



パブリッククラウドにおける **ASA Virtual** の **ASA クラスタ**

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。以下を使用して、パブリッククラウドに ASA 仮想 クラスタを展開できます。

- Amazon Web Services (AWS)
- Microsoft Azure

現在は、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（104 ページ）](#) を参照してください。

- [パブリッククラウドにおける ASA Virtual クラスタリングについて（2 ページ）](#)
- [ASA Virtual クラスタリングのライセンス（7 ページ）](#)
- [ASA Virtual クラスタリングの要件と前提条件（7 ページ）](#)
- [ASA Virtual クラスタリングのガイドライン（9 ページ）](#)
- [AWS でクラスタを展開する（10 ページ）](#)
- [Azure でクラスタを展開する（34 ページ）](#)
- [Azure での ASA 仮想 クラスタリング自動スケールソリューション（47 ページ）](#)
- [GCP の自動スケールソリューションを使用した Cisco ASA Virtual クラスタリング（70 ページ）](#)
- [クラスタリング動作のカスタマイズ（82 ページ）](#)
- [クラスタノードの管理（87 ページ）](#)
- [クラスタのモニタリング（92 ページ）](#)
- [クラスタリングの参考資料（104 ページ）](#)
- [パブリッククラウドにおける ASA Virtual クラスタリングの履歴（121 ページ）](#)

パブリッククラウドにおける ASA Virtual 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 仮想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。
- ロードバランサ：外部ロードバランシングには次のオプションがあります。
 - AWS Gateway Load Balancer

AWS ゲートウェイロードバランサは、透過的なネットワークゲートウェイと、トラフィックを分散し、仮想アプライアンスをオンデマンドで拡張するロードバランサを組み合わせます。ASA 仮想は、Geneve インターフェイスのシングルアームプロキシを使用して分散データプレーン（ゲートウェイ ロードバランサ エンドポイント）を備えたゲートウェイロードバランサ集中型コントロールプレーンをサポートします。

- シスコクラウドサービスルータなどの内部および外部ルータを使用した等コストマルチパスルーティング（ECMP）

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 仮想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 仮想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 仮想を設定する必要があります。



(注) レイヤ2スパンドEtherChannelsはロードバランシングで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。インターフェイス構成は、制御ノードでのみ設定する必要があり、各インターフェイスはDHCPを使用します。



(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 の詳細については、[VXLAN インターフェイス](#)を参照してください。

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 Cisco 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、タグging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、Cisco 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。



(注) to-the-box トラフィックをノードの管理 IP アドレスに転送する必要があります。to-the-box トラフィックは、クラスタ制御リンクを介して他のノードに転送されません。

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。

ASA Virtual 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 Virtual クラスタリングの要件と前提条件

モデルの要件

- ASAv30、ASAv50、ASAv100
- 以下のパブリッククラウドサービス：
 - Amazon Web Services (AWS)
 - Microsoft Azure
- 最大 16 ノード

『[ASA Virtual Getting Started Guide](#)』の ASA virtual の一般的な要件も参照してください。

ハードウェアおよびソフトウェアの要件

クラスタ内のすべてのノード：

- 同じパフォーマンス層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。ソフトウェアバージョンが一致しないとパフォーマンスが低下する可能性があるため、すべてのノードを同じメンテナンス期間でアップグレードするようにしてください。
- 単一の可用性ゾーンを展開がサポートされています。
- クラスタ制御リンクのインターフェイスは同じサブネット内に配置される必要があるため、クラスタは同じサブネットに展開する必要があります。

MTU

クラスタ制御リンクに接続されているポートに適切な MTU 値（高い値）が設定されていること。MTU の不一致がある場合、クラスタの形成に失敗します。クラスタに参加したノードは、クラスタ制御リンク MTU と一致するパケットサイズで制御ノードに ping を送信することで MTU の互換性をチェックします。最初の ping が失敗すると、ノードは、ping が成功するまで、より小さいパケットサイズ（MTU を 2 で、次に 4 で、8 で割った値）を使用して ping を試行します。通知が生成されるため、接続スイッチの MTU 不一致を修正して再試行することができます。

クラスタ制御リンクの MTU は、データインターフェイスよりも 154 バイト大きく設定されているはずです。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）と 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。

GWLB を使用する AWS の場合、データインターフェイスは Geneve カプセル化を使用します。この場合、イーサネットデータグラム全体がカプセル化されるため、新しいパケットのサイズが大きくなるため、より大きな MTU が必要になります。送信元インターフェイス MTU をネットワーク MTU + 306 バイトに設定する必要があります。したがって、標準の 1500 MTU ネットワークパスの場合、送信元インターフェイスの MTU は 1806 であり、クラスタ制御リンクの MTU は +154 の 1960 である必要があります。

GWLB を使用する Azure の場合、データインターフェイスは VXLAN カプセル化を使用します。この場合、イーサネットデータグラム全体がカプセル化されるため、新しいパケットのサイズが大きくなるため、より大きな MTU が必要になります。送信元インターフェイス MTU をネットワーク MTU + 54 バイトに設定する必要があります。

次の表は、推奨されるクラスタ制御リンク MTU とデータインターフェイス MTU を示しています。



(注) クラスタ制御リンクの MTU を 2561 ～ 8362 に設定することは推奨されません。ブロックプールの処理が原因で、この MTU サイズはシステム動作に最適ではありません。

表 1: 推奨 MTU

パブリック クラウド	クラスタ制御リンク MTU	データインターフェイス MTU
GWLB を使用した AWS	1960	1806
AWS	1654	1500
GWLB を使用した Azure	1554	1454
Azure	1554	1400

ASA Virtual クラスタリングのガイドライン

ハイ アベイラビリティ

クラスタリングでは、高可用性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（EtherChannel インターフェイスの追加または削除、ASA virtual 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、冗長スイッチシステム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や無効なインターフェイスに対するインターフェイスモニタリングを無効にする必要があります。トポロジの変更が完了して、コンフィギュレーション変更がすべてのユニットに同期されたら、インターフェイスのヘルス 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。
- 復号された TLS/SSL 接続の場合、復号状態は同期されず、接続オーナーに障害が発生すると、復号された接続がリセットされます。新規ノードへの接続を新たに確立する必要があります。復号されていない接続（復号しないルールに一致）は影響を受けず、正しく複製されます。
- ダイナミックスケールリングはサポートされていません。
- 各メンテナンスウィンドウの完了後にグローバル展開を実行します。
- 自動スケールグループ（AWS）やスケールセット（Azure）から一度に複数のデバイスを削除しないでください。また、自動スケールグループ（AWS）やスケールセット（Azure）

からデバイスを削除する前に、デバイスで **cluster disable** コマンドを実行することを推奨します。

- クラスタ内のデータノードと制御ノードを無効にする場合は、制御ノードを無効にする前にデータノードを無効にすることを推奨します。クラスタ内に他のデータノードがあるときに制御ノードが無効になっている場合は、いずれかのデータノードを制御ノードに昇格させる必要があります。ロールの変更はクラスタを妨害する可能性があることに注意してください。
- このガイドに記載されている Day 0 構成スクリプトでは、要件に応じて IP アドレスを変更し、カスタムインターフェイス名を指定して、CCL-Link インターフェイスのシーケンスを変更することができます。

クラスタリングのデフォルト

- cLACP システム ID は自動生成され、システムの優先順位はデフォルトでは 1 になっています。
- 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モニタリング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。
- 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。

AWS でクラスタを展開する

AWS にクラスタを展開する場合、手動で展開するか、スタックを展開する CloudFormation テンプレートを使用できます。AWS ゲートウェイロードバランサ、または Cisco Cloud Services Router などの非ネイティブのロードバランサでクラスタを使用できます。

AWS ゲートウェイロードバランサおよび Geneve シングルアームプロキシ

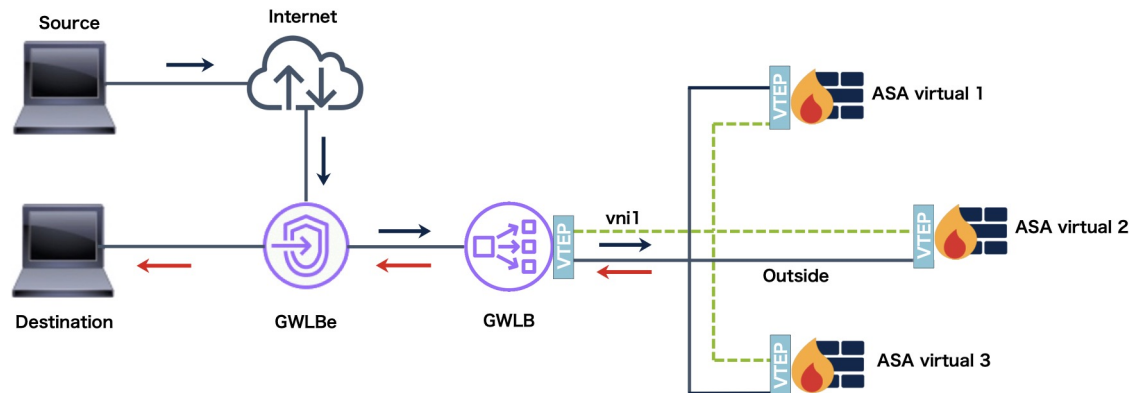


(注) この使用例は、現在サポートされている Geneve インターフェイスの唯一の使用例です。

AWS ゲートウェイロードバランサは、透過的なネットワークゲートウェイと、トラフィックを分散し、仮想アプライアンスをオンデマンドで拡張するロードバランサを組み合わせます。ASA Virtual は、分散データプレーン（ゲートウェイロードバランサエンドポイント）を備え

たゲートウェイロードバランサ集中型コントロールプレーンをサポートします。次の図は、ゲートウェイロードバランサのエンドポイントからゲートウェイロードバランサに転送されるトラフィックを示しています。ゲートウェイロードバランサは、複数の ASA Virtual の間でトラフィックのバランスをとり、トラフィックをドロップするか、ゲートウェイロードバランサに送り返す（Uターントラフィック）前に検査します。ゲートウェイロードバランサは、トラフィックをゲートウェイロードバランサのエンドポイントと宛先に送り返します。

図 1: Geneve シングルアームプロキシ



トポロジの例

AWS リージョン内の単一および複数の可用性ゾーンにおける自動スケーリングを使用した ASA 仮想 クラスターリング

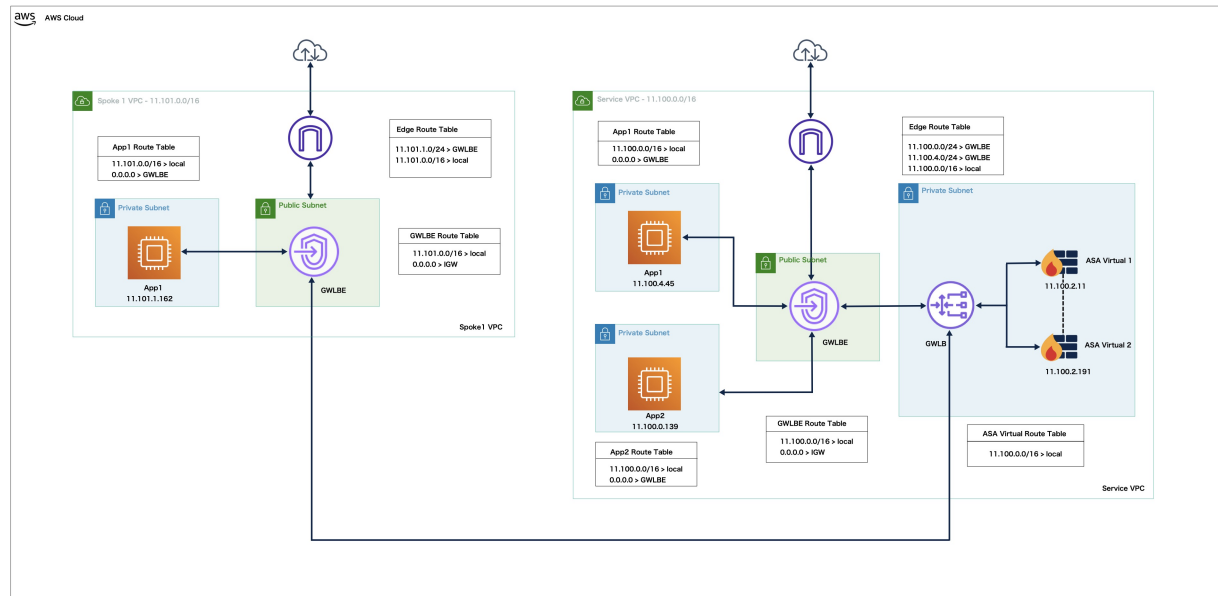
可用性ゾーンは、スタンドアロンデータセンター、または独立して動作する AWS リージョン内の一連の独立したデータセンターです。各ゾーンには独自のネットワークインフラストラクチャ、接続、および電源があり、1つのゾーンで障害が発生しても他のゾーンには影響しません。冗長性と信頼性を向上させるために、企業は、ディザスタリカバリ計画で複数の可用性ゾーンを使用しています。

複数の可用性ゾーンに ASA 仮想を展開し、動的拡張を使用してクラスターリングを設定すると、インフラストラクチャの可用性と拡張性が大幅に向上します。さらに、同じリージョン内で複数の可用性ゾーンを利用することにより、冗長性が向上し、障害発生時の高可用性が保証されます。

クラスタ制御リンク（CCL）の IP 割り当てメカニズムを変更して、AWS 上の ASA 仮想 クラスタについて、単一と複数の可用性ゾーンの展開を両方サポートできます。次に示すトポロジは、自動スケーリング機能を備えた単一および複数の可用性ゾーンにおける着信と発信の両方のトラフィックフローを示しています。

単一の可用性ゾーンにおける自動スケーリングを使用した ASA 仮想 クラスターリング

GWLB に接続されているクラスタには、2つの ASA 仮想 インスタンスがあります。

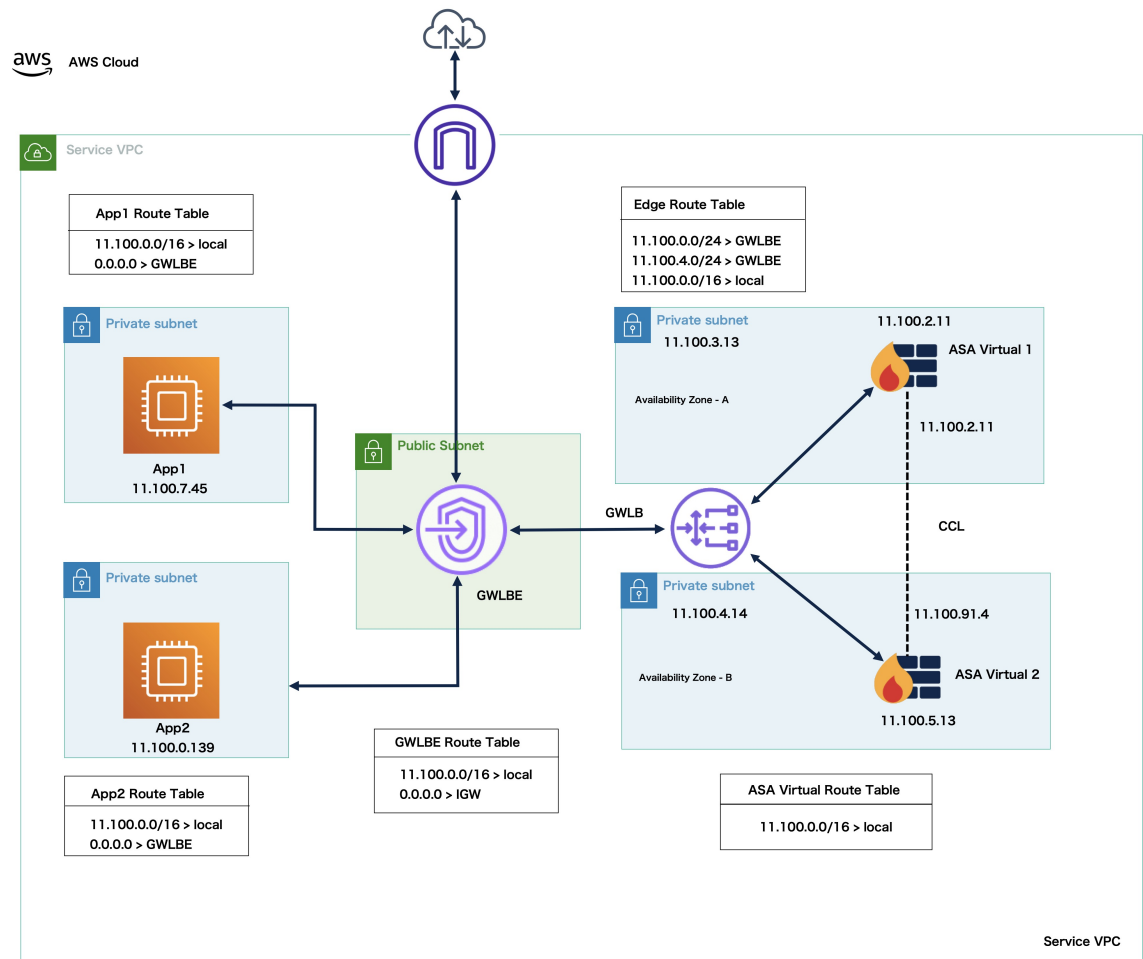


インターネットからの着信トラフィックは、GWLBエンドポイントに送られ、そこからGWLBにトラフィックが送信されます。その後、トラフィックはASA仮想クラスタに転送されます。トラフィックは、クラスタ内のASA仮想インスタンスによって検査された後、アプリケーション VM App1 に転送されます。

