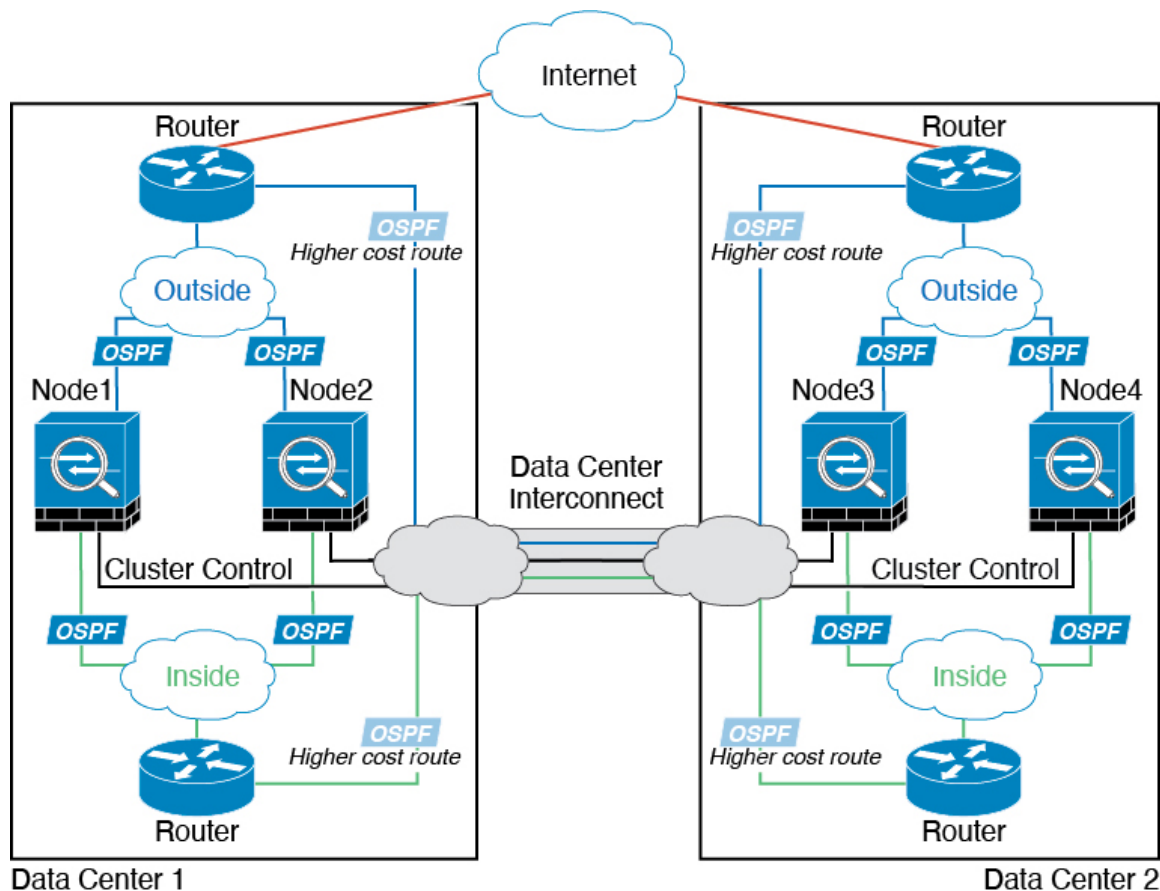
ASA 仮想クラスタリングの例

以下の例には、一般的な導入での ASA のクラスタ関連のすべてのコンフィギュレーションが含まれます。

個別インターフェイスルーテッドモード ノースサウス サイト間の例

次の例では、内部ルータと外部ルータの間に配置された（ノースサウス挿入）2つのデータセンターのそれぞれに2つの ASA クラスタノードがある場合を示します。クラスタノードは、DCI経由のクラスタ制御リンクによって接続されています。各データセンターの内部ルータと

外部ルータは、OSPFとPBRまたはECMPを使用してクラスタメンバー間でトラフィックをロードバランスします。DCIに高コストルートを割り当てることにより、特定のサイトのすべてのASA クラスタノードがダウンしない限り、トラフィックは各データセンター内に維持されます。1つのサイトのすべてのクラスタノードに障害が発生した場合、トラフィックは各ルータから DCI 経由で他のサイトの ASA クラスタノードに送られます。



クラスタリングの参考資料

このセクションには、クラスタリングの動作に関する詳細情報が含まれます。

ASA の各機能とクラスタリング

ASA の一部の機能は ASA クラスタリングではサポートされず、一部の機能は制御ノードだけでサポートされます。その他の機能については適切な使用に関する警告があります。

クラスタリングでサポートされない機能

次の各機能は、クラスタリングが有効なときは設定できず、コマンドは拒否されます。

- TLS プロキシを使用するユニファイド コミュニケーション機能
- リモート アクセス VPN (SSL VPN および IPsec VPN)
- 仮想トンネルインターフェイス (VTI)
- 次のアプリケーション インспекション :
 - CTIQBE
 - H323、H225、および RAS
 - IPsec パススルー
 - MGCP
 - MMP
 - RTSP
 - SCCP (Skinny)
 - WAAS
 - WCCP
- ボットネット トラフィック フィルタ
- Auto Update Server
- DHCP クライアント、サーバー、およびプロキシ。DHCP リレーはサポートされていません。
- VPN ロード バランシング
- フェールオーバー
- 統合ルーティングおよびブリッジング
- FIPS モード

クラスタリングの中央集中型機能

次の機能は、制御ノード上だけでサポートされます。クラスタの場合もスケーリングされません。



(注) 中央集中型機能のトラフィックは、クラスタ制御リンク経由でメンバーノードから制御ノードに転送されます。

再分散機能を使用する場合は、中央集中型機能のトラフィックが中央集中型機能として分類される前に再分散が行われて、制御ノード以外のノードに転送されることがあります。この場合は、トラフィックが制御ノードに送り返されます。

中央集中型機能については、制御ノードで障害が発生するとすべての接続がドロップされるので、新しい制御ノード上で接続を再確立する必要があります。

• 次のアプリケーション インспекション :

- DCERPC
- ESMTP
- IM
- NetBIOS
- PPTP
- RADIUS
- RSH
- SNMP
- SQLNET
- SUNRPC
- TFTP
- XDMCP

- スタティック ルート モニタリング
- ネットワーク アクセスの認証および許可。アカウントは非集中型です。
- フィルタリング サービス
- サイト間 VPN
- マルチキャストルーティング