App1 からの発信トラフィックは、GWLB エンドポイント > GWLB > ASA > GWLB > GWLB エンドポイントに送信され、そこからインターネットに送信されます。

自動スケールソリューションを伴う複数の可用性ゾーンにおける自動スケールを使用したASA仮想 クラスタリング

GWLB に接続されている異なる可用性ゾーンのクラスタには、2 つの ASA 仮想 インスタンスがあります。



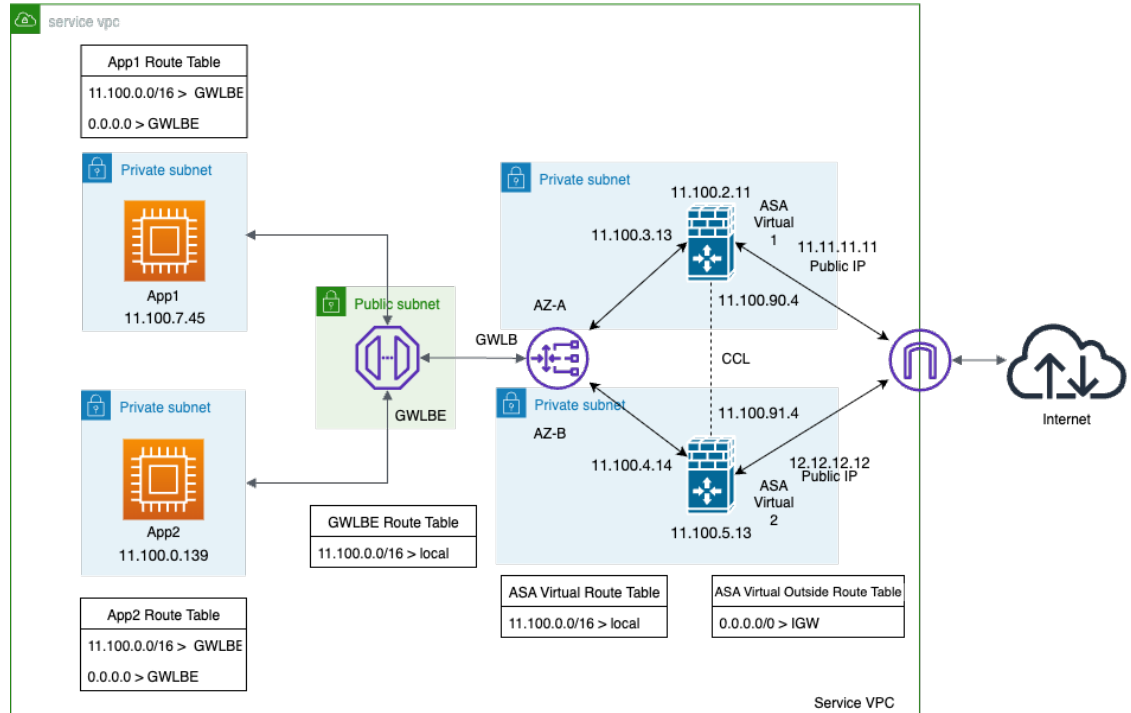
インターネットからの着信トラフィックは、GWLBEエンドポイントに送られ、そこからGWLBにトラフィックが送信されます。その後、可用性ゾーンに基づいて、トラフィックがASA 仮想クラスタにルーティングされます。トラフィックは、クラスタ内のASA 仮想インスタンスによって検査された後、アプリケーション VM App1 に転送されます。

AWS ゲートウェイロードバランサおよび Geneve デュアルアームプロキシ

デュアルアームプロキシは、ASA Virtual を有効にしてトラフィックを検査し、ネットワークアドレス変換 (NAT) を適用して、外部インターフェイスからインターネット ゲートウェイ経由でインターネットに直接送信するネットワーク展開モードです。この直接出力パスにより、GWLBEとその出力時のエンドポイントがバイパスされ、トラフィックフローが合理化されて効率が向上します。このアプローチは、複数の VPC からのアウトバウンドトラフィックが単一の共通出力ポイントを共有できるマルチ VPC 環境に特に効果的です。これにより、イン

フラストラクチャの要件が軽減され、ソリューションの費用対効果が高まります。さらに、トラフィックをより効率的に処理するためのクラスタリングもサポートしています。

トポロジの例

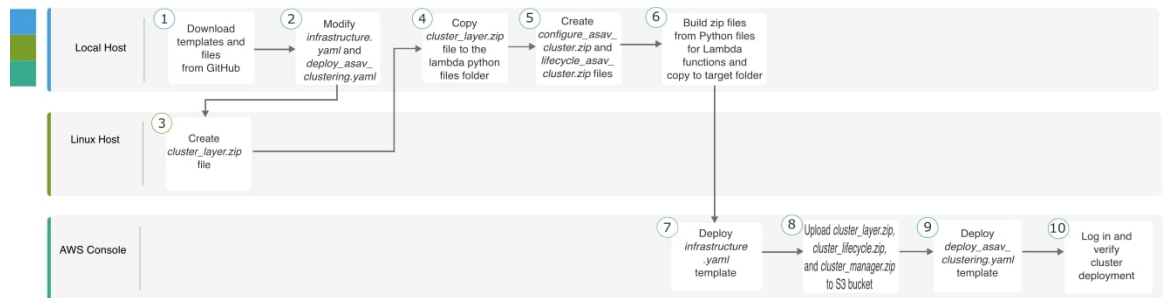


この展開の目的は、インターネットに送信される前にアプリケーショントラフィックを検査することです。これは、デュアルアームデザイン内の個別のアベイラビリティゾーンにわたって展開された、クラスタ化された Cisco ASA Virtual インスタンスによって行われます。内部インターフェイスが入力トラフィックを、外部インターフェイスが出力トラフィックを処理します。このフローでは、アプリケーションからのトラフィックは GWLB にルーティングされ、検査のために Cisco ASA Virtual の内部インターフェイスに転送されます。NAT の適用後、Cisco ASA Virtual は、外部インターフェイスを介してインターネットゲートウェイに直接トラフィックを送信します。

AWS で ASA Virtual クラスタを展開するためのエンドツーエンドのプロセス

テンプレートベースの展開

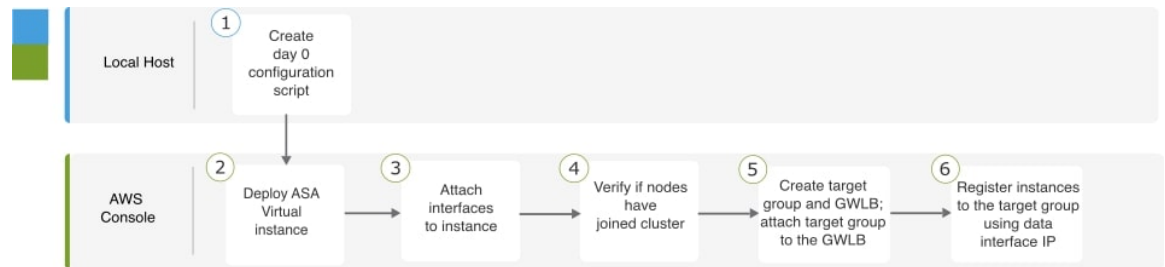
次のフローチャートは、AWS での ASA Virtual クラスタのテンプレートベース展開のワークフローを示しています。



	ワークスペース	手順
①	ローカルホスト	GitHub からテンプレートとファイルをダウンロードします。
②	ローカルホスト	infrastructure.yaml および deploying_asav_clustering.yaml テンプレートを変更します。
③	Linux ホスト	cluster_layer.zip ファイルを作成します。
④	ローカルホスト	cluster_layer.zip ファイルを Lambda Python ファイルフォルダにコピーします。
⑤	ローカルホスト	configure_asav_cluster.zip および lifecycle_asav_cluster.zip ファイルを作成します。
⑥	ローカルホスト	Lambda 関数の Python ファイルから zip ファイルを作成し、ターゲットフォルダにコピーします。
⑦	AWS コンソール	Infrastructure.yaml テンプレートを展開します。
⑧	AWS コンソール	cluster_layer.zip、cluster_lifecycle.zip、および cluster_manager.zip を S3 バケットにアップロードします。
⑨	AWS コンソール	deploy_asav_clustering.yaml テンプレートを展開します。
⑩	AWS コンソール	ログインして、クラスタの展開を確認します。

手動展開

次のフローチャートは、AWS での ASA Virtual クラスタの手動展開のワークフローを示しています。



	ワークスペース	手順
①	ローカルホスト	Day 0 構成スクリプトを作成します。
②	AWS コンソール	ASA virtual インスタンスを展開します。
③	AWS コンソール	インスタンスにインターフェイスを接続します。
④	AWS コンソール	ノードがクラスタに参加しているかどうかを確認します。
⑤	AWS コンソール	ターゲットグループと GWLB を作成します。ターゲットグループを GWLB に割り当てます。
⑥	AWS コンソール	データインターフェイス IP を使用してターゲットグループにインスタンスを登録します。

テンプレート

以下のテンプレートは [GitHub](#) で入手できます。パラメータ値は、テンプレートで指定されたパラメータ名、デフォルト値、使用可能な値、および説明により自明です。

- [infrastructure.yaml](#) : インフラストラクチャ展開用のテンプレート。
- [deploy_asav_clustering.yaml](#) : クラスタ展開用のテンプレート。



(注) クラスタノードを展開する前に、サポートされている AWS インスタンスタイプのリストを確認してください。このリストは、`deploy_asav_clustering.yaml` テンプレートのパラメータ `InstanceType` に使用可能な値の下にあります。

AWS における GWLB を使用した ASA 仮想 クラスタリングのターゲットフェールオーバーの設定

AWS の ASA virtual クラスタリングは、ゲートウェイロードバランサ (GWLB) を使用して、指定された ASA virtual ノードにインスペクション用のネットワークパケットをバランシング

および転送します。GWLB は、ターゲットノードのフェールオーバーまたは登録解除が発生した場合に、ターゲットノードにネットワークパケットを送信し続けるように設計されています。

AWS でターゲットフェールオーバー機能を使用すると、計画メンテナンス中またはターゲットノードの障害時にノードの登録が解除された場合に、GWLB がネットワークパケットを正常なターゲットノードにリダイレクトできるようになります。この機能ではクラスタのステータスフル フェールオーバーを利用しています。



- (注) GWLB が SSH、SCP、CURL、またはその他のプロトコルを使用してトラフィックをルーティングしている間にターゲットノードに障害が発生した場合、正常なターゲットへのトラフィックのリダイレクトに遅延が発生する可能性があります。この遅延は、トラフィックフローの再調整と再ルーティングが原因で発生します。

AWS では、AWS ELB API または AWS コンソールを介してターゲットフェールオーバーを設定できます。

- AWS API (AWS Elastic Load Balancing (ELB) API 内) の `modify-target-group-attributes` で、次の 2 つの新しいパラメータを変更することにより、フロー処理の動作を定義できます。
 - `target_failover.on_unhealthy` : ターゲットが正常でなくなった場合に GWLB がネットワークフローをどのように処理するかを定義します。
 - `target_failover.on_deregistration` : ターゲットが登録解除された場合に GWLB がネットワークフローをどのように処理するかを定義します。

次のコマンドは、これら 2 つのパラメータを定義するサンプルの API パラメータ設定を示しています。

```
aws elbv2 modify-target-group-attributes \  
--target-group-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:target-group/tg-1a2b3c4d5e6f7g8h9i0j \  
--attributes \  
Key=target_failover.on_unhealthy, Value=rebalance[no_rebalance] \  
Key=target_failover.on_deregistration, Value=rebalance[no_rebalance]
```

詳細については、AWS のマニュアルの「[TargetGroupAttribute](#)」を参照してください。

- AWS コンソール : EC2 コンソールでは、次のオプションを設定することにより、[ターゲットグループ (Target Group)] ページの [ターゲットフェールオーバー (Target Failover)] オプションを有効にすることができます。
 - ターゲットグループの属性を編集する
 - ターゲットフェールオーバーを有効にする
 - 再調整フローを確認する

ターゲットフェールオーバーを有効にする方法の詳細については、「[ターゲットフェールオーバー機能を有効にする](#)」を参照してください。

CloudFormation テンプレートを使用した AWS へのスタックの展開

CloudFormation テンプレートを使用して AWS にスタックを展開します。

始める前に

- Python 3 をインストールした Linux コンピューターが必要です。

手順

ステップ 1 テンプレートを準備します。

- GitHub リポジトリをローカルフォルダに複製します。 <https://github.com/CiscoDevNet/cisco-asav/tree/master/cluster/aws> を参照してください。
- 必要なパラメータを使用して、**infrastructure.yaml** および **deploy_asav_clustering.yaml** を変更します。
- cluster_layer.zip** という名前のファイルを作成して、重要な Python ライブラリを Lambda 関数に提供します。

cluster_layer.zip ファイルを作成するには、Python 3.9 がインストールされた Amazon Linux を使用することをお勧めします。

(注)

Amazon Linux 環境が必要な場合は、Amazon Linux 2023 AMI を使用して EC2 インスタンスを作成するか、Amazon Linux の最新バージョンを実行する AWS Cloudshell を使用できます。

cluster-layer.zip ファイルを作成するには、最初に Python ライブラリパッケージの詳細で構成される **requirements.txt** ファイルを作成してから、シェルスクリプトを実行する必要があります。

1. Python パッケージの詳細を指定して、**requirements.txt** ファイルを作成します。

以下は、**requirements.txt** ファイルで指定するサンプルパッケージの詳細です。

```
$ cat requirements.txt
pycryptodome
paramiko
requests
scp
jsonschema
cffi
zip
importlib-metadata
```

2. 次のシェルスクリプトを実行して、**cluster_layer.zip** ファイルを作成します。

```
$ pip3 install --platform manylinux2014_x86_64
--target=./python/lib/python3.9/site-packages
--implementation cp --python-version 3.9 --only-binary=:all:
--upgrade -r requirements.txt
$ zip -r cluster_layer.zip ./python
```

(注)

インストール中に urllib3 や暗号化などの依存関係の競合エラーが発生した場合は、競合するパッケージを推奨バージョンと一緒に **requirements.txt** ファイルに含めることをお勧めします。その後、インストールを再度実行して競合を解決できます。

- d) 結果の **cluster_layer.zip** ファイルを Lambda Python ファイルフォルダにコピーします。
- e) **configure_asav_cluster.zip** および **lifecycle_asav_cluster.zip** ファイルを作成します。

make.py ファイルは、複製されたリポジトリの最上位ディレクトリにあります。これにより、python ファイルが Zip ファイルに圧縮され、ターゲットフォルダにコピーされます。

python3 make.py build

(注)

クラスタを展開する前に、ASAv クラスタリング自動スケール展開に必要な VPC、サブネット、S3 バケットなどのインフラストラクチャ要件に合わせて、AWS CloudFormation テンプレートファイル (**infrastructure.yaml** および **deploy_asav_clustering.yaml**) をカスタマイズしていることを確認します。

すべてのクラスタ インフラストラクチャ サブネットは /27 CIDR を使用する必要があります

ステップ 2 Infrastructure.yaml を展開し、クラスタ展開の出力値をメモします。インフラストラクチャスタックを展開する前に、使用する AWS リージョンと可用性ゾーンを特定することが重要です。各 AWS リージョンには異なる可用性ゾーンと VPC インフラストラクチャのセットがあるため、展開に適したリージョンと可用性ゾーンを選択することが不可欠です。

- a) AWS コンソールで、[CloudFormation] に移動し、[新しいリソース (標準)] を使用 (With new resources(standard)) を選択して、[スタックの作成 (Create stack)] をクリックします。
- b) [テンプレートファイルのアップロード (Upload a template file)] を選択し、[ファイルの選択 (Choose file)] をクリックして、ターゲットフォルダから **infrastructure.yaml** を選択します。
- c) [次へ (Next)] をクリックして、必要な情報を入力します。

パラメータ	使用できる値/タイプ	説明
ClusterName	文字列	一意のクラスタ名を入力します。
ClusterNumber	番号	一意のクラスタ番号を入力します。
VpcCidr	文字列	新しい VPC の CIDR ブロックを入力します。
NoOfAZs	番号	サブネットが作成される可用性ゾーンの合計数 (最小は 1、最大は 3 でリージョンによって異なります)。 リリース 9.22 以降の場合にのみ、NoOfAZs = 2 または 3 を選択してください。それ以前のリリースでは、1 を選択します。

パラメータ	使用できる値/タイプ	説明
		管理サブネット、内部サブネット、および CCL サブネットは、以下のパラメータに基づいて AZ にまたがります。
ListOfAZs	リスト	可用性ゾーンを選択します（数は可用性ゾーンの数と一致する必要があります）
MgmtSubnetNames	カンマ区切りリスト	管理サブネット名を入力します（ルートとしてインターネット GW を使用）
MgmtSubnetCidrs	カンマ区切りリスト	管理サブネット CIDR を入力します
InsideSubnetNames	カンマ区切りリスト	内部サブネット名を入力します（プライベートルート付き）
InsideSubnetCidrs	カンマ区切りリスト	内部サブネット CIDR リストを入力します
CCLSubnetNames	カンマ区切りリスト	CCL サブネット名を入力します
CCLSubnetCidrs	カンマ区切りリスト	CCL サブネット CIDR を入力します
LambdaAZs	リスト	Lambda の 2 つの可用性ゾーンを選択します
LambdaSubnetNames	カンマ区切りリスト	Lambda 関数の Lambda サブネット名（ルートとして NAT GW を使用）を入力します
LambdaSubnetCidrs	カンマ区切りリスト	Lambda サブネット CIDR を入力します

- d) クラスタの一意の**クラスタ名**と**クラスタ番号**を入力します。
- e) [可用性ゾーン (Availability Zone)] リストから可用性ゾーンを選択します。このフィールドには、ClusterFormation テンプレートを使用してインフラストラクチャ スタックを展開するために選択した AWS リージョンに基づく可用性ゾーンのみが表示されます。
- f) [Inside subnet CIDR] にカンマで区切られた事前定義値のサブネット CIDR を入力します。
- g) Lambda が AZ 間で通信するために必要な可用性ゾーンを [List of Lambda AZs] から選択します。
- h) [Next]、[Create stack] の順にクリックします。
インフラストラクチャ スタックの展開ステータスを示すメッセージ「CREATE_COMPLETE」が表示されます。
- i) 展開が完了したら、[出力 (Outputs)] に移動し、S3 の **BucketName** を書き留めます。

図 2: Infrastructure.yaml の出力

Outputs (13)				
Search outputs				
Key	Value	Description	Export name	
AZ	sa-east-1a	Availability zone	-	
BucketName	ran-cl5-infra-s3bucketcluster-kckr7518u00l	Name of the Amazon S3 bucket	-	
BucketUrl	http://ran-cl5-infra-s3bucketcluster-kckr7518u00l.s3-website-sa-east-1.amazonaws.com	URL of S3 Bucket Static Website	-	
CCLSubnetId	subnet-050feb347e57eba99	CCL subnet ID	-	
EIPforNATgw	52.67.246.95	EIP reserved for NAT GW	-	
InInterfaceSGId	sg-0333e92f36b2aa0bf	Security Group ID for Instances Inside Interface	-	
InsideSubnetIds	subnet-047c0a2beffb5a70f	Inside subnet ID	-	
InstanceSGId	sg-0c0c6bfb5ba5f1c10	Security Group ID for Instances Management Interface	-	
LambdaSecurityGroupId	sg-01771b0d3012a40c5	Security Group ID for Lambda Functions	-	
LambdaSubnetIds	subnet-0fb24785c687d50e4,subnet-0f1996a02ffaa2e62	List of lambda subnet IDs (comma seperated)	-	
MgmtSubnetIds	subnet-02d4a757b95a9a5b9	Mangement subnet ID	-	
UseGWLB	Yes	Use Gateway Load Balancer	-	
VpcName	vpc-003b592ad2518d03d	Name of the VPC created	-	

ステップ 3 cluster_layer.zip、configure_asav_cluster.zip、および lifecycle_asav_cluster.zip を infrastructure.yaml で作成した S3 バケットにアップロードします。

ステップ 4 deploy_asav_clustering.yaml を展開します。

- [CloudFormation]に移動し、[新しいリソース（標準）を使用（With new resources(standard)）]を選択して、[スタックの作成（Create stack）]をクリックします。
- [テンプレートファイルのアップロード（Upload a template file）]を選択し、[ファイルの選択（Choose file）]をクリックして、ターゲットフォルダから **deploy_asav_clustering.yaml** を選択します。
- [次へ（Next）]をクリックして、必要な情報を入力します。
- 次のクラスタとインフラストラクチャの設定情報を入力します。

パラメータ	使用できる値/タイプ	説明
クラスタの設定		
ClusterGrpNamePrefix	文字列	これはクラスタ名のプレフィックスです。クラスタ番号がサフィックスとして追加されます。

パラメータ	使用できる値/タイプ	説明
ClusterNumber	文字列	これは、クラスタグループ名 (ASAv-Cluster) のサフィックスとして付加されます。この値が1の場合、グループ名は ASAv-Cluster-1 になります。1桁以上3桁以下の数字である必要があります。
ClusterSize	番号	クラスタ内の Cisco ASA Virtual ノードの総数。 最小値 : 1 最大値 : 16
インフラストラクチャの詳細		
NoOfAZs	文字列	これは、Cisco ASA Virtual が展開される可用性ゾーンの合計数です。(可用性ゾーンの数は1〜3でリージョンによって異なります)。 サブネットは、これらの可用性ゾーンに作成されます。 このリストで使用可能な可用性ゾーンは、クラスタの展開用に選択されたリージョンに基づいています。 (注) 管理、内部、およびクラスタ制御リンク (CCL) のサブネットは、このパラメータに基づいて3つの可用性ゾーンにわたって作成されます。
AZ	文字列	可用性ゾーンリストは、展開するリージョンに基づきます。 [可用性ゾーン (Availability Zone)] リストで、有効な可用性ゾーン (1つの可用性ゾーン、2つの可用性ゾーン、または3つの可用性ゾーン) を選択します。 カウントは、可用性ゾーン数のパラメータの値と一致する必要があります。
NotifyEmailID	文字列	クラスタイベントの電子メールの送信先となる電子メールアドレス。この電子メール通告を受信するには、サブスクリプション電子メール要求を承認する必要があります。 例 : admin@company.com
VpcId	文字列	クラスタグループの VPC ID。 タイプ : AWS::EC2::VPC::Id

パラメータ	使用できる値/タイプ	説明
S3BktName	文字列	アップロードされた Lambda zip ファイルを含む S3 バケツ。正しいバケツ名を指定する必要があります。
MgmtSubnetIds	リスト	可用性ゾーンごとに「1つ」のサブネットのみを入力します。 同じ可用性ゾーンから複数のサブネットを選択する場合、間違ったサブネットを選択すると、Cisco ASA Virtual インスタンスの展開中に問題が発生する可能性があります。 タイプ : List<AWS::EC2::Subnet::Id>
InsideSubnetIds	リスト	可用性ゾーンごとに少なくとも「1つ」のサブネットを入力します。 同じ可用性ゾーンから複数のサブネットを選択する場合、間違ったサブネットを選択すると、Cisco ASA Virtual インスタンスの展開中に問題が発生する可能性があります。 タイプ : List<AWS::EC2::Subnet::Id>
LambdaSubnets	リスト	Lambda 関数用に少なくとも「2つ」のサブネットを入力します。Lambda 関数が、パブリック DNS である AWS サービスと通信できるようにするには、入力する「2つ」のサブネットに NAT ゲートウェイが必要です。 タイプ : List<AWS::EC2::Subnet::Id>
CCLSubnetIds	文字列	必要に応じて、複数の AZ からサブネットを選択します。
CCLSubnetRanges	文字列	さまざまな可用性ゾーンの CCL サブネットの IP アドレス範囲を入力します。 最初の 4 つの予約済み IP アドレスを除外します。クラスタ制御リンク (CCL) の IP アドレスプール。 IP アドレスは、CCL IP プールから Cisco ASA Virtual の CCL インターフェイスに割り当てられます。
MgmtInterfaceSG	リスト	Cisco ASA Virtual インスタンスのセキュリティグループ ID を選択します。 タイプ : List<AWS::EC2::SecurityGroup::Id>

パラメータ	使用できる値/タイプ	説明
InsideInterfaceSG	リスト	NGFWv インスタンスの内部インターフェイスのセキュリティグループ ID を選択します。 指定されているすべての AZ のセキュリティグループを追加してください。 タイプ : List<AWS::EC2::SecurityGroup::Id>
LambdaSG	リスト	Lambda 関数のセキュリティグループを選択します。 発信接続が [ANYWHERE] に設定されていることを確認します。 タイプ : List<AWS::EC2::SecurityGroup::Id>
CCLInterfaceSG	リスト	Cisco ASA Virtual インスタンスの CCL インターフェイスのセキュリティグループ ID を選択します。
GWLB の設定		
DeployGWLBE	文字列	[はい (Yes)] をクリックして GWLB エンドポイントを展開します。 デフォルトでは、この値は [いいえ (No)] に設定されています。
VpcIdLBE	文字列	ゲートウェイロードバランサのエンドポイントを展開する VPC を入力します。 (注) GWLB エンドポイントを展開しない場合は、このフィールドに値を入力しないでください。
GWLBSubnetId	文字列	サブネット ID を 1 つだけ入力します。 (注) GWLB エンドポイントを展開しない場合は、このフィールドに値を入力しないでください。 サブネットが正しい VPC および指定した可用性ゾーンに属していることを確認します。
TargetFailover	文字列	ターゲットに障害が発生した場合または登録解除された場合のターゲット フェールオーバー サポートを有効にします (このパラメータの値はデフォルトで [再調整 (rebalance)] に設定されています)。

パラメータ	使用できる値/タイプ	説明
		• [再調整なし (no_rebalance)] : 既存のフローを障害が発生したターゲットに送信し、新しいフローを正常なターゲットに送信し、後方互換性を確保します。 • [再調整 (rebalance)] : 新しいフローが正常なターゲットに送られるようにしながら、既存のフローを再配布します。
TgHealthPort	文字列	GWLB の正常性チェックポートを入力します。 (注) デフォルトでは、このポートはトラフィックに使用されません。 指定した値が有効な TCP ポートであることを確認します。デフォルト : 8080
Cisco NGFWv インスタンスの設定		
InstanceType	文字列	Cisco ASA Virtual EC2 インスタンスタイプ。 選択したインスタンスタイプが AWS リージョンでサポートされていることを確認します。 デフォルトでは、c5.xlarge が選択されています。
LicenseType	文字列	Cisco ASA Virtual EC2 インスタンス ライセンス タイプを選択します。AMI-ID パラメータに入力する AMI ID が同じライセンスタイプであることを確認します。 デフォルトでは、[BYOL] が選択されています。
AssignPublicIP	文字列	AWS IP アドレスプールから Cisco ASA Virtual のパブリック IP アドレスを割り当てるには、値を [true] に設定します。
AmiID	文字列	リージョン、バージョン、およびライセンスタイプ (BYOL または PAYG) に従って正しい AMI ID を選択します。 タイプ : AWS::EC2::Image::Id
KmsArn	文字列	既存の KMS の ARN (保存時に暗号化するための AWS KMS キー) を入力します。

パラメータ	使用できる値/タイプ	説明
		このフィールドに値を指定する場合、Cisco ASA Virtual インスタンスの管理者パスワードは暗号化されたパスワードである必要があります。 暗号化パスワードの生成例 : <code>"aws kms encrypt --key-id <KMS ARN> --plaintext <password> "</code> パスワードの暗号化は、指定された ARN のみを使用して実行する必要があります。
スケーリングの設定		
CpuThresholds	カンマ区切りリスト	(任意) ゼロ以外の下限しきい値と上限しきい値を指定すると、スケールポリシーが作成されます。(0,0)を選択すると、CPU スケーリングアラームまたはポリシーは作成されません。評価ポイントとデータポイントは、デフォルト値または推奨値にします。 デフォルトでは、このテンプレートでは自動スケールが有効になっています。自動スケールは展開後に無効にできます。

- e) [次へ (Next)] をクリックします。
- f) クリックしてすべての AWS CloudFormation オプションを確認します。
- g) [送信 (Submit)] をクリックしてクラスタを展開します。
- h) [Next]、[Create Stack] の順にクリックします。

図 3: 展開されたリソース

Resources (21)					
Search resources					
Logical ID	Physical ID	Type	Status	Status reason	
ASAvGroup	ran-cl5-1	AWS::AutoScaling::AutoScalingGroup	CREATE_COMPLETE		
ASAvLaunchTemplate	lt-056fd20764270c893	AWS::EC2::LaunchTemplate	CREATE_COMPLETE		
CLSmanagerTopic	arn:aws:sns:sa-east-1:797661843114:ran-cl5-1-cluster-manager-topic	AWS::SNS::Topic	CREATE_COMPLETE		
ClusterManager	ran-cl5-1-manager-lambda	AWS::Lambda::Function	CREATE_COMPLETE		
ClusterManagerLogGrp	/aws/lambda/ran-cl5-1-manager-lambda	AWS::Logs::LogGroup	CREATE_COMPLETE		
ClusterManagerSNS1	arn:aws:sns:sa-east-1:797661843114:ran-cl5-1-cluster-manager-topic:e13bfc0-d698-4215-88a5-278474e22c32	AWS::SNS::Subscription	CREATE_COMPLETE		
ClusterManagerSNS1Permission	ran-cl5-ClusterManagerSNS1Permission-S6BQAE05OG6U	AWS::Lambda::Permission	CREATE_COMPLETE		
InstanceEvent	ran-cl5-1-notify-instance-event	AWS::Events::Rule	CREATE_COMPLETE		
InstanceEventInvokeLambdaPermission	ran-cl5-InstanceEventInvokeLambdaPermission-1XPS21Q4G2DY6	AWS::Lambda::Permission	CREATE_COMPLETE		

ステータスが CREATE_IN_PROGRESS から CREATE COMPLETE に変わり、展開が成功したことが示されます。

ステップ 5 いずれかのノードにログインし、**show cluster info** コマンドを入力して、クラスタの展開を確認します。

show cluster info

```
Cluster oneclicktest-cluster: On
Interface mode: individual
Cluster Member Limit : 16
This is "200" in state CONTROL_NODE
ID : 0
Version : 9.19.1
Serial No.: 9AU42EN5D1E
CCL IP : 169.254.200.200
CCL MAC : 4201.0a0a.0fc7
Module : ASAv
Resource : 4 cores / 8192 MB RAM
Last join : 15:26:22 UTC Jul 17 2022
Last leave: N/A
Other members in the cluster:
Unit "204" in state DATA_NODE
ID : 1
Version : 9.19.1
Serial No.: 9AJ9N46947R
CCL IP : 169.254.200.204
CCL MAC : 4201.0a0a.0fcb
Module : ASAv
```

```
Resource : 4 cores / 8192 MB RAM
Last join : 16:57:42 UTC Jul 17 2022
Last leave: 16:03:25 UTC Jul 17 2022
```

自動スケーリングのパラメータ設定

展開が完了したら、ASAv 自動スケーリンググループの [最小容量 (Minimum capacity)]、[最大容量 (Maximum capacity)]、および [必要な容量 (Desired capacity)] を指定する必要があります。自動スケーリング機能を検証する必要があります。

手順

ステップ 1 AWS コンソールから、[サービス (Services)] > [EC2] > [自動スケーリンググループ (Auto Scaling groups)] > 作成された **ClusterAutoscale** グループの順に選択します。

[EC2](#) > [Auto Scaling groups](#) > [ASAvMZ-cluster-1](#)

Edit ASAvMZ-cluster-1 [Info](#)

Group size [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum scaling limits.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances) ▼

Desired capacity

Specify your group size.