個々のノードに適用される機能

これらの機能は、クラスタ全体または制御ノードではなく、各 ASA ノードに適用されます。

- QoS : QoS ポリシーは、コンフィギュレーション複製の一部としてクラスタ全体で同期されます。ただし、ポリシーは各ノードに個別に適用されます。たとえば、出力に対してポーリングを設定する場合は、適合レートおよび適合バースト値は、特定の ASA から出て

行くトラフィックに適用されます。3 ノードから成るクラスタがあり、トラフィックが均等に分散している場合、適合レートは実際にクラスタのレートの3 倍になります。

- 脅威検出：脅威検出はノードごとに個別に機能します。たとえば、上位統計情報はノード固有です。たとえば、ポートスキャン検出が機能しないのは、スキャントラフィックが全ノード間でロードバランシングされ、1 つのノードですべてのトラフィックを確認できないためです。
- リソース管理：マルチコンテキストモードでのリソース管理は、ローカル使用状況に基づいて各ノードに個別に適用されます。
- LISP トラフィック：UDP ポート 4342 上の LISP トラフィックは、各受信ノードによって検査されますが、ディレクタは割り当てられません。各ノードは、クラスタ間で共有される EID テーブルに追加されますが、LISP トラフィック自体はクラスタ状態の共有に参加しません。

ネットワーク アクセス用の AAA とクラスタリング

ネットワーク アクセス用の AAA は、認証、許可、アカウントिंगの3 つのコンポーネントで構成されます。認証と許可は、クラスタリング制御ノード上で中央集中型機能として実装されており、データ構造がクラスタデータノードに複製されます。制御ノードが選択された場合、確立済みの認証済みユーザーおよびユーザーに関連付けられた許可を引き続き中断なく運用するために必要なすべての情報を新しい制御ノードが保有します。ユーザー認証のアイドルおよび絶対タイムアウトは、制御ノードが変更されたときも維持されます。

アカウントINGは、クラスタ内の分散型機能として実装されています。アカウントINGはフロー単位で実行されるため、フローに対するアカウントINGが設定されている場合、そのフローを所有するクラスタノードがアカウントING開始と停止のメッセージをAAA サーバーに送信します。

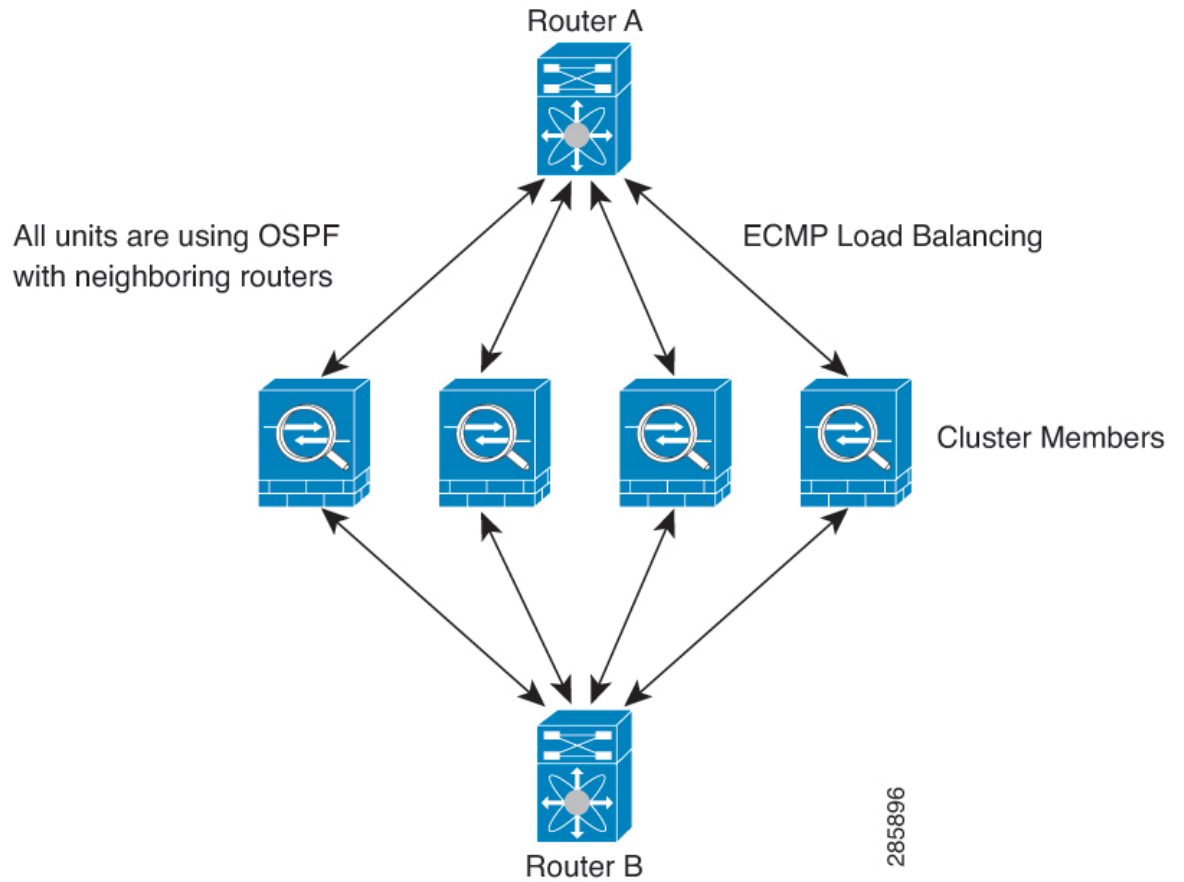
接続設定とクラスタリング

接続制限は、クラスタ全体に適用されます (**set connection conn-max**、**set connection embryonic-conn-max**、**set connection per-client-embryonic-max** および **set connection per-client-max** コマンドページを参照)。各ノードには、ブロードキャストメッセージに基づくクラスタ全体のカウンタの推定値があります。クラスタ全体で接続制限を設定しても、効率性を考慮して、厳密に制限数で適用されない場合があります。各ノードでは、任意の時点でのクラスタ全体のカウンタ値が過大評価または過小評価される可能性があります。ただし、ロードバランシングされたクラスタでは、時間の経過とともに情報が更新されます。

ダイナミック ルーティングおよびクラスタリング

個別インターフェイスモードでは、各ノードがスタンドアロンルータとしてルーティングプロトコルを実行します。ルートの学習は、各ノードが個別に行います。

図 1: 個別インターフェイス モードでのダイナミック ルーティング



上の図では、ルータ A はルータ B への等コストパスが 4 本あることを学習します。パスはそれぞれ 1 つの ASA を通過します。ECMP を使用して、4 パス間でトラフィックのロードバランシングを行います。各 ASA は、外部ルータと通信するときに、それぞれ異なるルータ ID を選択します。

管理者は、各ノードに異なるルータ ID が設定されるように、ルータ ID のクラスタプールを設定する必要があります。

EIGRP は、個別のインターフェイスモードのクラスタピアとのネイバー関係を形成しません。



- (注) 冗長性確保のためにクラスタが同一ルータに対して複数の隣接関係を持つ場合、非対称ルーティングが原因で許容できないトラフィック損失が発生する場合があります。非対称ルーティングを避けるためには、同じトラフィックゾーンにこれらすべての ASA インターフェイスをまとめます。

FTP とクラスタリング

- FTPDチャネルとコントロールチャネルのフローがそれぞれ別のクラスタメンバーによって所有されている場合は、Dチャネルのオーナーは定期的にアイドルタイムアウトアップデートをコントロールチャネルのオーナーに送信し、アイドルタイムアウト値を更新します。ただし、コントロールフローのオーナーがリロードされて、コントロールフローが再ホスティングされた場合は、親子フロー関係は維持されなくなります。したがって、コントロールフローのアイドルタイムアウトは更新されません。
- FTPアクセスにAAAを使用する場合、制御チャネルのフローは制御ノードに集中されません。

ICMP インспекションとクラスタリング

クラスタを通過する ICMP および ICMP エラーパケットのフローは、ICMP/ICMP エラーインспекションが有効かどうかによって異なります。ICMP インспекションを使用しない場合、ICMP は一方向のフローであり、ディレクタフローはサポートされません。ICMP インспекションを使用する場合、ICMP フローは双方向になり、ディレクタ/バックアップフローによってバックアップされます。検査された ICMP フローの違いの1つは、転送されたパケットのディレクタ処理にあります。ディレクタは、パケットをフォワーダに返す代わりに、フローオーナーに ICMP エコー応答パケットを転送します。

マルチキャストルーティングとクラスタリング

個別インターフェイスモードでは、マルチキャストに関してユニットが個別に動作することはありません。データおよびルーティングのパケットはすべて制御ユニットで処理されて転送されるので、パケットレプリケーションが回避されます。

NAT とクラスタリング

NAT は、クラスタの全体的なスループットに影響を与えることがあります。インバウンドおよびアウトバウンドの NAT パケットが、それぞれクラスタ内の別の ASA に送信されることがあります。ロードバランシングアルゴリズムは IP アドレスとポートに依存していますが、NAT が使用される場合は、インバウンドとアウトバウンドとで、パケットの IP アドレスやポートが異なるからです。NAT オーナーではない ASA に到着したパケットは、クラスタ制御リンクを介してオーナーに転送されるため、クラスタ制御リンクに大量のトラフィックが発生します。NAT オーナーは、セキュリティおよびポリシーチェックの結果に応じてパケットの接続を作成できない可能性があるため、受信側ノードは、オーナーへの転送フローを作成しないことに注意してください。

それでもクラスタリングで NAT を使用する場合は、次のガイドラインを考慮してください。

- プロキシ ARP なし：個別インターフェイスの場合は、マッピングアドレスについてプロキシ ARP 応答が送信されることはありません。これは、クラスタに存在しなくなった可能性のある ASA と隣接ルータとがピア関係を維持することを防ぐためです。アップストリームルータは、メインクラスタ IP アドレスを指すマッピングアドレスについてはステティックルートまたは PBR とオブジェクトトラッキングを使用する必要があります。こ

これは、スパンドEtherChannelの問題ではありません。クラスタインターフェイスには関連付けられたIPアドレスが1つしかないためです。

- 個別インターフェイスのインターフェイス PAT なし：インターフェイス PAT は、個別インターフェイスではサポートされていません。
- ポートブロック割り当てによる PAT：この機能については、次のガイドラインを参照してください。
 - ホストあたりの最大制限は、クラスタ全体の制限ではなく、ノードごとに個別に適用されます。したがって、ホストあたりの最大制限が1に設定されている3ノードクラスタでは、ホストからのトラフィックが3つのノードすべてにロードバランシングされている場合、3つのブロックを各ノードに1つずつ割り当てることができます。
 - バックアッププールからバックアップノードで作成されたポートブロックは、ホストあたりの最大制限の適用時には考慮されません。
 - PAT プールが完全に新しいIPアドレスの範囲で変更される On-the-fly PAT ルールの変更では、新しいプールが有効になっていてもいまだ送信中の xlate バックアップ要求に対する xlate バックアップの作成が失敗します。この動作はポートのブロック割り当て機能に固有なものではなく、プールが分散されトラフィックがクラスタノード間でロードバランシングされるクラスタ展開でのみ見られる一時的な PAT プールの問題です。
 - クラスタで動作している場合、ブロック割り当てサイズを変更することはできません。新しいサイズは、クラスタ内の各デバイスをリロードした後にのみ有効になります。各デバイスのリロードの必要性を回避するために、すべてのブロック割り当てルールを削除し、それらのルールに関連するすべての xlate をクリアすることをお勧めします。その後、ブロックサイズを変更し、ブロック割り当てルールを再作成できます。
- ダイナミック PAT の NAT プールアドレス配布：PAT プールを設定すると、クラスタはプール内の各IPアドレスをポートブロックに分割します。デフォルトでは、各ブロックは512ポートですが、ポートブロック割り当てルールを設定すると、代わりにユーザのブロック設定が使用されます。これらのブロックはクラスタ内のノード間で均等に分散されるため、各ノードには PAT プール内のIPアドレスごとに1つ以上のブロックがあります。したがって、想定される PAT 接続数に対して十分である場合には、クラスタの PAT プールに含めるIPアドレスを1つだけにすることができます。PAT プールの NAT ルールで予約済みポート1～1023を含めるようにオプションを設定しない限り、ポートブロックは1024～65535のポート範囲をカバーします。
- 複数のルールにおける PAT プールの再利用：複数のルールで同じ PAT プールを使用するには、ルールにおけるインターフェイスの選択に注意を払う必要があります。すべてのルールで特定のインターフェイスを使用するか、あるいはすべてのルールで「任意の」インターフェイスを使用するか、いずれかを選択する必要があります。ルール全般にわたって特定のインターフェイスと「任意」のインターフェイスを混在させることはできません。混在させると、システムがリターントラフィックとクラスタ内の適切なノードを一致