7

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity

1

Equal or less than
desired capacity

Max desired capacity

16

Equal or greater than
desired capacity

ステップ 2 [必要な容量 (Desired capacity)] を設定してから、[スケーリング制限 (Scaling limits)] 容量を設定します。

ステップ 3 CPU メトリックデータが使用可能かどうかと、AWS Cloudwatch アラームでスケーリングが予期したとおりに発生しているかどうかを確認します。

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
ASAvMZ-cluster-1-i-00cd9230817c747a4	i-00cd9230817c747a4	Running	c5.xlarge	2/2 checks passed	No alarms	ap-south-2b	ec2-18-6
ASAvMZ-cluster-1-i-03606293835beaea1	i-03606293835beaea1	Terminated	c5.xlarge	-	No alarms	ap-south-2b	-
ASAvMZ-cluster-1-i-038692319c2fe0ead	i-038692319c2fe0ead	Running	c5.xlarge	2/2 checks passed	No alarms	ap-south-2c	ec2-18-6
ASAvMZ-cluster-1-i-042fb42a59c6cf506	i-042fb42a59c6cf506	Running	c5.xlarge	2/2 checks passed	No alarms	ap-south-2a	ec2-18-6
ASAvMZ-cluster-1-i-055e4def0c4b46267	i-055e4def0c4b46267	Running	c5.xlarge	2/2 checks passed	No alarms	ap-south-2a	ec2-18-6
ASAvMZ-cluster-1-i-06dc45ed055bcc91	i-06dc45ed055bcc91	Running	c5.xlarge	2/2 checks passed	No alarms	ap-south-2c	ec2-18-6
ASAvMZ-cluster-1-i-091fa0fe9cfc4d14	i-091fa0fe9cfc4d14	Terminated	c5.xlarge	-	No alarms	ap-south-2c	-
ASAvMZ-cluster-1-i-0bc61188bcae0b9db	i-0bc61188bcae0b9db	Running	c5.xlarge	2/2 checks passed	No alarms	ap-south-2c	ec2-18-6

Instance: i-038692319c2fe0ead (ASAvMZ-cluster-1-i-038692319c2fe0ead)	
Instance summary Instance ID i-038692319c2fe0ead (ASAvMZ-cluster-1-i-038692319c2fe0ead)	Public IPv4 address 18.61.28.175 [open address] Instance state Running
IPv6 address -	Private IPv4 addresses 10.3.92.13 10.3.102.7 10.3.252.24 Public IPv4 DNS ec2-18-61-28-175.ap-south-2.compute.amazonaws.com [open address]

スタックの更新による Cisco ASA Virtual クラスタリングでの IMDSv2 必須モードの設定

AWS にすでに展開されている Cisco ASA Virtual の Auto Scale グループインスタンスの IMDSv2 必須モードを設定できます。

始める前に

IMDSv2 必須モードは、Cisco ASA Virtual バージョン 9.20.3 以降でのみサポートされています。展開に IMDSv2 必須モードを設定する前に、既存のインスタンスのバージョンが IMDSv2 API と互換性がある（バージョン 9.20.3 以降にアップグレードされている）ことを確認してください。

手順

- ステップ 1 AWS コンソールで、[CloudFormation] に移動し、[Stacks] をクリックします。
- ステップ 2 最初に展開されたクラスタリングインスタンスのスタックを選択します。
- ステップ 3 [更新 (Update)] をクリックします。
- ステップ 4 [Update stack] ページで、[Replace existing template] をクリックします。
- ステップ 5 [テンプレートの指定 (Specify template)] セクションで、[テンプレートファイルのアップロード (Upload a template file)] をクリックします。
- ステップ 6 IMDSv2 をサポートするテンプレートを選択してアップロードします。
- ステップ 7 テンプレートの入力パラメータの値を指定します。
- ステップ 8 スタックを更新します。

AWS でのクラスタの手動展開

クラスタを手動で展開するには、Day-0 構成を準備し、各ノードを展開します。

AWS 向け Day 0 構成の作成

次のコマンドを使用して各クラスタノードのブートストラップ設定を指定します。

ゲートウェイロードバランサの例

次の実行コンフィギュレーションの例では、U ターントラフィック用の 1 つの Geneve インターフェイスと、クラスタ制御リンク用の 1 つの VXLAN インターフェイスを備えたゲートウェイロードバランサの構成を作成します。

```
cluster interface-mode individual force
policy-map global_policy
class inspection_default
no inspect h323 h225
no inspect h323 ras
no inspect rtsp
no inspect skinny

int m0/0
management-only
nameif management
security-level 100
ip address dhcp setroute
no shut

interface TenGigabitEthernet0/0
nameif geneve-vtep-ifc
security-level 0
ip address dhcp
no shutdown

interface TenGigabitEthernet0/1
nve-only cluster
nameif ccl_link
security-level 0
ip address dhcp
no shutdown

interface vni1
description Clustering Interface
segment-id 1
vtep-nve 1

interface vni2
proxy single-arm
nameif ge
security-level 0
vtep-nve 2

object network ccl_link
range 10.1.90.4 10.1.90.254 //Mandatory user input, use same range on all nodes
object-group network cluster_group
network-object object ccl_link
nve 2
encapsulation geneve
```

```

source-interface geneve-vtep-ifc
nve 1
encapsulation vxlan
source-interface ccl_link
peer-group cluster_group

cluster group asav-cluster // Mandatory user input, use same cluster name on all nodes
local-unit 1 //Value in bold here must be unique to each node
cluster-interface vn1l ip 169.254.200.193 255.255.255.224 //Value in bold here must be
unique to each node
priority 1
enable noconfirm

mtu geneve-vtep-ifc 1806
mtu ccl_link 1960
aaa authentication listener https geneve-vtep-ifc port 7575 //Use same port number on
all nodes
jumbo-frame reservation
wr mem

```



(注) AWS ヘルスチェックの設定では、ここで設定した **aaa authentication listener https** ポートを必ず指定してください。

非ネイティブロードバランサの例

次の例では、管理、内部、および外部インターフェイスと、クラスタ制御リンク用の VXLAN インターフェイスを使用して、非ネイティブロードバランサ用の構成を作成します。

```

cluster interface-mode individual force
interface Management0/0
management-only
nameif management
ip address dhcp

interface GigabitEthernet0/0
no shutdown
nameif outside
ip address dhcp

interface GigabitEthernet0/1
no shutdown
nameif inside
ip address dhcp

interface GigabitEthernet0/2
nve-only cluster
nameif ccl_link
ip address dhcp
no shutdown

interface vn1l
description Clustering Interface
segment-id 1
vtep-nve 1

jumbo-frame reservation
mtu ccl_link 1654
object network ccl_link

```

```

range 10.1.90.4 10.1.90.254          //mandatory user input
object-group network cluster_group
network-object object ccl_link

nve 1
encapsulation vxlan
source-interface ccl_link
peer-group cluster_group

cluster group asav-cluster          //mandatory user input
local-unit 1                        //mandatory user input
cluster-interface vni1 ip 169.254.200.193 255.255.255.224 //mandatory user input
priority 1
enable

```



(注) 上記の設定をコピーして貼り付ける場合は、設定から **//mandatory user input** を必ず削除してください。

クラスタノードの展開

クラスタを形成するためにクラスタノードを展開します。

手順

ステップ 1 必要な数のインターフェイス（ゲートウェイロードバランサ（GWLB）を使用している場合は 3 つのインターフェイス、非ネイティブロードバランサを使用している場合は 4 つのインターフェイス）でクラスタの Day 0 構成を使用することにより、ASA Virtual インスタンスを展開します。これを行うには、[インスタンスの詳細設定（Configure Instance Details）] > [高度な詳細（Advanced Details）] セクションで、クラスタの Day 0 構成に貼り付けます。

(注)

次の順序でインスタンスにインターフェイスを接続していることを確認します。

- AWS ゲートウェイロードバランサの 3 つのインターフェイス：管理、外部、クラスタ制御リンク。
- 非ネイティブロードバランサの 4 つのインターフェイス：管理、内部、外部、クラスタ制御リンク。

AWS での ASA Virtual の展開の詳細については、「[Deploy the ASA Virtual on AWS](#)」を参照してください。

ステップ 2 ステップ 1 を繰り返して、必要な数の追加ノードを展開します。

ステップ 3 ASA Virtual コンソールで **show cluster info** コマンドを使用して、すべてのノードがクラスタに正常に参加したかどうかを確認します。

ステップ 4 AWS ゲートウェイロードバランサを設定します。

- a) ターゲットグループと GWLB を作成します。

- b) ターゲットグループを GWLB に割り当てます。

(注)

正しいセキュリティグループ、リスナー設定、およびヘルスチェック設定を使用するように GWLB を設定していることを確認します。

- c) IP アドレスを使用して、データインターフェイス（内部インターフェイス）をターゲットグループに登録します。詳細については、「[Create a Gateway Load Balancer](#)」を参照してください。

AWS での ASA virtual のターゲットフェールオーバーの有効化

ASA virtual のデータインターフェイスは、AWS の GWLB のターゲットグループに登録されます。ASA virtual クラスタリングでは、各インスタンスはターゲットグループに関連付けられています。GWLB は、ターゲットグループのターゲットノードとして識別または登録されているこの正常なインスタンスにトラフィックを負荷分散して送信します。

始める前に

手動で、または CloudFormation テンプレートを使用して、AWS で ASA virtual スタックを展開している必要があります。

CloudFormation テンプレートを使用してクラスタを展開する場合、クラスタ展開ファイル `deploy_asav_clustering.yaml` の [GWLB の設定 (GWLB Configuration)] セクションで使用可能な [再調整 (rebalance)] 属性を割り当てることにより、[ターゲットフェールオーバー (Target Failover)] のパラメータを有効にすることもできます。テンプレートでは、このパラメータの値はデフォルトで [再調整 (rebalance)] に設定されています。ただし、AWS コンソールでは、このパラメータのデフォルト値は [再調整なし (no_rebalance)] に設定されています。

それぞれの説明は次のとおりです。

- **再調整なし (no_rebalance)** : GWLB は、機能していないターゲットまたは登録解除されたターゲットにネットワークフローを送信し続けます。
- **再調整 (rebalance)** : 既存のターゲットが機能していないか登録解除された場合、GWLB はネットワークフローを別の正常なターゲットに送信します。

AWS でのスタックの展開については、以下を参照してください。

- [AWS でのクラスタの手動展開](#)
- [CloudFormation テンプレートを使用した AWS へのスタックの展開](#)

手順

ステップ 1 AWS コンソールで、[サービス (Services)] > [EC2] に移動します。

- ステップ 2** [ターゲットグループ (Target Groups)] をクリックして、ターゲットグループのページを表示します。
- ステップ 3** ASA virtual インスタンスの IP アドレスが登録されているターゲットグループを選択して開きます。ターゲットグループの詳細ページが表示されます。
- ステップ 4** [属性 (Attributes)] メニューに移動します。
- ステップ 5** [編集 (Edit)] をクリックして属性を編集します。
- ステップ 6** [フローの再調整 (Rebalance Flows)] スライダボタンを右に切り替えます。これによりターゲットフェールオーバーが有効になり、ターゲットフェールオーバーや登録解除の際に既存のネットワークパケットを再調整して正常なターゲットノードに転送するように GWLB が設定されます。
-

Azure でクラスタを展開する

Azure サービスチェーンでは、ASA virtual がインターネットと顧客サービス間のパケットをインターセプトできる透過的なゲートウェイとして機能します。Azure での ASA virtual インスタンスのクラスタリングは、単一のデバイスとして抽象化することにより、マルチノード ASAv のスループットをスケールアップするのに役立ちます。

ASAv は、2 つの論理インターフェイスで構成されています。インターネットに向けた**外部インターフェイス**と、顧客サービスに向けた**内部インターフェイス**です。これらのインターフェイスは、ペアリングされたプロキシで VXLAN セグメントを利用することにより、ASAv の単一のネットワーク インターフェイス カード (NIC) で定義されます。

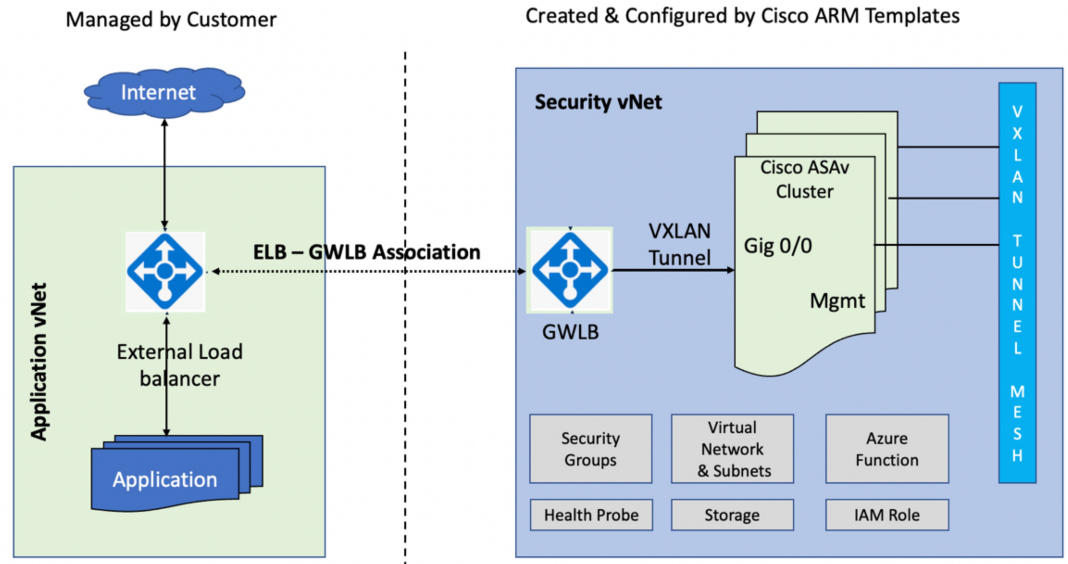
Azure Gateway Load Balancer について

Azure Gateway Load Balancer (GWLB) は、トラフィック検査のために VXLAN セグメントを介して ASAv にルーティングすることにより、着信トラフィックと発信トラフィックのバランスを取り、管理するのに役立ちます。ASAv クラスタ環境では、Azure GWLB はトラフィックの負荷に応じて ASAv ノードのスループットレベルを自動的にスケールアップします。GWLB を使用すると、ルートを手動で更新することなく、ネットワーク仮想アプライアンスへの対称フローまたは一貫性のあるルートを確保できます。この機能により、パケットは同じネットワークパスを双方向で通過できます。

次の図は、外部 VXLAN セグメント上のパブリック ゲートウェイ ロードバランサから Azure GWLB に転送されるトラフィックを示しています。**ゲートウェイロードバランサ**は、まず複数の ASAv の間でトラフィックのバランスを取り、トラフィックをドロップするか、内部の VXLAN セグメント上の GWLB に送り返す前に検査します。Azure GWLB は、トラフィックをパブリック ゲートウェイ ロードバランサと宛先に送り返します。

次の図は、Azure の GWLB と ASAv 間のネットワークフローを示しています。

図 4: GWLB を使用した Azure での ASA 仮想クラスタリング



Azure でのクラスタ展開について

カスタマイズされた Azure Resource Manager (ARM) テンプレートを使用して、Azure GWLB の仮想マシンスケールセットを展開できます。

クラスタ展開の後、デイズロ設定を使用して、または Azure portal の Function アプリを介して、クラスタ上の各ノードを手動で設定できます。

Azure Resource Manager テンプレートを使用したクラスタの展開

Azure Resource Manager (ARM) テンプレートを使用してクラスタを形成するようにクラスタノード（仮想マシンスケールセット）を展開します。

始める前に

- Azure クラスタを手動で作成するには、デイズロ設定で構成テキストファイルを準備する必要があります。「[Prepare Configuration File for Cluster deployment on Azure](#)」を参照してください。

手順

ステップ 1 テンプレートを準備します。

- GitHub リポジトリをローカルフォルダに複製します。<https://github.com/CiscoDevNet/cisco-asav/tree/master/cluster/azure> を参照してください。

- b) GWLB の場合、`azure_asav_gwlb_cluster.json` と `asav-gwlb-cluster-config.txt` を必要なパラメータで変更します。

ステップ 2 Azure ポータル : <https://portal.azure.com> にログインします。

ステップ 3 リソースグループを作成します。

[Home](#) > [Resource groups](#) >

Create a resource group ...

Basics Tags Review + create

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)

Project details

Subscription * ⓘ

MSDN Dev/Test Pay-As-You-Go(Converted to EA)

Resource group * ⓘ

asav-cluster-demo ✓

Resource details

Region * ⓘ

(US) East US

ステップ 4 ASAv クラスタの管理、外部、およびクラスタ制御リンク（CCL）の 3 つのサブネットを持つ仮想ネットワークを作成します。

[Home](#) > [Resource groups](#) > [asav-cluster-demo](#) > [Marketplace](#) > [Virtual network](#) >

Create virtual network ...

[Basics](#) IP Addresses Security Tags Review + create

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. [Learn more about virtual network](#)

Project details

Subscription * ⓘ

MSDN Dev/Test Pay-As-You-Go(Converted to EA) ▼

Resource group * ⓘ

asav-cluster-demo ▼

[Create new](#)

Instance details

Name *

asav-cluster-vnet ✓

Region *

East US ▼

[Review + create](#)[< Previous](#)[Next : IP Addresses >](#)[Download a template for automation](#)

ステップ 5 サブネットを追加します。

[Home](#) > [Resource groups](#) > [asav-cluster-demo](#) > [Marketplace](#) > [Virtual network](#) >

Create virtual network ...

Basics IP Addresses Security Tags Review + create

The virtual network's address space, specified as one or more address prefixes in CIDR notation (e.g. 192.168.1.0/24).

IPv4 address space

10.0.0.0/16 10.0.0.0 - 10.0.255.255 (65536 addresses)



Address space '10.0.0.0/16 (10.0.0.0 - 10.0.255.255)' overlaps with address space '10.0.0.0/16 (10.0.0.0 - 10.0.255.255)' of virtual network 'waweb-eastu-4034838410-vnet'. Virtual networks with overlapping address space cannot be peered. If you intend to peer these virtual networks, change address space '10.0.0.0/16 (10.0.0.0 - 10.0.255.255)'. [Learn more](#)

☐ Add IPv6 address space

The subnet's address range in CIDR notation (e.g. 192.168.1.0/24). It must be contained by the address space of the virtual network.

[+](#) Add subnet Remove subnet

☐ Subnet name	Subnet address range	NAT gateway
☐ Management	10.0.0.0/24	-
☐ Data	10.0.1.0/24	-
☐ Ccl	10.0.2.0/24	-

A NAT gateway is recommended for outbound internet access from subnets. Edit the subnet to add a NAT gateway. [Learn more](#)

[Review + create](#)

[< Previous](#)

[Next : Security >](#)

[Download a template for automation](#)

ステップ 6 カスタムテンプレートを展開します。

- [作成 (Create)] >> [テンプレートの展開 (Template deployment)] (カスタムテンプレートを使用して展開) をクリックします。
- [エディタでテンプレートを作成する (Build your template in the editor)] をクリックします。
- [ファイルのロード (Load File)] をクリックし、Azure で使用しているロードバランサタイプに応じて、azure_asav_gwlb_cluster.json。
- [保存 (Save)] をクリックします。

ステップ 7 インスタンスの詳細を設定します。

ステップ 8 必要な値を入力し、[確認して作成 (Review + create)] をクリックします。

[Home](#) > [Microsoft.VirtualNetwork-20230119131203 | Overview](#) > [asav-cluster-vnet](#) > [asav-cluster-demo](#) > [Marketplace](#) > [Template deployment \(deploy using custom templates\)](#) >

Custom deployment

Deploy from a custom template

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	MSDN Dev/Test Pay-As-You-Go(Converted to EA)
Resource group *	asav-cluster-demo

[Create new](#)

Instance details

Region *	(US) East US
Resource Name Prefix	asavcluster
Virtual Network Rg	asav-cluster-demo
Virtual Network Name	asav-cluster-vnet
Mgmt Subnet	Management
Data Interface Subnet	Data
Gateway Load Balancer IP	10.0.2.4
Ccl Subnet	Ccl
Internal Port Number	2000
External Port Number	2001
Internal Segment Id	800
External Segment Id	801

[Review + create](#) [< Previous](#) [Next : Review + create >](#)

ステップ 9 検証に合格したら、[作成 (Create)] をクリックします。

Custom deployment ...