させることができなくなる場合があります。ルールごとに固有の PAT プールを使用することは、最も信頼性の高いオプションです。

- ラウンドロビンなし：PATプールのラウンドロビンは、クラスタリングではサポートされません。
- 拡張 PAT なし：拡張 PAT はクラスタリングでサポートされません。
- 制御ノードによって管理されるダイナミック NAT xlate：制御ノードが xlate テーブルを維持し、データノードに複製します。ダイナミック NAT を必要とする接続をデータノードが受信したときに、その xlate がテーブル内がない場合、データノードは制御ノードに xlate を要求します。データノードが接続を所有します。
- 旧式の xlates：接続所有者の xlate アイドル時間が更新されません。したがって、アイドル時間がアイドルタイムアウトを超える可能性があります。refcnt が 0 で、アイドルタイマー値が設定されたタイムアウトより大きい場合は、旧式の xlate であることを示します。
- per-session PAT 機能：クラスタリングに限りませんが、per-session PAT 機能によって PAT の拡張性が向上します。クラスタリングの場合は、各データノードが独自の PAT 接続を持てます。対照的に、multi-session PAT 接続は制御ノードに転送する必要があり、制御ノードがオーナーとなります。デフォルトでは、すべての TCP トラフィックおよび UDP DNS トラフィックは per-session PAT xlate を使用します。これに対し、ICMP および他のすべての UDP トラフィックは multi-session を使用します。TCP および UDP に対しこれらのデフォルトを変更するように per-session NAT ルールを設定できますが、ICMP に per-session PAT を設定することはできません。H.323、SIP、または Skinny などの multi-session PAT のメリットを活用できるトラフィックでは、関連付けられている TCP ポートに対し per-session PAT を無効にできます（それらの H.323 および SIP の UDP ポートはデフォルトですでに multi-session になっています）。per-session PAT の詳細については、ファイアウォールの設定ガイドを参照してください。
- 次のインスペクション用のスタティック PAT はありません。
 - FTP
 - PPTP
 - RSH
 - SQLNET
 - TFTP
 - XDMCP
 - SIP
- 1 万を超える非常に多くの NAT ルールがある場合は、デバイスの CLI で **asp rule-engine transactional-commit nat** コマンドを使用してトランザクションコミット モデルを有効にする必要があります。有効にしないと、ノードがクラスタに参加できない可能性があります。

SCTP とクラスタリング

SCTP アソシエーションは、（ロードバランシングにより）任意のノードに作成できますが、マルチホーミング接続は同じノードに存在する必要があります。

SIP インспекションとクラスタリング

制御フローは、（ロードバランシングにより）任意のノードに作成できますが、子データフローは同じノードに存在する必要があります。

TLS プロキシ設定はサポートされていません。

SNMP とクラスタリング

SNMP エージェントは、個々の ASA を、その 診断インターフェイスのローカル IP アドレスによってポーリングします。クラスタの統合データをポーリングすることはできません。

SNMP ポーリングには、メインクラスタ IP アドレスではなく、常にローカルアドレスを使用してください。SNMP エージェントがメインクラスタ IP アドレスをポーリングする場合、新しい制御ノードが選択されると、新しい制御ノードのポーリングは失敗します。

クラスタリングで SNMPv3 を使用している場合、最初のクラスタ形成後に新しいクラスタノードを追加すると、SNMPv3 ユーザーは新しいノードに複製されません。SNMPv3 ユーザーは、制御ノードに再追加して、新しいノードに強制的に複製するようにするか、データノードに直接追加する必要があります。

STUN とクラスタリング

ピンホールが複製される時、STUN インспекションはフェールオーバーモードとクラスタモードでサポートされます。ただし、トランザクション ID はノード間で複製されません。STUN 要求の受信後にノードに障害が発生し、別のノードが STUN 応答を受信した場合、STUN 応答はドロップされます。

syslog および NetFlow とクラスタリング

- **Syslog** : クラスタの各ノードは自身の syslog メッセージを生成します。ロギングを設定して、各ノードの syslog メッセージヘッダーフィールドで同じデバイス ID を使用するか、別の ID を使用するかを設定できます。たとえば、ホスト名設定はクラスタ内のすべてのノードに複製されて共有されます。ホスト名をデバイス ID として使用するようにロギングを設定した場合、すべてのノードで生成される syslog メッセージが1つのノードから生成されているように見えます。クラスタブートストラップ設定で割り当てられたローカルノード名をデバイス ID として使用するようにロギングを設定した場合、syslog メッセージはそれぞれ別のノードから生成されているように見えます。
- **NetFlow** : クラスタの各ノードは自身の NetFlow ストリームを生成します。NetFlow コレクタは、各 ASA を独立した NetFlow エクスポートとしてのみ扱うことができます。

Cisco TrustSec とクラスタリング

制御ノードだけがセキュリティグループタグ (SGT) 情報を学習します。その後、制御ノードからデータノードに SGT が渡されるため、データノードは、セキュリティポリシーに基づいて SGT の一致を判断できます。

VPN とクラスタリング

サイト間 VPN は、中央集中型機能です。制御ノードのみが VPN 接続をサポートします。



(注) リモート アクセス VPN は、クラスタリングではサポートされません。

VPN 機能を使用できるのは制御ノードだけであり、クラスタの高可用性機能は活用されません。制御ノードで障害が発生した場合は、すべての既存の VPN 接続が失われ、VPN ユーザにとってはサービスの中断となります。新しい制御ノードが選定されたときに、VPN 接続を再確立する必要があります。

PBR または ECMP を使用するときの個別インターフェイスへの接続については、ローカルアドレスではなく、常にメインクラスタ IP アドレスに接続する必要があります。