Deploy from a custom template

✓ Validation Passed

If any Microsoft products are included in a Marketplace offering (e.g. Windows Server or SQL Server), such products are licensed by Microsoft and not by any third party.

Basics

Subscription	MSDN Dev/Test Pay-As-You-Go(Converted to EA)
Resource group	sumis-asav-clustering
Region	East US
Resource Name Prefix	asacuster
Virtual Network Rg	asav-demo-clustering
Virtual Network Name	asav-clustering-vnet
Mgmt Subnet	Mgmt
Data Interface Subnet	Data
Gateway Load Balancer IP	172.23.2.4
Ccl Subnet	CCL
Internal Port Number	2000
External Port Number	2001
Internal Segment Id	800
External Segment Id	801
Cluster Group Name	asav-gwlb-cluster
Image Id	/subscriptions/33d2517e-ca88-46aa-beb2-74ff1dd61b41/resourceGroups/su...
Vm Size	Standard_D3_v2
Asa Admin User Name	cisco
Asa Admin User Password	*****
Asav Node Count	4
Asav Config File Url	https://asavconfigs.blob.core.windows.net/asav-configfiles/asav-configurati...

Create

< Previous

Next

ステップ 10 インスタンスの実行後、いずれかのノードにログインし、**show cluster info** コマンドを入力して、クラスタの展開を確認します。

```
> show cluster info
Cluster gwlb-cluster-template-with-AN: On
Interface mode: individual
Cluster Member Limit : 16
This is "12" in state CONTROL_NODE
ID       : 0
Version  : 99.19(1)180
Serial No.: 9AKGFV8VH4G
CCL IP   : 10.1.1.12
CCL MAC  : 000d.3a55.5470
Module   : NGFWv
Resource : 8 cores / 28160 MB RAM
Last join : 11:13:24 UTC Sep 5 2022
Last leave: N/A
```

次のタスク

[Azure でクラスタを設定する \(41 ページ\)](#)。

Azure でクラスタを設定する

Azure の ASAv ノードでクラスタを設定するには、構成ファイルを使用するか、Azure Function App を使用して手動で設定できます。ネイティブ GWLB でクラスタを使用できます。

Azure でクラスタを作成するための構成ファイルを準備する

構成ファイルまたは Azure portal の Function App を使用して、ASA 仮想 ノードでクラスタを手動で設定できます。

ASA 仮想 ノードでクラスタを手動で設定するには、asav-gwlb-cluster-config.txt を設定しておく必要があります。このファイルでは、クラスタの ASA 仮想 ノードで設定されている範囲オブジェクト、デイズロ、クラスタグループ名、ライセンスタイプなどのパラメータを定義する必要があります。

このセクションでは、GWLB を使用して Azure の ASA 仮想 ノードを設定するためのクラスタの構成ファイルの作成について説明します。

手順

ステップ 1 Cisco GitHub リポジトリディレクトリ asav-cluster/sample-config-file から **asav-gwlb-cluster-config.txt** をダウンロードします。

ステップ 2 クラスタ作成のためにデイズロ設定を準備できます。

次の day0 構成のサンプルは、GWLB を使用した Azure でのクラスタ作成に必要なパラメータを理解するのに役立ちます。

• GWLB クラスタ作成用のデイズロ設定のサンプル

以下は、GWLB クラスタ作成に使用される asav-gwlb-cluster-config.txt ファイルに必要なデイズロ設定のサンプルです。

```
cluster interface-mode individual force
  policy-map global_policy
  class inspection_default
  no inspect h323 h225
  no inspect h323 ras
  no inspect rtsp
  no inspect skinny

interface GigabitEthernet0/0
  nameif vxlan_tunnel
  security-level 0
  ip address dhcp
  no shutdown

interface GigabitEthernet0/1
  nve-only cluster
```

```

        nameif ccl_link
        security-level 0
        ip address dhcp
        no shutdown

interface vni1
    description ClusterInterface
    segment-id 1
    vtep-nve 1

interface vni2
    proxy paired
    nameif GWLB-backend-pool
    internal-segment-id 800
    external-segment-id 801
    internal-port 2000
    external-port 2001
    security-level 0
    vtep-nve 2

object network ccl#link
    range <CCLSubnetStartAddress> <CCLSubnetEndAddress>
    object-group network cluster#group
    network-object object ccl#link

nve 1
    encapsulation vxlan
    source-interface ccl_link
    peer-group cluster#group

nve 2
    encapsulation vxlan
    source-interface vxlan_tunnel
    peer ip <GatewayLoadbalancerIp>

mtu vxlan_tunnel 1454
mtu ccl_link 1374
cluster group <ClusterGroupName>
local-unit <Last Octet of CCL Interface IP>
cluster-interface vni1 ip 169.254.200.<Last Octet of CCL Interface IP>
255.255.255.0
priority 1
enable

```

上記のデイズロ設定のサンプルでは、カプセル化タイプが **vxlan** としてメンションされている場合、GWLB 関連の設定が有効になります。**InternalPort** と **ExternalPort** は vxlan トンネルインターフェイスの設定に使用され、**InternalSegId** と **ExternalSegId** は内部および外部インターフェイスの識別子として使用されます。

(注)

デイズロ設定では、開始アドレス (<CCLSubnetStartAddress>) とクラスタ制御リンクの終了アドレスを指定する必要があります。したがって、StartAddress は常に x.x.x.4 で開始する必要があり、EndAddress は最適な範囲内にある必要があります。広範囲のアドレスを追加するとパフォーマンスに影響する可能性があるため、必要な数のアドレス (最大 16) のみを指定することをお勧めします。

例: CCL サブネットが 192.168.3.0/27 の場合、StartAddress は 192.168.3.4 になり、EndAddress は 192.168.3.30 になります。

すべてのクラスタ インフラストラクチャ サブネットは /27 CIDR を使用する必要があります

以下は、vni インターフェイスに必要な設定のサンプルです。

```
interface vni2
  proxy paired
  nameif GWLB-backend-pool
  internal-segment-id 800
  external-segment-id 801
  internal-port 2000
  external-port 2001
  security-level 0
  vtep-nve 2
```

- ステップ 3** 構成ファイルを Azure ストレージにアップロードし、この場所のパス（URL）をメモします。この URL パスは、ASA 仮想 ノードでのクラスタの手動設定に必要です。

構成ファイルを使用して手動でクラスタを設定する

構成ファイルを使用して、Azure の ASAv ノードでクラスタを手動で構成するには以下の手順を実行します。

始める前に

構成ファイルを準備し、アップロード先の Azure ストレージの場所をメモしておく必要があります。「Prepare Cluster Configuration File for Azure」を参照してください。

手順

- ステップ 1** Azure ポータルにログインします。

- ステップ 2** Azure に展開された ASAv インスタンスを開きます。

- ステップ 3** 次のコマンドを実行して、Azure ストレージコンテナにアップロードしたファイルの URL を指定して、クラスタ構成ファイルを ASAv ノードにコピーします。

copy <Config File URL> running-config

- ステップ 4** 次のコマンドを実行して、ASAv インスタンスでクラスタを設定します。

```
cluster group <ClusterGroupName>
  local-unit <Last Octet of the Management Interface IP>
  cluster-interface vni1 ip 169.254.200.<Last Octet of the Management Interface
IP> 255.255.255.0
  priority 1
  enable
```

- ステップ 5** 手順 2 ～ 4 を繰り返して、すべての ASAv ノードでクラスタを設定します。

Azure Function App を使用してクラスタを設定する

Azure Function App サービスを使用して Azure の ASAv ノードでクラスタを設定するには、以下の手順を実行します。

手順

- ステップ 1 Azure ポータルにログインします。
- ステップ 2 [Function App] をクリックします。
- ステップ 3 [展開センター (Deployment Center)] > [FTPSの資格情報 (FTPS credentials)] > [ユーザースコープ (User scope)] > [ユーザー名とパスワードの設定 (Configure Username and Password)] > をクリックして FTPS の資格情報を作成し、[保存 (Save)] をクリックします。

The screenshot shows the 'FTPS credentials' settings page in the Azure Portal. At the top, there are links for 'Save', 'Discard', 'Browse', 'Manage publish profile', 'Sync', and 'Leave Feedback'. Below these, the 'Settings' tab is active, and 'FTPS credentials' is selected. The page explains that App Service supports multiple technologies to access, publish, and modify the content of your app, and that FTPS credentials can be scoped to the application or the user.

FTPS endpoint: A text box contains the URL 'https://waws-prod-blu-231.ftp.azurewebsites.windows.net/site/wwwroot'.

Application scope: This section explains that application scope credentials are auto-generated and provide access only to this specific app or deployment slot. It lists supported technologies: FTPS, Local Git, and WebDeploy. It notes that these credentials cannot be configured manually but can be reset anytime. A 'Learn more' link is provided.

Username: A text box contains 'gwlban-function-app\gwlban-function-app'.

Password: A masked password field with a 'Reset' button to its right.

User scope: This section explains that user scope credentials are defined by you, the user, and can be used with all the apps to which you have access. It lists supported technologies: FTPS, Local Git, and WebDeploy. It provides the format for the username: 'gwlban-function-app\sumis'. It also notes that authenticating with Git requires only the username 'sumis'. A 'Learn more' link is provided.

Username: A text box contains 'sumis'.

Password: A masked password field.

Confirm Password: A masked password field.

- ステップ 4 ローカルのターミナルで次のコマンドを実行して、Cluster_Function.zip ファイルを関数アプリにアップロードします。

```
curl -X POST -u <Userscope_Username> --data-binary @"Cluster_Function.zip"
https://<Function_App_Name>.scm.azurewebsites.net/api/zipdeploy
```

図 5: 機能

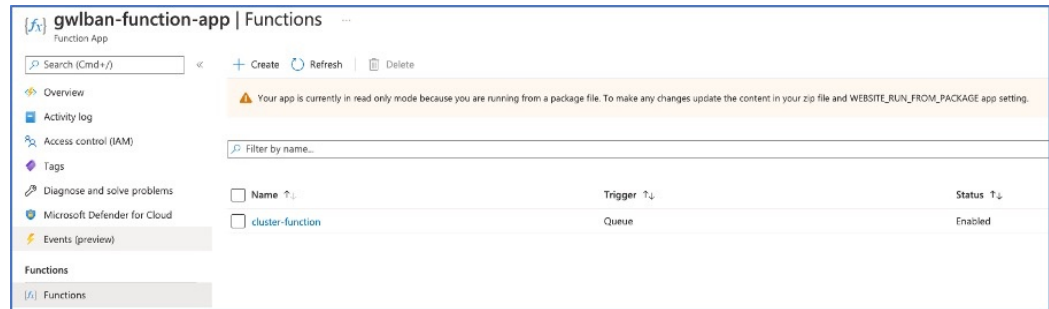


図 6: キュー

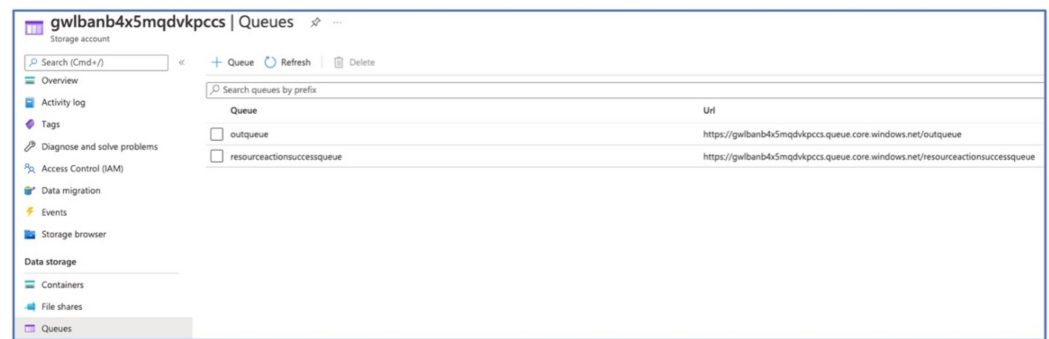
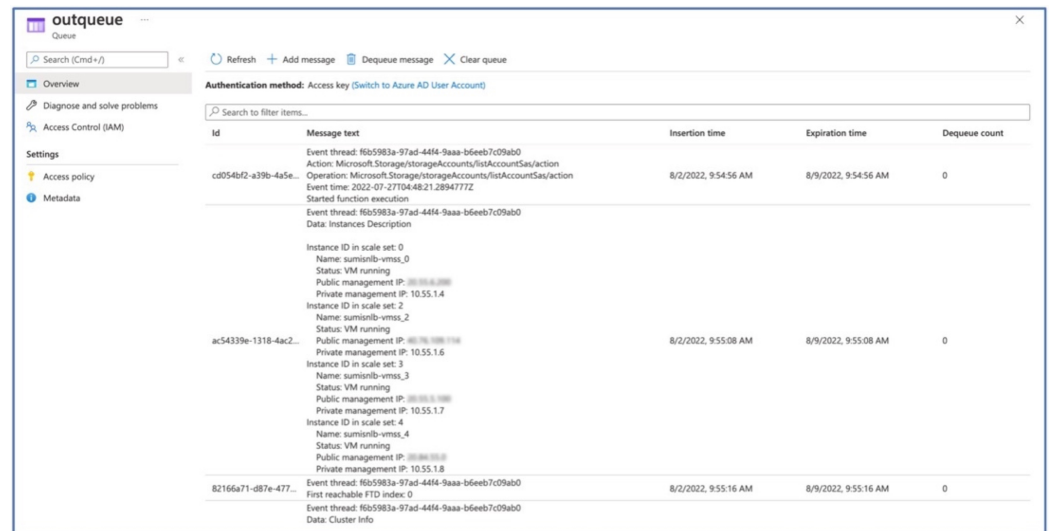


図 7: アウトキュー



関数が Function アプリにアップロードされます。関数が開始され、ストレージアカウントのアウトキューにログが表示されます。

関数の実行後、クラスタはすべての ASA v ノードで有効になります。

Azure における ASA Virtual クラスタのトラブルシューティング

トラフィックの問題

トラフィックが機能していない場合は、次のことを確認します。

1. ロードバランサを使用した ASA virtual インスタンスの正常性プローブのステータスが正常であることを確認します。

ASA virtual インスタンスの正常性プローブのステータスが正常でない場合は、次を実行します。

1. ASA virtual で設定されたスタティックルートを確認します。
2. デフォルトゲートウェイがデータサブネットのゲートウェイ IP であることを確認します。
3. ASA virtual インスタンスが正常性プローブトラフィックを受信していることを確認します。
4. ASA virtual で設定されているアクセスポリシーが正常性プローブトラフィックを許可していることを確認します。

クラスタの問題

クラスタが形成されていない場合は、次のことを確認してください。

- ネットワーク仮想化エンドポイント (NVE-only) のクラスタインターフェイスの IP アドレス。他のノードの NVE-only のクラスタインターフェイスに ping を実行できることを確認します。
- NVE-only のクラスタインターフェイスの IP アドレスは、オブジェクトグループの一部です。NVE がオブジェクトグループで設定されていることを確認します。
- クラスタグループのクラスタインターフェイスに正しい VNI インターフェイスがあります。この VNI インターフェイスには、対応するオブジェクトグループの NVE があります。
- 各ノードには独自の IP インターフェイスがあります。クラスタ内のノード間の接続を確保するために、ノードが相互に ping できることを確認します。
- テンプレートのデプロイ中に指定された CCL サブネットの開始アドレスと終了アドレスが正しいことを確認します。開始アドレスは、サブネット内で使用可能な最初の IP アドレスで始まる必要があります。たとえば、サブネットが 192.168.1.0/27 の場合、開始アドレスは 192.168.1.4 である必要があります（最初の 3 つの IP アドレスは azure によって予約されています）。

ロールに関連する問題

同じリソースグループに再度リソースを展開しているときにロールに関連するエラーが発生した場合は、次の手順を実行します。

特定のロールに関連する問題がある場合、エラーメッセージが表示されます。

エラーメッセージの例を次に示します。

```
"error": {  
  "code": "RoleAssignmentUpdateNotPermitted",  
  "message": "Tenant ID, application ID, principal ID, and scope are not allowed to be updated."}
```

ターミナルから以下のコマンドを実行して、以下のロールを削除します。

- ストレージキューデータ寄稿者ロールを削除するコマンド：

```
az role assignment delete --resource-group <Resource Group Name> --role "Storage Queue  
Data Contributor"
```

- 寄稿者ロールを削除するコマンド：

```
az role assignment delete --resource-group <Resource Group Name> --role "Contributor"
```

Azure での ASA 仮想 クラスタリング自動スケールソリューション

Azure リージョンでの一般的なクラスタ展開には、定義された数の ASA 仮想 インスタンス（ノード）が含まれます。Azure リージョンのトラフィックが変化しても、ノードのダイナミックスケールリング（自動スケール）が行われず、以前からのクラスタ配置のままだと、リソースが十分に活用されなかったり、遅延を引き起こしたりします。Cisco は、Azure リージョンのノードのダイナミックスケールリングをサポートする ASA 仮想 クラスタリング向けに自動スケールソリューションをバージョン 9.23 以降で提供しています。このソリューションにより、ネットワークトラフィックに基づいてクラスタからノードを追加または削除して、スケールインまたはスケールアウトを行えます。CPU やメモリのメトリックなどの Azure VMSS メトリックからのリソース使用率の統計に基づくロジックを使用して、クラスタに対してノードを動的に追加または削除します。

Azure の自動スケールソリューションを使用した ASA 仮想 クラスタリングは、ネットワークロードバランサ（NLB または サンドイッチトポロジ）とゲートウェイロードバランサ（GWLБ）の両方をサポートします。[トポロジの例（48 ページ）](#) を参照してください

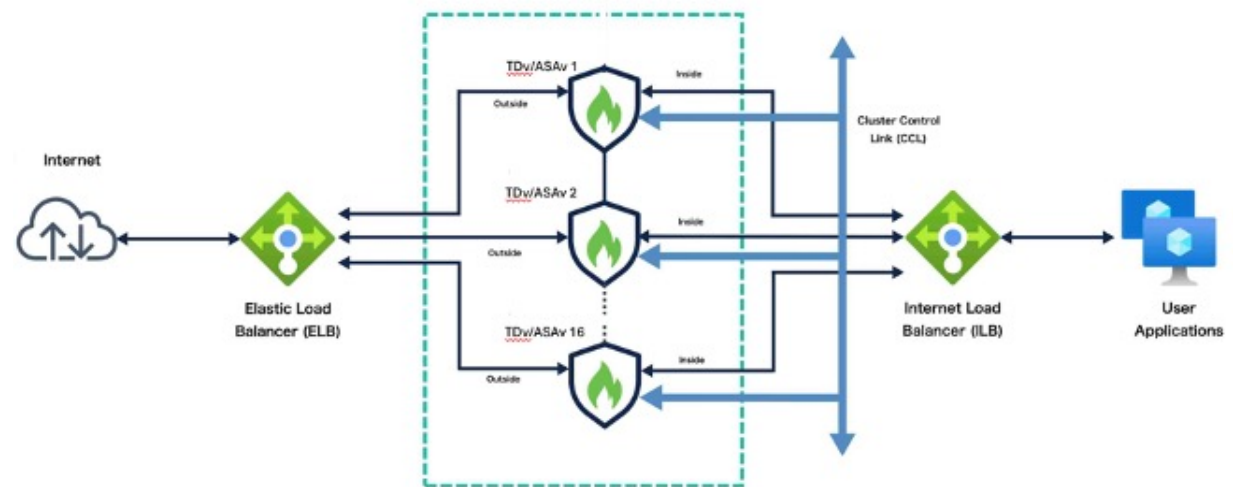
Cisco では、NLB や GWLB を使用して ASA 仮想 クラスタを Azure に自動スケールして展開するための個別の Azure Resource Manager（ARM）テンプレートと、関数アプリや論理アプリなど、Azure サービスを展開するためのインフラストラクチャテンプレートと構成テンプレートを提供しています。

トポロジの例

ASA 仮想 サンドイッチトポロジ（ネットワーク ロードバランサ）を使用した Azure での自動スケールのクラスタリング

サンドイッチトポロジ（NLB）を使用する Azure の自動スケール対応 ASA 仮想 クラスタリングのユースケースは、Azure の内部ロードバランサ（ILB）と Azure の外部ロードバランサ（ELB）にサンドイッチされるように ASA 仮想 スケールセットを配置する、自動水平スケールリングソリューションです。

このトポロジでは、ASA 仮想 は、管理、内部、外部、および CCL サブネットの 4 つのインターフェイスのみを使用します。



サンドイッチトポロジ（NLB）を使用した Azure での自動スケールによる ASA 仮想 クラスタリング

以下では、NLB 機能を使用して Azure での自動スケールを行う ASA 仮想 クラスタのフローの概要を示します。

- ELB は、インターネットからのトラフィックをスケールセット内の ASA 仮想 インスタンスに分散させます。その後、ファイアウォールがアプリケーションにトラフィックを転送します。
- ILB は、アプリケーションからのアウトバウンドインターネットトラフィックをスケールセット内の ASA 仮想 インスタンスに分散させます。その後、ファイアウォールがインターネットにトラフィックを転送します。
- ネットワークパケットが、単一の接続で両方（内部および外部）のロードバランサを通過することはありません。
- スケールセット内の ASA 仮想 インスタンスの数は、負荷条件に基づいて自動的にスケールリングおよび設定されます。

ゲートウェイロードバランサを使用した Azure での自動スケールによる ASA 仮想 クラスタリング

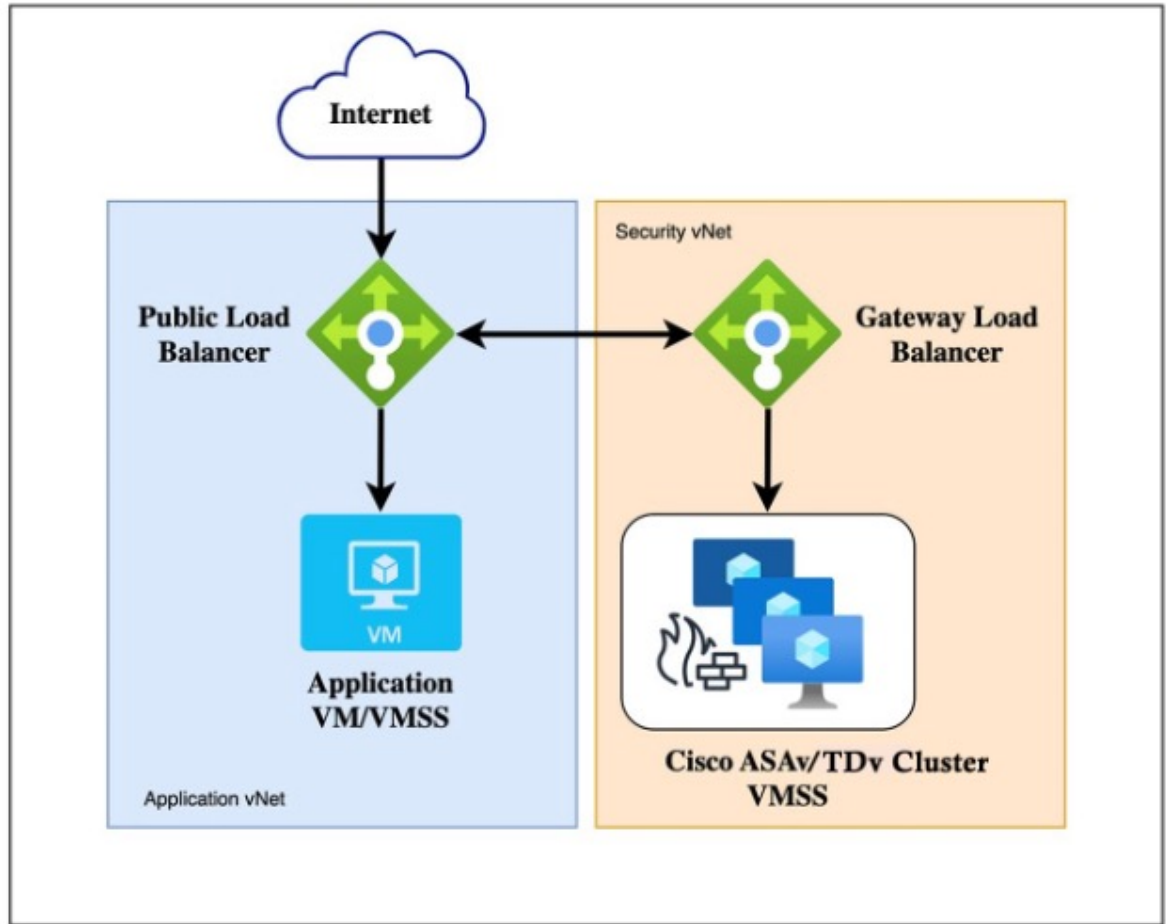
自動スケールソリューションを使用した Azure ゲートウェイロードバランサ (GWLB) と ASA 仮想 クラスタの統合により、クラスタセットアップでのインスタンスの展開、管理、およびスケーリングが簡素化されます。Azure ゲートウェイロードバランサ (GWLB) は、アプリケーションサーバーなどの Azure VM との間のインターネットトラフィックが、ルーティングの変更を必要とせずに Cisco Secure Firewall によって検査されるようにします。また、この統合により、運用の複雑さが軽減され、ファイアウォールでのトラフィックの単一のエントリポイントとエグジットポイントが提供されます。アプリケーションとインフラストラクチャは、送信元 IP アドレスの可視性を維持できます。一部の環境では、この可視性が非常に重要です。

ASA 仮想 は、この使用例では、管理、データ、CCL インターフェイスの 3 つのインターフェイスのみを使用します。



(注)

- Azure GWLB を展開する場合、ネットワークアドレス変換 (NAT) は必要ありません。
- IPv4 だけがサポートされます。



以下では、GWLB 機能を使用して、Azure で自動スケールを行う ASA 仮想クラスタのフローの概要を説明します。

- インターネットからの着信トラフィックは、GWLB エンドポイントに送られ、そこから GWLB にトラフィックが送信されます。
- その後、トラフィックは ASA 仮想 クラスタにルーティングされます。自動スケールソリューションでは、スケールインまたはスケールアウトのロジックを適用して、トラフィック負荷に基づいてクラスタからノードを追加または削除します。
- トラフィックは、クラスタ内の ASA 仮想 インスタンスによって検査された後、アプリケーション VM に転送されます。

Azure での ASA 仮想 クラスタリングの自動スケールロジック

スケーリングポリシー

自動スケールを備えたクラスタでは、ノードのスケーリングは次のポリシーに基づいて決定されます。

- スケーリングポリシー 1：1 つのクラスタノードがリソース使用率の制限を超えている。
- スケーリングポリシー 2：すべてのノードの全体的な平均リソース使用率。

スケールアウト

スケールアウトとは、トラフィック負荷のしきい値がクラスタのいずれかのノードに設定されている CPU またはメモリの制限を超えたときに、クラスタに新しいノードを追加するプロセスです。

次に、スケールアウト中にクラスタに新しいノードを追加するプロセスを示します。

1. 新しい ASA 仮想 インスタンスが起動します。
2. ASA 仮想 に適切な設定が適用されます。
3. 適切なライセンスが適用されます。
4. 新しい ASA 仮想 インスタンスがクラスタに追加されます。

スケールアウトプロセス中に新しい ASA 仮想 インスタンスの設定が失敗した場合（確率は低い）、失敗したインスタンスは終了し、新しいインスタンスが起動して設定されます。

スケールイン

スケールインは、設定されたスケールインしきい値に達した場合、およびクラスタインスタンスの合計数が最小クラスタサイズを超えた場合に、クラスタからノードを削除するプロセスです。

次に、スケールイン中にクラスタ内のノードを終了するプロセスを示します。

1. CPU またはメモリ使用率が最も低い ASA 仮想 インスタンスを、VMSS メトリックを使用して識別します。
2. 使用率が同じ最小のインスタンスが複数ある場合、VMSS の VM インデックスが高いインスタンスがスケールイン用に選択されます。
3. このインスタンスへの新しい接続は、適切な設定とポリシーによって無効になります。
4. インスタンスがスマートライセンスから登録解除されます（BYOL に該当）。
5. インスタンスが終了します。

Azure 関数 (Function App)

Function アプリケーションは、ASA 仮想 クラスタの有効化と登録に役立ちます。Function アプリケーションは、自動スケール展開を使用した ASA 仮想 クラスタリングのホスティングプランを選択するのに役立ちます。

次の 2 種類のホスティングプランが提供されています。

• 消費

- これは、自動スケールを使用した ASA 仮想 クラスタリングのデフォルトのホスティングプランです。
- このプランでは、リージョンの Azure データセンター IP アドレスへの SSH ポートを開くことにより、Function アプリが ASA 仮想 インスタンスに接続できます。

• プレミアム

- 展開時に Function アプリに対してこのホスティングプランを選択できます。
- このプランでは、Function アプリにネットワークアドレス変換 (NAT) ゲートウェイを追加して、Function アプリのアウトバウンド IP アドレスを制御できます。このプランでは、NAT ゲートウェイの固定 IP アドレスからのみ ASA 仮想 インスタンスへの SSH アクセスを許可するため、セキュリティが強化されます。

GitHub での展開とインフラストラクチャのテンプレート

Cisco では、Function App、Logic App、自動スケールリンググループなどの複数の Azure サービスを使用して ASA 仮想 クラスタの Virtual Machine Scale Set (VMSS) を展開するための Azure Resource Manager (ARM) テンプレートとスクリプトを提供しています。

ASA 仮想 クラスタの自動スケールソリューションは、GWLB および NLB ロードバランサのサポートを提供する ARM テンプレートベースの展開です。

自動スケール ソリューション テンプレートを使用した ASA 仮想 クラスタリング

Azure Resource Manager (ARM) テンプレート

クラスタに Azure で使用している (NLB または GWLB) ロードバランサに基づいて、自動スケールソリューション用に 2 セットのテンプレートが用意されています。

GitHub では、次のテンプレートを使用できます。

- NLB を使用した ASA 仮想 クラスタリングの自動スケールソリューションテンプレート : `azure_autoscale_clustering/asav_cluster/arm_templates/` フォルダにある `azure_asav_nlb_cluster_autoscale.json`
- GWLB を使用した ASA 仮想 クラスタリングの自動スケール ソリューションテンプレート : `azure_autoscale_clustering/asav_cluster/arm_templates/` フォルダにある `azure_asav_gwlb_cluster_autoscale.json`

GitHub 上の Azure インフラストラクチャおよび構成テンプレート

以下は、Azure での自動スケールを使用したクラスタリング用の Azure インフラストラクチャの設定に必要なテンプレートです。

- ASA 仮想 インスタンスでクラスタを有効にする Function app :
azure_autoscale_clustering/asav_cluster/azure_function_app フォルダにある cluster_functions.zip。
- ASA 仮想の展開、スケールイン、およびスケールアウトのワークフローに使用する Logic App コード : azure_autoscale_clustering/asav_cluster/logic_app/ フォルダにある logic_app.txt。

入力パラメータ

次の表に、テンプレート パラメータおよび例を示します。各パラメータの値を決定すると、Azure サブスクリプションに Azure Resource Manager (ARM) テンプレートを展開するときに、それらのパラメータを使用して ASA 仮想を作成できます。Azure 向けの GWLB を使用した自動スケールソリューションによるクラスタリングでは、テンプレートで追加の入力パラメータを設定する必要があるため、ネットワーキング インフラストラクチャも作成されます。パラメータの意味は一目瞭然なので説明を省略します。

表 2: テンプレートパラメータ

パラメータ名	使用できる値/タイプ	説明	リソースの作成タイプ
resourceNamePrefix	文字列* (3 ~ 10 文字)	すべてのリソースは、このプレフィックスを含む名前で作成されます。 注：小文字のみを使用してください。 例：asav	新規作成
virtualNetworkRg	文字列	仮想ネットワークのリソースグループの名前。 例：cisco-virtualnet-rg	既存
virtualNetworkName	文字列	仮想ネットワーク名（作成済み） 例：cisco-virtualnet	既存
mgmtSubnet	文字列	管理サブネット名（作成済み） 例：cisco-mgmt-subnet	既存

パラメータ名	使用できる値/ タイプ	説明	リソースの作成 タイプ
dataSubnet	文字列	データサブネット名（作成済み） 例：cisco-data-subnet	
cclSubnet	文字列	クラスタ制御リンクのサブネット名。 例：cisco-ccl-subnet	
cclSubnetStartAddr	文字列	CCL サブネット IP アドレスの範囲開始。 例：3.4.5.6	
cclSubnetEndAddr	文字列	CCL サブネット IP アドレスの範囲終了。 例：5.6.7.8	
gwlbIP	文字列	GWLB は既存のデータサブネットに作成されます。 例：10.0.2.4	
dataNetworkGatewayIp	文字列	データサブネットのゲートウェイ IP アドレス。 例：10.0.2.7	
insideSubnet	文字列	内部サブネット名（作成済み） 例：cisco-inside-subnet	既存
internalLbIp	文字列	内部サブネットの内部ロードバランサの IP アドレス（作成済み）。 例：1.2.3.4	既存
outsideSubnet	文字列	外部サブネット名（作成済み） 例：cisco-outside-subnet	既存
softwareVersion	文字列	ASA 仮想バージョン（展開時にドロップダウンリストから選択）。	既存