VPN 関連のキーと証明書は、すべてのノードに複製されます。

パフォーマンス スケーリング係数

複数のユニットをクラスタに結合すると、期待できる合計クラスタパフォーマンスは、最大合計スループットの約 80% になります。

たとえば、モデルが単独稼働で約 10 Gbps のトラフィックを処理できる場合、8 ユニットのクラスタでは、最大合計スループットは 80 Gbps (8 ユニット x 10 Gbps) の約 80% で 64 Gbps になります。

制御ノードの選定

クラスタのノードは、クラスタ制御リンクを介して通信して制御ノードを選定します。方法は次のとおりです。

1. ノードに対してクラスタリングをイネーブルにしたとき（または、クラスタリングがイネーブル済みの状態でそのユニットを初めて起動したとき）に、そのノードは選定要求を 3 秒間隔でブロードキャストします。
2. プライオリティの高い他のノードがこの選定要求に応答します。プライオリティは 1 ~ 100 の範囲内で設定され、1 が最高のプライオリティです。
3. 45 秒経過しても、プライオリティの高い他のノードからの応答を受信していない場合は、そのノードが制御ノードになります。



(注) 最高のプライオリティを持つノードが複数ある場合は、クラスタノード名、次にシリアル番号を使用して制御ノードが決定されます。

4. 後からクラスタに参加したノードのプライオリティの方が高い場合でも、そのノードが自動的に制御ノードになることはありません。既存の制御ノードは常に制御ノードのままです。ただし、制御ノードが応答を停止すると、その時点で新しい制御ノードが選定されます。
5. 「スプリットブレイン」シナリオで一時的に複数の制御ノードが存在する場合、優先順位が最も高いノードが制御ノードの役割を保持し、他のノードはデータノードの役割に戻ります。



(注) ノードを手動で強制的に制御ノードにすることができます。中央集中型機能については、制御ノード変更を強制するとすべての接続がドロップされるので、新しい制御ノード上で接続を再確立する必要があります。

ASA 仮想クラスタ内のハイアベイラビリティ

ASA 仮想クラスタリングは、ノードとインターフェイスの正常性をモニタリングし、ノード間で接続状態を複製することにより、ハイアベイラビリティを実現します。

ノードヘルスマニタリング

各ノードは、クラスタ制御リンクを介してブロードキャストハートビートパケットを定期的に送信します。設定可能なタイムアウト期間内にデータノードからハートビートパケットまたはその他のパケットを受信しない場合、制御ノードはクラスタからデータノードを削除します。データノードが制御ノードからパケットを受信しない場合、残りのノードから新しい制御ノードが選択されます。

ノードで実際に障害が発生したためではなく、ネットワークの障害が原因で、ノードがクラスタ制御リンクを介して相互に通信できない場合、クラスタは「スプリットブレイン」シナリオに移行する可能性があります。このシナリオでは、分離されたデータノードが独自の制御ノードを選択します。たとえば、2つのクラスタロケーション間でルータに障害が発生した場合、ロケーション1の元の制御ノードは、ロケーション2のデータノードをクラスタから削除します。一方、ロケーション2のノードは、独自の制御ノードを選択し、独自のクラスタを形成します。このシナリオでは、非対称トラフィックが失敗する可能性があることに注意してください。クラスタ制御リンクが復元されると、より優先順位の高い制御ノードが制御ノードの役割を保持します。

詳細については、[制御ノードの選定 \(65 ページ\)](#) を参照してください。

インターフェイス モニタリング

各ノードは、使用中のすべての指名されたハードウェアインターフェイスのリンクステータスをモニタし、ステータス変更を制御ノードに報告します。

ヘルスマニタリングを有効化すると、すべての物理インターフェイスがデフォルトでモニターされるため、オプションでインターフェイスごとのモニタリングを無効化することができます。指名されたインターフェイスのみモニターできます。

ノードのモニタ対象のインターフェイスが失敗した場合、そのノードはクラスタから削除されます。ASAがメンバーをクラスタから削除するまでの時間は、そのノードが確立済みメンバーであるか、またはクラスタに参加しようとしているかによって異なります。ASAは、ノードがクラスタに参加する最初の90秒間はインターフェイスを監視しません。この間にインターフェイスのステータスが変化しても、ASAはクラスタから削除されません。ノード状態に関係なく、ノードは500ミリ秒後に削除されます。

障害後のステータス

クラスタ内のノードで障害が発生したときに、そのノードでホストされている接続は他のノードにシームレスに移行されます。トラフィックフローのステート情報は、制御ノードのクラスタ制御リンクを介して共有されます。

制御ノードで障害が発生した場合、そのクラスタの他のメンバーのうち、優先順位が最高（番号が最小）のメンバーが制御ノードになります。

障害イベントに応じて、ASAは自動的にクラスタへの再参加を試みます。



- (注) ASAが非アクティブになり、クラスタへの自動再参加に失敗すると、すべてのデータインターフェイスがシャットダウンされ、管理専用インターフェイスのみがトラフィックを送受信できます。管理インターフェイスは、そのノードがクラスタIPプールから受け取ったIPアドレスを使用して引き続き稼働状態となります。ただし、リロードする場合、クラスタでノードがまだ非アクティブになっていると、管理インターフェイスは無効になります。さらに設定を行う場合は、コンソールポートを使用する必要があります。

クラスタへの再参加

クラスタノードがクラスタから削除された後、クラスタに再参加するための方法は、削除された理由によって異なります。

- クラスタ制御リンクの障害：（最初の参加時）クラスタ制御リンクの問題を解決した後、CLIで **cluster group** 名を入力してから **enable** と入力して、クラスタリングを再び有効化することによって、手動でクラスタに再参加する必要があります。
- クラスタに参加した後に障害が発生したクラスタ制御リンク：ASAは、無限に5分ごとに自動的に再参加を試みます。この動作は設定可能です。
- データインターフェイスの障害：ASAは自動的に最初は5分後、次に10分後、最終的に20分後に再参加を試みます。20分後に参加できない場合、ASAはクラスタリングをディ

セーブにします。データインターフェイスの問題を解決した後、CLIで **cluster group name** と入力してから **enable** と入力して、クラスタリングを手動で有効化する必要があります。この動作は設定可能です。

- ノードの障害：ノードがヘルスチェック失敗のためクラスタから削除された場合、クラスタへの再参加は失敗の原因によって異なります。たとえば、一時的な電源障害の場合は、クラスタ制御リンクが稼働していて、クラスタリングが **enable** コマンドでまだイネーブルになっているなら、ノードは再起動するとクラスタに再参加することを意味します。ASA は 5 秒ごとにクラスタへの再参加を試みます。
- 内部エラー：内部の障害には、アプリケーション同期のタイムアウト、矛盾したアプリケーション ステータスなどがあります。ノードは、5 分、10 分、20 分の間隔で自動的にクラスタに再参加しようとしています。この動作は設定可能です。