パラメータ名	使用できる値/ タイプ	説明	リソースの作成タイプ
vmSize	文字列	ASA 仮想インスタンスのサイズ（展開時にドロップダウンリストから選択）。	該当なし
asaAdminUserName	文字列 *	ASA 仮想「admin」ユーザーのユーザー名。 これは「admin」にはできません。VM 管理者ユーザー名のガイドラインについては、「Azure」を参照してください。 （注） テンプレートには、このパラメータのコンプライアンスチェック機能はありません。	新規作成
asaAdminUserPassword	文字列 *	ASA 仮想管理者ユーザのパスワード。 パスワードの長さは 12 ～ 72 文字で、小文字、大文字、数字、特殊文字を使用する必要があります。また、文字の繰り返しは 2 回までにする必要があります。 （注） テンプレートには、このパラメータのコンプライアンスチェック機能はありません。	新規作成
clusterGroupName	文字列	Cisco ASA Virtual デバイスに登録するときに使用されるクラスタグループの名前。 例：asav-cluster	
asaLicensingSku	文字列	Cisco ASA Virtual のライセンスモード（PAYG または BYOL）。	

パラメータ名	使用できる値/ タイプ	説明	リソースの作成タイプ
healthCheckPortNumber	文字列	ゲートウェイロードバランサーで正常性プローブを作成するときに使用されるヘルスチェックのポート番号。 例：8080	
functionHostingPlan	文字列	機能展開のホスティングプラン（ consumption は消費ホスティングプランを使用、 premium はプレミアム ホスティング プランを使用）。 デフォルト：consumption	
functionAppSubnet	文字列	関数アプリサブネット名（作成済み）。 例：asav-fapp-subnet	
functionAppSubnetCIDR	文字列	関数アプリサブネットのCIDR（作成済み）。 例：10.0.4.0/27	
scalingMetricsList	文字列	スケーリングの決定に使用されるメトリック。 許可：CPU	

パラメータ名	使用できる値/ タイプ	説明	リソースの作成タイプ
scalingPolicy	POLICY-1/POLICY-2	POLICY-1：設定された期間に、いずれかの ASA 仮想の平均負荷がスケールアウトしきい値を超えるとスケールアウトがトリガーされます。 POLICY-2：設定された期間に、VMSS のすべての ASA 仮想 デバイスの平均負荷がスケールアウトしきい値を超えるとスケールアウトがトリガーされます。 どちらの場合も、スケールインロジックは同じままです。設定された期間に、すべての ASA 仮想 デバイスの平均負荷がスケールインしきい値を下回るとスケールインがトリガーされます。	該当なし
scalingMetricsList	文字列	スケーリングの決定に使用されるメトリック。 許可：CPU デフォルト：CPU	該当なし
scaleInThreshold	文字列	スケールインしきい値（パーセント単位）。 デフォルト：10 ASA 仮想 メトリックがこの値を下回ると、スケールインがトリガーされます。	該当なし

パラメータ名	使用できる値/ タイプ	説明	リソースの作成タイプ
scaleOutThreshold	文字列	スケールアウトしきい値（パーセント単位）。 デフォルト：80 ASA 仮想メトリック（CPU 使用率）がこの値を上回ると、スケールアウトがトリガーされます。 「scaleOutThreshold」は、常に「scaleInThreshold」より大きくする必要があります。	該当なし
asavClusterSize	文字列	任意の時点でスケールセットで使用可能な Cisco ASA Virtual インスタンスのデフォルトノード数。 例：4	
minAsaCount	整数	任意の時点でスケールセットで使用可能な最小 ASA 仮想インスタンス数。 例：2。	該当なし
maxAsaCount	整数	スケールセットで許可される最大 ASA 仮想 インスタンス数。 例：10 （注） Auto Scale ロジックではこの変数の範囲はチェックされないため、慎重に入力してください。	該当なし

パラメータ名	使用できる値/ タイプ	説明	リソースの作成タイプ
metricsAverageDuration	整数	ドロップダウンから選択します。 この数値は、メトリックが平均化される時間（分単位）を表します。 この変数の値が5（5分）の場合、Auto Scale Manager がスケジュールされると、メトリックの過去5分間の平均がチェックされ、その結果に基づいてスケーリングの判断が行われます。 （注） Azure の制限により、有効な数値は1、5、15、および30だけです。	該当なし
initDeploymentMode	BULK/STEP	主に最初の展開、またはスケールセットにASA 仮想インスタンスが含まれていない場合に適用されます。 BULK : Auto Scale Manager は、「minAsaCount」個のASA 仮想インスタンスを同時に展開しようとしています。 STEP : Auto Scale Manager は、スケジュールされた間隔ごとに「minAsaCount」個のASA 仮想デバイスを1 つずつ展開します。	
smartLicenseToken	文字列	ASA 仮想を登録するためのスマートライセンストークン。	
licenseThroughput	文字列	ASA 仮想のスマートライセンスの権限付与層。	

パラメータ名	使用できる値/ タイプ	説明	リソースの作成 タイプ
asavConfigFileUrl	文字列	ASA 仮想 構成ファイルのファイルパス。 例： <code>https://path_to_asav_config_file/config_file</code> 構成ファイルが ASAv からアクセス可能であることを確認してください。	該当なし
* Azure には、新しいリソースの命名規則に関する制限があります。制限を確認するか、またはすべて小文字を使用してください。スペースやその他の特殊文字は使用しないでください。			

自動スケールを備えた ASA 仮想 クラスタの展開プロセスとリソース

自動スケールを備えた ASA 仮想 クラスタの Azure での展開プロセスには次の作業が含まれます。

- ARM テンプレートを展開します。
- クラスタリング機能を構築して展開します。
- 論理アプリケーションを更新して有効化します。

Azure リソース マネージャ テンプレート展開リソース

サンドイッチトポロジ (NLB) の ARM テンプレート

(`azure_asav_nlb_cluster_autoscale.json`) を使用して、Azure に自動スケールを使用して ASA 仮想 クラスタを展開すると、リソースグループ内に次のリソースが作成されます

- 仮想マシンスケールセット (VMSS)
- 外部ロードバランサ
- 内部ロードバランサ
- Azure Function App
- Logic App
- セキュリティグループ (データインターフェイスおよび管理インターフェイス用)

GWLB の ARM テンプレート: `azure_asav_gwlb_cluster_autoscale.json` を使用して、Azure に自動スケールを使用して ASA 仮想 クラスタを展開すると、リソースグループ内に次のリソースが作成されます

- 仮想マシン (VM) または仮想マシンスケールセット (VMSS)
- ゲートウェイロードバランサ (GWLB)

- Azure Function App
- Logic App
- ネットワーキング インフラストラクチャ
- 展開に必要なセキュリティグループおよびその他のコンポーネント。

自動スケールソリューションを使用した ASA 仮想 クラスタの展開

ARM テンプレートを使用して、Azure に自動スケールソリューションを備えた ASA 仮想 クラスタリングを展開します。トポロジ、サンドイッチ（NLB）または GWLB のユースケースに基づいて、Azure の自動スケールソリューションを使用して ASA 仮想 クラスタリングを展開するための適切な ARM テンプレートをダウンロードして構成する必要があります。

始める前に

GitHub からの展開パッケージのダウンロード

Azure 向けの NLB ソリューションを使用する ASA 仮想 クラスタリング自動スケールは、Azure Resource Manager（ARM）テンプレートベースの展開であり、Azure が提供するサーバーレス インフラストラクチャ（論理アプリ、Azure 関数、ロードバランサ、仮想マシンスケールセットなど）を使用します。

Azure 向けの GWLB ソリューションを使用する ASA 仮想 クラスタリング自動スケールは、ARM テンプレートベースの展開であり、GWLB、ネットワーク インフラストラクチャ、仮想マシンスケールセット、サーバーレスコンポーネント、および他の必要なリソースを作成します。

両方のソリューションの展開手順はほぼ同じです。

Azure 向けの自動スケールソリューションを使用する ASA 仮想 クラスタリングの起動に必要なファイルをダウンロードします。

該当するバージョン用の展開スクリプトとテンプレートは、[GitHub](#) リポジトリから入手できます。

手順

- ステップ 1** Microsoft アカウントのユーザー名とパスワードを使用して、Microsoft Azure ポータル (<https://portal.azure.com>) にログインします。
- ステップ 2** [Resource Groups] ブレードにアクセスするには、サービスのメニューから [Resource groups] をクリックします。サブスクリプション内のすべてのリソースグループがブレードに一覧表示されます。新しいリソースグループを作成するか、既存の空のリソースグループを選択します。
例：secure-firewall-asav-demo。
- ステップ 3** [リソースの作成 (+) (Create a resource (+))] をクリックして、テンプレート展開用の新しいリソースを作成します。[Create Resource Group] ブレードが表示されます。

ステップ 4 サービスのメニューから [Virtual Network] をクリックして、仮想ネットワークブレードにアクセスします。サブネットを含む仮想ネットワークを作成します。

- GWLB 展開の場合、管理、データ、および CCL サブネットを含む仮想ネットワークを作成します。
- NLB 展開の場合、管理、内部、外部、CCL のサブネットを持つ仮想ネットワークを作成します。

ステップ 5 [Search the Marketplace] で、「**Template deployment**」（カスタムテンプレートを使用した展開）と入力し、Enter を押します。

ステップ 6 [作成 (Create)] をクリックします。テンプレートを作成するためのオプションは複数あります。[エディタで独自のテンプレートを作成する (Build your own template in editor)] を選択します。

Home > Create a resource > Marketplace > Template deployment (deploy using custom templates) >

Custom deployment

Deploy from a custom template

New! Deployment Stacks let you manage the lifecycle of your deployments. Try it now →

Select a template **Basics** Review + create

Template

 Customized template 
16 resources

 Edit template  Edit parameters  Visualize

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

Resource group * ⓘ
[Create new](#)

Instance details

Region * ⓘ	East US
Resource Name Prefix ⓘ	asavgwlb
Virtual Network Rg ⓘ	secure-firewall-demo-rg
Virtual Network Name ⓘ	secure-firewall-demo-vnet
Virtual Network Cidr ⓘ	10.0.0.0/16
Mgmt Subnet ⓘ	Management
Data Interface Subnet ⓘ	Data

ステップ7 [Edit template] ウィンドウで、すべてのデフォルトコンテンツを削除し、更新した `azure_asav_gwlb_cluster_custom_image.json` または `azure_asav_nlb_cluster_custom_image.json` から（Azure に展開する自動スケールソリューションのタイプに応じて）コンテンツをコピーして、[Save] をクリックします。または、[Load file] をクリックし、コンピューターからこのファイルを参照してアップロードします。

Gateway Load Balancer IP ⓘ	10.0.1.4	✓
Data Network Gateway Ip ⓘ	10.0.1.1	✓
Ccl Subnet ⓘ	Ccl	✓
Ccl Subnet Start Addr ⓘ	10.0.3.4	✓
Ccl Subnet End Addr ⓘ	10.0.3.24	✓
Internal Port Number ⓘ	2000	✓
External Port Number ⓘ	2001	✓
Internal Segment Id ⓘ	800	✓
External Segment Id ⓘ	801	✓
Function Hosting Plan ⓘ	consumption	▼
Function App Subnet ⓘ		✓
Function App Subnet CIDR ⓘ		✓
Cluster Group Name ⓘ	asavgwlb-cls	✓
Image Id ⓘ	/subscriptions/1fd9165-db4d-4fc9-814b-8475c5adc637/resourceGro...	✓
Vm Size ⓘ	Standard_D4_v2	▼
Asa Admin User Name ⓘ	cisco	✓
Asa Admin User Password ⓘ	*****	
Scale In Threshold ⓘ	10	✓
Scale Out Threshold ⓘ	60	✓
Asav Cluster Size ⓘ	4	✓
Metrics Average Duration ⓘ	1	▼

Asav Cluster Size ①	4	✓
Metrics Average Duration ①	1	▼
Init Deployment Mode ①	BULK	▼
Scaling Policy ①	POLICY-2	▼
License Throughput ①	2G	▼
Smart License Token ①	Zjc0YTMynjQtdMDE0Yy00ZTA0LTgyZTktdZTEyNmYyNDM0MWFkLTE3Mj...	✓
Asav Config File Url ①		

Previous
Next
Review + create

ステップ 8 パラメータ フィールド セクションで、すべてのパラメータを入力します。各パラメータの詳細については、[入力パラメータ \(53 ページ\)](#) を参照してください。次に、[Review+Create] をクリックします。

ステップ 9 テンプレートの展開が成功すると、Azure 向けの ASA 仮想 自動スケールソリューションに必要なすべてのリソースが作成されます。[タイプ (Type)] 列には、Logic App、VMSS、ロードバランサ、パブリック IP アドレスなどの各リソースが示されます。

Function App への Azure Cluster 自動スケール機能の展開

ARM テンプレートの展開が完了すると、次の名前で関数アプリが作成されます。

<resourceNamePrefix>-function-app。関数アプリを作成したら、次の手順を実行して Azure クラスタ自動スケール機能を関数アプリに展開します。

手順

ステップ 1 ローカルコンピューターで、ターゲットフォルダに移動し、次のコマンドを実行して Azure クラスタ自動スケール機能を機能アプリに展開します。

```
az functionapp deployment source config-zip -g <Resource Group Name> -n <Function App Name> --src <cluster_functions.zip> --build-remote true
```

ステップ 2 以下に示すように、アップロードされた機能が機能アプリの[概要 (Overview)]セクションに表示されているかどうかを確認して、機能が正常に展開されていることを確認します。

Functions Metrics Properties Notifications (0)

{ } Set up local environment Refresh

Filter by name...

Name	Trigger	Status	Monitor
AsaScaleIn	HTTP	Enabled	Invocations and more
AsaScaleOut	HTTP	Enabled	Invocations and more
AutoScaleManager	HTTP	Enabled	Invocations and more
CheckASAvLicenseConfig	HTTP	Enabled	Invocations and more
CleanupASAvConfiguration	HTTP	Enabled	Invocations and more
ConfigureASAv	HTTP	Enabled	Invocations and more
DeleteUnconfiguredAsa	HTTP	Enabled	Invocations and more
GetAsaPublicIp	HTTP	Enabled	Invocations and more
StopNewConnections	HTTP	Enabled	Invocations and more
WaitForAsaToComeUp	HTTP	Enabled	Invocations and more

Azure Logic App の更新

Logic App は、Auto Scale 機能の Orchestrator として機能します。ARM テンプレートによってスケルトン Logic App が作成されます。このアプリケーションを手動で更新して、Auto Scale Orchestrator として機能するために必要な情報を提供する必要があります。

手順

ステップ 1 リポジトリから、LogicApp.txt ファイルをローカルシステムに取得し、次のように編集します。

重要

手順をすべて読んで理解してから続行してください。

手動の手順は、ARM テンプレートでは自動化されないため、Logic App のみ後で個別にアップグレードできます。

- 必須: すべての「SUBSCRIPTION_ID」を検索し、サブスクリプション ID 情報に置き換えます。
- 必須: すべての「RG_NAME」を検索し、リソースグループ名に置き換えます。
- 必須: すべての「FUNCTIONAPPNAME」を検索し、Function App 名に置き換えます。

次の例は、LogicApp.txt ファイルの行の一部を示しています。

```
"AutoScaleManager": {
  "inputs": {
    "function": {
      "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/AutoScaleManager"
```

```

    }
  },
  "Deploy_Changes_to_ASA": {
    "inputs": {
      "body": "@body('AutoScaleManager')",
      "function": {
        "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/FG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeployConfiguration"
      }
    },
    "DeviceDeRegister": {
      "inputs": {
        "body": "@body('AutoScaleManager')",
        "function": {
          "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/FG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeviceDeRegister"
        }
      },
      "runAfter": {
        "Delay_For_connection_Draining": [

```

- d) (任意) トリガー間隔を編集するか、デフォルト値 (5) のままにします。これは、Auto Scale 機能が定期的にトリガーされる時間間隔です。次の例は、LogicApp.txt ファイルの行の一部を示しています。

```

"triggers": {
  "Recurrence": {
    "conditions": [],
    "inputs": {},
    "recurrence": {
      "frequency": "Minute",
      "interval": 5
    }
  },

```

- e) (任意) ドレインする時間を編集するか、デフォルト値 (5) のままにします。これは、スケールイン操作中にデバイスを削除する前に、ASA Virtual から既存の接続をドレインする時間間隔です。次の例は、LogicApp.txt ファイルの行の一部を示しています。

```

"actions": {
  "Branch_based_on_Scale-In_or_Scale-Out_condition": {
    "actions": {
      "Delay_For_connection_Draining": {
        "inputs": {
          "interval": {
            "count": 5,
            "unit": "Minute"
          }
        }
      }
    }
  }
}