データ パス接続状態の複製

どの接続にも、1つのオーナーおよび少なくとも1つのバックアップオーナーがクラスタ内にあります。バックアップオーナーは、障害が発生しても接続を引き継ぎません。代わりに、TCP/UDP のステート情報を保存します。これは、障害発生時に接続が新しいオーナーにシームレスに移管されるようにするためです。バックアップオーナーは通常ディレクタでもありません。

トラフィックの中には、TCP または UDP レイヤよりも上のステート情報を必要とするものがあります。この種類のトラフィックに対するクラスタリングのサポートの可否については、次の表を参照してください。

表 1:クラスタ全体で複製される機能

トラフィック	状態のサポート	注
アップタイム	○	システムアップタイムをトラッキングします。
ARP テーブル	あり	—
MAC アドレス テーブル	あり	—
ユーザ アイデンティティ	○	AAA ルール (uauth) が含まれます。
IPv6 ネイバー データベース	あり	—
ダイナミック ルーティング	あり	—
SNMP エンジン ID	なし	—
Firepower 4100/9300 の分散型 VPN (サイト間)	Yes	バックアップセッションがアクティブセッションになると、新しいバックアップセッションが作成されます。

ASA 仮想クラスタが接続を管理する方法

接続をクラスタの複数のノードにロードバランシングできます。接続のルールにより、通常動作時とハイ アベイラビリティ状況時の接続の処理方法が決まります。

接続のルール

接続ごとに定義された次のルールを参照してください。

- **オーナー**：通常、最初に接続を受信するノード。オーナーは、TCP 状態を保持し、パケットを処理します。1 つの接続に対してオーナーは 1 つだけです。元のオーナーに障害が発生すると、新しいノードが接続からパケットを受信したときにディレクタがそれらのノードの新しいオーナーを選択します。
- **バックアップオーナー**：オーナーから受信した TCP/UDP ステート情報を格納するノード。障害が発生した場合、新しいオーナーにシームレスに接続を転送できます。バックアップオーナーは、障害発生時に接続を引き継ぎません。オーナーが使用不可能になった場合、（ロードバランシングに基づき）その接続からのパケットを受信する最初のノードがバックアップオーナーに問い合わせ、関連するステート情報を取得し、そのノードが新しいオーナーになります。

ディレクタ（下記参照）がオーナーと同じノードでない限り、ディレクタはバックアップオーナーでもあります。オーナーが自分をディレクタとして選択した場合は、別のバックアップオーナーが選択されます。

1 台のシャーシに最大 3 つのクラスタノードを搭載できる Firepower 9300 のクラスタリングでは、バックアップオーナーがオーナーと同じシャーシにある場合、シャーシ障害からフローを保護するために、別のシャーシから追加のバックアップオーナーが選択されません。

サイト間クラスタリングのディレクタローカリゼーションを有効にすると、ローカルバックアップとグローバルバックアップの 2 つのバックアップオーナー権限があります。オーナーは、常に同じサイトのローカルバックアップをオーナー自身として選択します（サイト ID に基づいて）。グローバルバックアップはどのサイトにも配置でき、ローカルバックアップと同一ノードとすることもできます。オーナーは、両方のバックアップへ接続ステート情報を送信します。

サイトの冗長性が有効になっており、バックアップオーナーがオーナーと同じサイトに配置されている場合は、サイトの障害からフローを保護するために、別のサイトから追加のバックアップオーナーが選択されます。シャーシバックアップとサイトバックアップは独立しているため、フローにはシャーシバックアップとサイトバックアップの両方が含まれている場合があります。

- **ディレクタ**：フォワーダからのオーナールックアップ要求を処理するノード。オーナーは、新しい接続を受信すると、送信元/宛先 IP アドレスおよびポートのハッシュに基づいてディレクタを選択し、新しい接続を登録するためにそのディレクタにメッセージを送信します。パケットがオーナー以外のノードに到着した場合、そのノードはどのノードがオーナーかをディレクタに問い合わせることで、パケットを転送できます。1 つの接続に対してディレクタは 1 つだけです。ディレクタが失敗すると、オーナーは新しいディレクタを選択します。

ディレクタがオーナーと同じノードでない限り、ディレクタはバックアップオーナーでもあります（上記参照）。オーナーがディレクタとして自分自身を選択すると、別のバックアップオーナーが選択されます。

サイト間クラスタリングのディレクタローカリゼーションを有効にすると、ローカルディレクタとグローバルディレクタの2つのディレクタ権限が区別されます。オーナーは、同一サイト（SiteIdに基づき）のローカルディレクタとして、常にオーナー自身を選択します。グローバルディレクタはどのサイトにも配置でき、ローカルディレクタと同一ノードとすることもできます。最初のオーナーに障害が発生すると、ローカルディレクタは、同じサイトの新しい接続オーナーを選択します。

ICMP/ICMPv6 ハッシュの詳細：

- エコーパケットの場合、送信元ポートは ICMP 識別子で、宛先ポートは 0 です。
 - 応答パケットの場合、送信元ポートは 0 で、宛先ポートは ICMP 識別子です。
 - 他のパケットの場合、送信元ポートと宛先ポートの両方が 0 です。
- **フォワーダ**：パケットをオーナーに転送するノード。フォワーダが接続のパケットを受信したときに、その接続のオーナーが自分ではない場合は、フォワーダはディレクタにオーナーを問い合わせしてから、そのオーナーへのフローを確立します。これは、この接続に関してフォワーダが受信するその他のパケット用です。ディレクタは、フォワーダにもなることができます。ディレクタローカリゼーションを有効にすると、フォワーダは常にローカルディレクタに問い合わせを行います。フォワーダがグローバルディレクタに問い合わせを行うのは、ローカルディレクタがオーナーを認識していない場合だけです。たとえば、別のサイトで所有されている接続のパケットをクラスタメンバーが受信する場合などです。フォワーダが SYN-ACK パケットを受信した場合、フォワーダはパケットの SYN キーからオーナーを直接取得できるので、ディレクタに問い合わせる必要がないことに注意してください。（TCP シーケンスのランダム化を無効にした場合は、SYN Cookie は使用されないで、ディレクタへの問い合わせが必要です）。存続期間が短いフロー（たとえば DNS や ICMP）の場合は、フォワーダは問い合わせの代わりにパケットを即座にディレクタに送信し、ディレクタがそのパケットをオーナーに送信します。1 つの接続に対して、複数のフォワーダが存在できます。最も効率的なスループットを実現できるのは、フォワーダが 1 つもなく、接続のすべてのパケットをオーナーが受信するという、優れたロードバランシング方法が使用されている場合です。



(注) クラスタリングを使用する場合は、TCP シーケンスのランダム化を無効にすることは推奨されません。SYN/ACK パケットがドロップされる可能性があるため、一部の TCP セッションが確立されない可能性があります。

- **フラグメントオーナー**：フラグメント化されたパケットの場合、フラグメントを受信するクラスタノードは、フラグメントの送信元と宛先の IP アドレス、およびパケット ID のハッシュを使用してフラグメントオーナーを特定します。その後、すべてのフラグメントがクラスタ制御リンクを介してフラグメント所有者に転送されます。スイッチのロードバ

ランスハッシュで使用される5タプルは、最初のフラグメントにのみ含まれているため、フラグメントが異なるクラスタノードにロードバランシングされる場合があります。他のフラグメントには、送信元ポートと宛先ポートは含まれず、他のクラスタノードにロードバランシングされる場合があります。フラグメント所有者は一時的にパケットを再アセンブルするため、送信元/宛先 IP アドレスとポートのハッシュに基づいてディレクタを決定できます。新しい接続の場合は、フラグメントの所有者が接続所有者として登録されます。これが既存の接続の場合、フラグメント所有者は、クラスタ制御リンクを介して、指定された接続所有者にすべてのフラグメントを転送します。その後、接続の所有者はすべてのフラグメントを再構築します。

接続でポートアドレス変換 (PAT) を使用すると、PAT のタイプ (per-session または multi-session) が、クラスタのどのメンバが新しい接続のオーナーになるかに影響します。

- per-session PAT : オーナーは、接続の最初のパケットを受信するノードです。
デフォルトでは、TCP および DNS UDP トラフィックは per-session PAT を使用します。
- multi-session PAT : オーナーは常に制御ノードです。multi-session PAT 接続がデータノードで最初に受信される場合、データノードがその接続を制御ノードに転送します。
デフォルトでは、UDP (DNS UDP を除く) および ICMP トラフィックは multi-session PAT を使用するため、それらの接続は常に制御ノードによって所有されています。

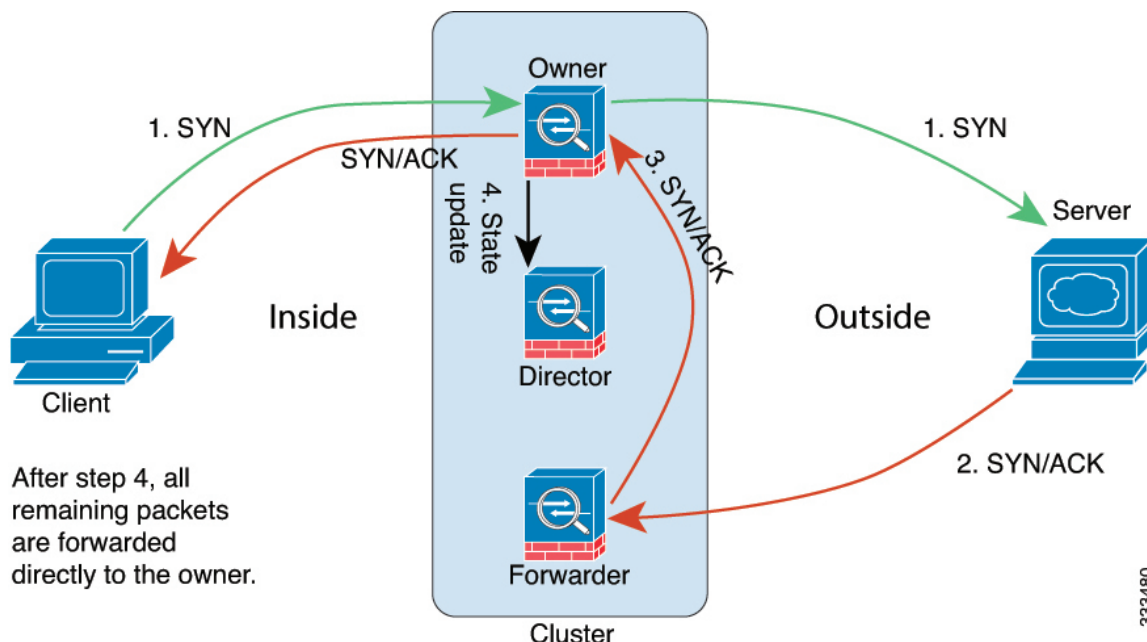
TCP および UDP の per-session PAT デフォルトを変更できるので、これらのプロトコルの接続は、その設定に応じて per-session または multi-session で処理されます。ICMP の場合は、デフォルトの multi-session PAT から変更することはできません。per-session PAT の詳細については、『ファイアウォールの構成ガイド』を参照してください。

新しい接続の所有権

新しい接続がロードバランシング経由でクラスタのノードに送信される場合は、そのノードがその接続の両方向のオーナーとなります。接続のパケットが別のノードに到着した場合は、そのパケットはクラスタ制御リンクを介してオーナーノードに転送されます。最適なパフォーマンスを得るには、適切な外部ロードバランシングが必要です。1つのフローの両方向が同じノードに到着するとともに、フローがノード間に均等に分散されるようにするためです。逆方向のフローが別のノードに到着した場合は、元のノードにリダイレクトされます。

TCP のサンプルデータフロー

次の例は、新しい接続の確立を示します。



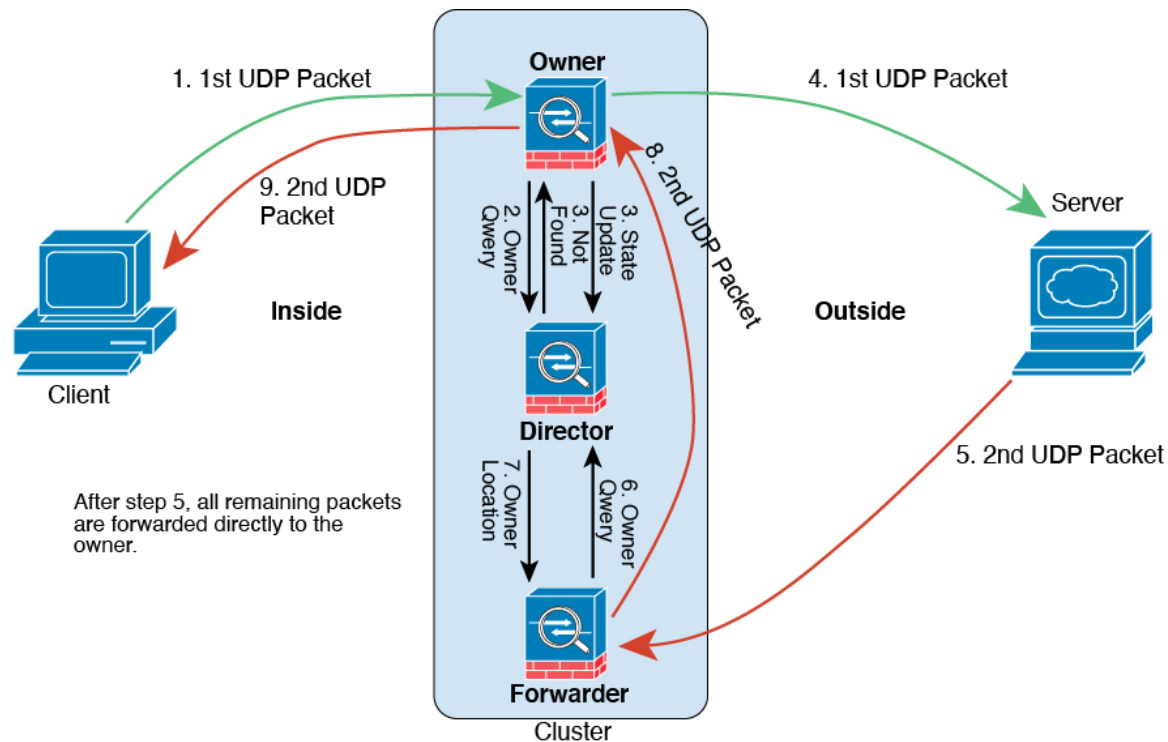
333480

1. SYN パケットがクライアントから発信され、ASA の1つ（ロードバランシング方法に基づく）に配信されます。これがオーナーとなります。オーナーはフローを作成し、オーナー情報をエンコードして SYN Cookie を生成し、パケットをサーバに転送します。
2. SYN-ACK パケットがサーバから発信され、別の ASA（ロードバランシング方法に基づく）に配信されます。この ASA はフォワーダです。
3. フォワーダはこの接続を所有してはいないので、オーナー情報を SYN Cookie からデコードし、オーナーへの転送フローを作成し、SYN-ACK をオーナーに転送します。
4. オーナーはディレクタに状態アップデートを送信し、SYN-ACK をクライアントに転送します。
5. ディレクタは状態アップデートをオーナーから受信し、オーナーへのフローを作成し、オーナーと同様に TCP 状態情報を記録します。ディレクタは、この接続のバックアップオーナーとしての役割を持ちます。
6. これ以降、フォワーダに配信されたパケットはすべて、オーナーに転送されます。
7. パケットがその他のノードに配信された場合、そのノードはディレクタに問い合わせ、オーナーを特定し、フローを確立します。
8. フローの状態が変化した場合、状態アップデートがオーナーからディレクタに送信されます。

ICMP および UDP のサンプルデータフロー

次の例は、新しい接続の確立を示します。

1. 図 2: ICMP および UDP データフロー



UDP パケットがクライアントから発信され、1つの ASA（ロードバランシング方法に基づく）に配信されます。

2. 最初の packets を受信したノードは、送信元/宛先 IP アドレスとポートのハッシュに基づいて選択されたディレクタノードをクエリします。
3. ディレクタは既存のフローを検出せず、ディレクタフローを作成して、以前のノードにパケットを転送します。つまり、ディレクタがこのフローのオーナーを選択したことになります。
4. オーナーはフローを作成し、ディレクタに状態アップデートを送信して、サーバーにパケットを転送します。
5. 2 番目の UDP パケットはサーバーから発信され、フォワーダに配信されます。
6. フォワーダはディレクタに対して所有権情報をクエリします。存続期間が短いフロー（DNS など）の場合、フォワーダはクエリする代わりにパケットを即座にディレクタに送信し、ディレクタがそのパケットをオーナーに送信します。
7. ディレクタは所有権情報をフォワーダに返信します。
8. フォワーダは転送フローを作成してオーナー情報を記録し、パケットをオーナーに転送します。
9. オーナーはパケットをクライアントに転送します。

新しいTCP 接続のクラスタ全体での再分散

アップストリームまたはダウンストリームルータによるロードバランシングの結果として、フロー分散に偏りが生じた場合は、新しいTCP フローを過負荷のノードから他のノードにリダイレクトするように設定できます。既存のフローは他のノードには移動されません。

ASA 仮想クラスタリングの履歴

機能名	バージョン	機能情報
バイアス言語の除去	9.19(1)	<p>「Master」と「Slave」という用語を含むコマンド、コマンド出力、syslog メッセージは、「Control」と「Control」に変更されました。</p> <p>新規/変更されたコマンド：cluster control-node、enable as-data-node、prompt、show cluster history、show cluster info</p>
VMware および KVM 用の ASA v30、ASA v50、および ASA v100 クラスタリング	9.17(1)	<p>ASA 仮想 クラスタリングを使用すると、最大 16 の ASA 仮想 を単一の論理デバイスとしてグループ化できます。クラスタは、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）を備える一方で、複数デバイスによって高いスループットおよび冗長性を達成します。ASA 仮想 クラスタリングは、ルーテッドファイアウォールモードで個別インターフェイスモードをサポートします。スパンド EtherChannels はサポートされていません。ASA 仮想 は、クラスタ制御リンクに VXLAN 仮想インターフェイス（VNI）を使用します。</p> <p>新規/変更されたコマンド：cluster-interface vni、nve-only cluster、peer-group、show cluster info、show cluster info instance-type、show nve 1</p>

【注意】シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2023 Cisco Systems, Inc. All rights reserved.

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。