```

- f) (任意) クールダウン時間を編集するか、デフォルト値 (10) のままにします。これは、スケールアウト完了後に NO ACTION を実行する時間です。次の例は、LogicApp.txt ファイルの行の一部を示しています。

```

"actions": {
  "Branch_based_on_Scale-Out_or_Invalid_condition": {
    "actions": {
      "Cooldown_time": {
        "inputs": {
          "interval": {
            "count": 10,
            "unit": "Second"
          }
        }
      }
    }
  }
}

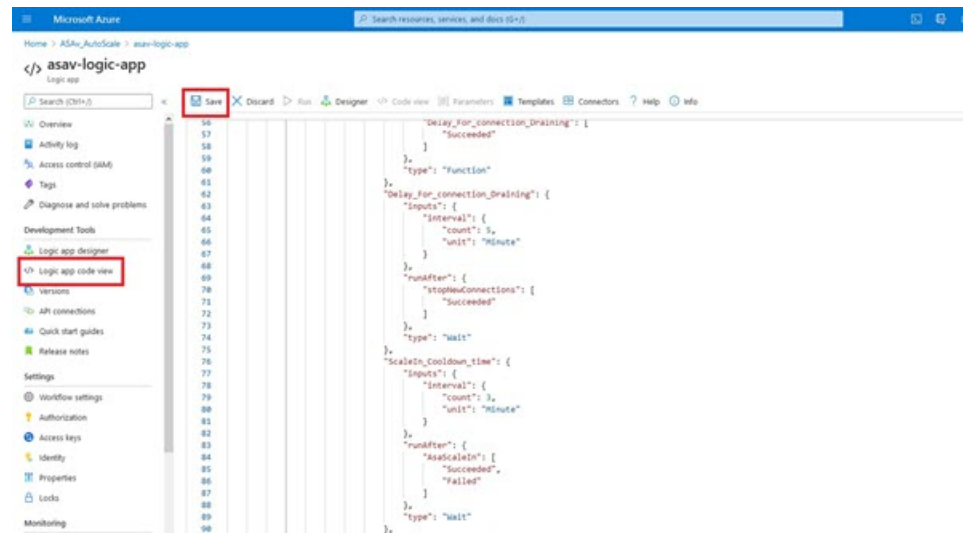
```

(注)

これらの手順は、Azure ポータルからも実行できます。詳細については、Azure のドキュメントを参照してください。

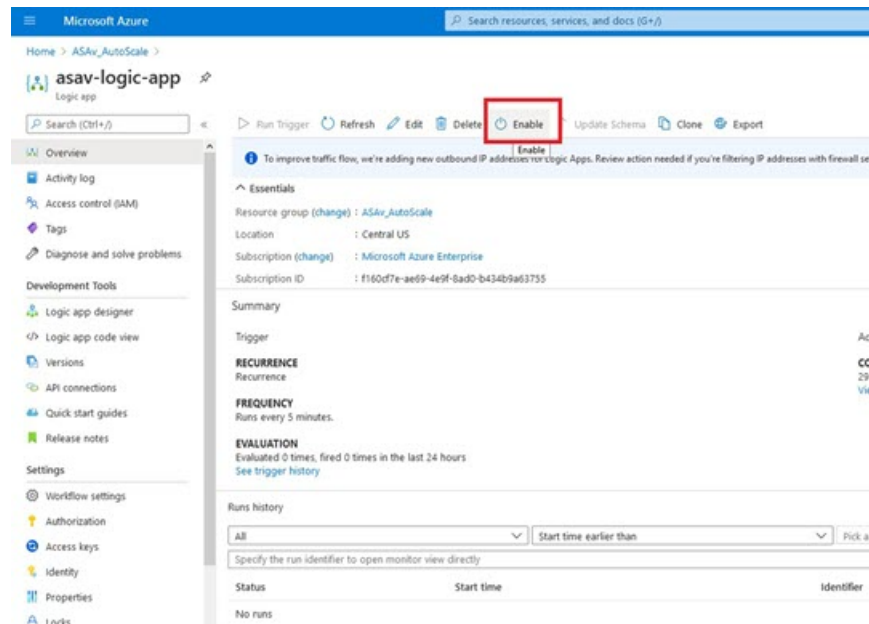
ステップ 2 [Logic Appコードビュー (Logic App code view)] に移動し、デフォルトの内容を削除して、編集した LogicApp.txt ファイルの内容を貼り付け、[保存 (Save)] をクリックします。

図 8: Logic App コードビュー



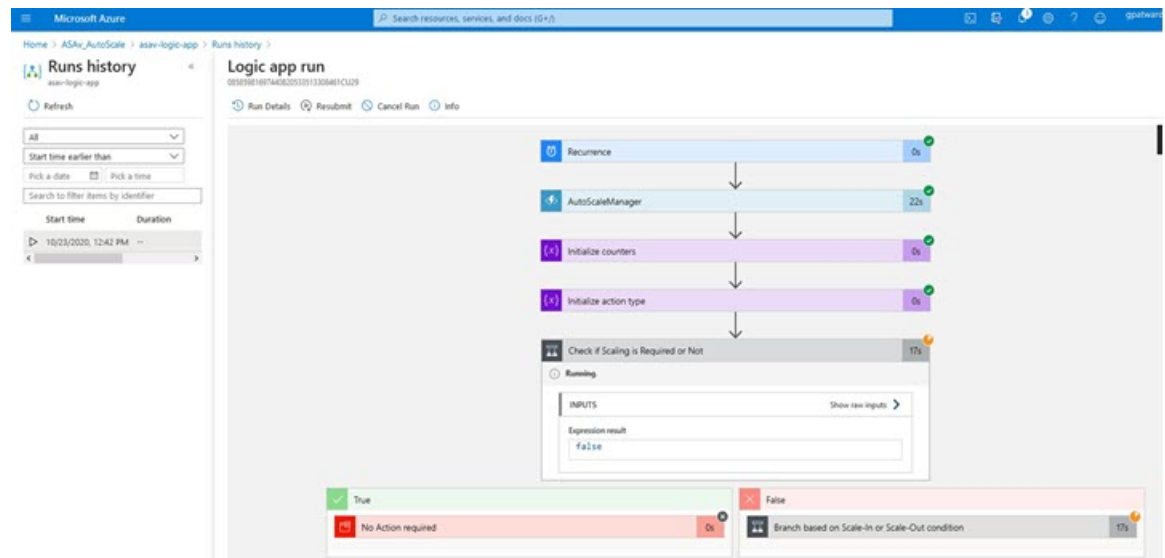
ステップ 3 Logic App を保存すると、[無効 (Disabled)] 状態になります。Auto Scale Manager を起動する場合は、[有効化 (Enable)] をクリックします。

図 9: Logic App の有効化



ステップ 4 有効にすると、タスクの実行が開始されます。[実行中 (Running)] ステータスをクリックしてアクティビティを表示します。

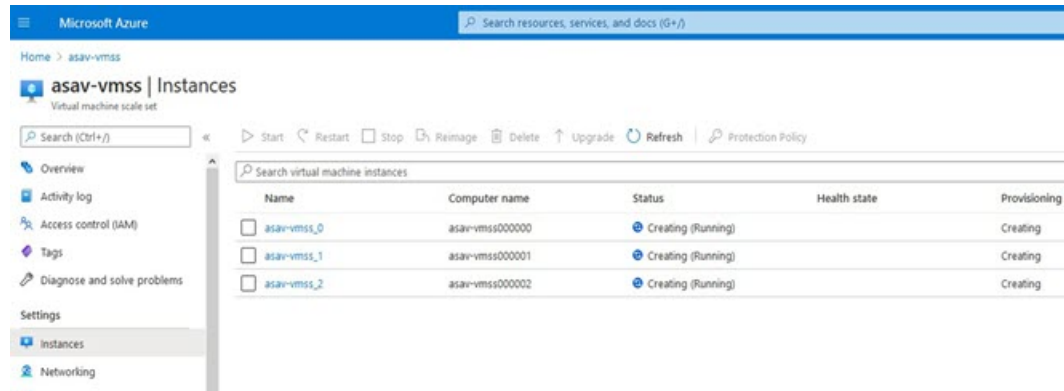
図 10: Logic App の実行ステータス



ステップ 5 Logic App が起動すると、導入関連のすべての手順が完了します。

ステップ 6 ASA Virtual インスタンスが作成されていることを VMSS で確認します。

図 11:稼働中のASA Virtual インスタンス



この例では、ARM テンプレートの展開で「minAsaCount」が「3」に設定され、「initDeploymentMode」が「BULK」に設定されているため、3 つの ASA Virtual インスタンスが起動されます。

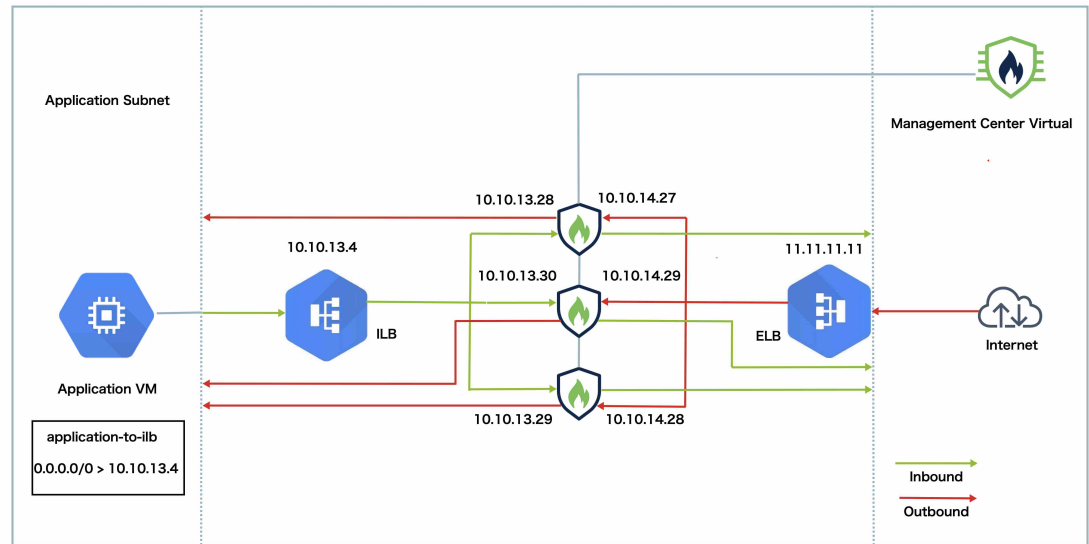
GCP の自動スケールソリューションを使用した Cisco ASA Virtual クラスタリング

リリース 9.24.1 以降、Google Cloud Platform (GCP) の Cisco ASA Virtual クラスタリングソリューションは、ダイナミック自動スケーリング機能で強化されています。Auto Scale ソリューションは、CPU 使用率のメトリックに基づいており、最適なリソース使用率の実現に役立ちます。それは、Terraform ベースのテンプレートを使用して展開されます。

スケーリングオプションは、要件に応じてダイナミッククラスタリングまたは固定ノードクラスタリングのいずれかから選択できます。スケーリングが開始される CPU しきい値を定義することもできます。

GCP クラスティング自動スケーリングソリューションのサンプルトポロジ

図 12: トポロジの例



このトポロジは、着信と発信の両方のトラフィックフローを示しています。

1. Cisco ASA Virtual クラスタは、内部ロードバランサと外部ロードバランサの間に置かれています。
2. インターネットからの着信トラフィックは、外部ロードバランサに送られ、そこから Cisco ASA Virtual クラスタにトラフィックが送信されます。
3. クラスタ内の Cisco ASA Virtual インスタンスはトラフィックを検査し、検査後、トラフィックをアプリケーション VM に転送します。
4. アプリケーション VM からの発信トラフィックは、内部ロードバランサに送信されます。ロードバランサはこのトラフィックを Cisco ASA Virtual クラスタに転送し、インターネットに送信します。

要件および前提条件

この項では、GCP での Cisco ASA Virtual クラスティング自動スケーリングソリューションの要件とサポートされている構成を示します。

モデルの要件

- サポートされている Cisco ASA Virtual モデル :
- 有効な Google Cloud Platform (GCP) アカウントが必要です。
- Google Cloud SDK をローカルにインストールするか、または GCP Cloud Shell にアクセスできる必要があります。

ライセンスニング

- 制御ノードのソフトウェア利用資格は、起動時の初期パフォーマンス階層構成に関係なく、クラスタ内のすべてのデータノードに適用されます。
- クラスタの形成後は、個々の Cisco ASA Virtual のパフォーマンス階層を変更できません。
- デフォルトでは、各ユニットには 100 kbps の評価ライセンスが含まれています。
- 登録解除後のライセンスの挙動：
 - 再起動まで：ユニットは既存のスループットを保持します。
 - 再起動後：スループットはデフォルトの 100 kbps に戻ります。
- BYOL (Bring Your Own License) ライセンスタイプのみがサポートされています。
- PAYG (ペイアズユーゴー) ライセンスタイプはサポートされていません。

サポートされている構成

- クラスタサイズ：1 ～ 16 ノード。
- トラフィックトポロジ：North-South および East-West。
- 展開モード：Terraform テンプレートをを使用して展開。

テンプレート

GitHub では、次のテンプレートを使用できます。

- `infrastructure/main.tf`：インフラストラクチャ展開用のテンプレート。
- `cluster_deployment/main.tf`：クラスタ展開用のテンプレート。

GCP での Cisco ASA Virtual クラスタリング自動スケールソリューションの展開

デプロイメントプロセスには次の 2 つの手順があります。

1. インフラストラクチャ デプロイメント (オプション)：必要なネットワーク インフラストラクチャを展開します。
2. クラスタデプロイメント：Cisco ASA Virtual クラスタを展開します。



- (注) Cisco ASA Virtual の展開に独自のインフラストラクチャを使用することを選択した場合、インフラストラクチャのデプロイメント手順をスキップできます。ただし、ソリューションを適切に機能させるために必要なすべてのリソースが提供されていることを確認することが重要です。

インフラストラクチャの展開のパラメータ

インフラストラクチャの展開には、次のパラメータが使用されます。

表 3: インフラストラクチャの展開のテンプレートパラメータのリスト (`infrastructure_params.tfvars`)

パラメータ	説明	例
project_id	GCP プロジェクト ID	test-project-12345
resource_name_prefix	文字列 (小文字のアルファベットのみのみ)	demoasav
region	展開する GCP リージョン	us-central1
mgmt_ip_cidr_range	管理サブネット CIDR	10.112.0.0/27
vpc_connector_ip_cidr_range	管理 VPC コネクタ CIDR/28 サブネットのみ	10.112.50.0/28
inside_ip_cidr_range	内部サブネット CIDR	10.112.1.0/27
outside_ip_cidr_range	外部サブネット CIDR	10.112.2.0/27
ccl_ip_cidr_range	CCL サブネット CIDR (/27 サブネット推奨)	10.112.100.0/27

GCP での Terraform を使用したインフラストラクチャの展開

GCP Infrastructure Manager は、スタックとしての Terraform の展開を促進します。このインターフェイスから、リソースの管理やクリーンアップを実行できます。

手順

ステップ 1 [GCP コンソール](#) に移動し、クレデンシアルを使用してログインします。

ステップ 2 Cloud シェルまたはローカル端末を開きます。

ローカル端末を使用している場合は、システムで GCP CLI が構成されているか確認します。

ステップ 3 新しいディレクトリを作成します。

例：

コマンド **mkdir infra** を使用して、Cloud シェルまたはローカル端末に **infra** という名前のディレクトリを作成します。

ステップ 4 新しく作成したディレクトリに移動します。

例：

現在のディレクトリを変更するには、**cd infra** コマンドを使用します。

ステップ 5 Cisco GitHub リポジトリから **infrastructure_params.tfvars** ファイルをコピーまたはダウンロードし、新しい **infra** フォルダに保存します。

ステップ 6 **infrastructure_params.tfvars** ファイルを開き、必要に応じてすべてのプレースホルダ値を更新します。[インフラストラクチャの展開のパラメータ \(73 ページ\)](#) を参照してください。

ステップ 7 プレースホルダ値を更新した後、以下のコマンドを使用して、展開を開始します。

```
gcloud infra-manager deployments \
  apply
  "projects/<project_id>/locations/<region>/deployments
  /<deployment_name>" \
  --location="<region>" \
  --git-source-repo="<repo name>" \
  --git-source-directory="infrastructure" \
  --git-source-ref="<branch name>" \
  --serviceaccount="
  projects/<project_id>/serviceAccounts/<service_account_name>" \
  --artifacts-gcs-bucket="gs://<bucket_name>/artifacts"
  \
  --inputs-file="/path/to/your/infra_params.tfvars"
```

表 4: パラメータの説明

パラメータ	説明
deployment_name	GCP Infrastructure Manager に表示される Terraform 展開スタックの名前。
location	デプロイメントが適用される場所またはリージョン。
git-source-repo	ソースコードを含む Git リポジトリの名前 (Cisco GitHub リンク)。
git-source-directory	インフラストラクチャコードが配置されている Git リポジトリ内のディレクトリ。
git-source-ref	デプロイメントに使用する Git リポジトリ内のブランチ名。
service-account	サービスアカウントの名前。
artifacts-gcs-bucket	展開に関連するアーティファクトを保存する Google Cloud Storage バケット (オプション)。

パラメータ	説明
inputs-file	インフラストラクチャ構成の入力パラメータへのファイルパス。

(注)

テンプレートを使用してインフラストラクチャを展開しない場合は、ソリューションを適切に展開および機能させるために、次のリソースを提供する必要があります。

- 管理 VPC
 - 管理サブネット
 - VPC コネクタの管理サブネット
 - 管理ファイアウォールルールの名前
- 内部 VPC
 - 内部サブネット
 - 内部ファイアウォールルール名
- 外部 VPC
 - 外部サブネット
 - 外部ファイアウォールルールの名前。
 - Nat GW
- CCL VPC
 - CCL サブネット
 - CCL ファイアウォールルール名
- 診断 VPC (オプション)
 - 診断ファイアウォールルールの名前
- 管理サブネットを持つ VPC コネクタ

クラスタの展開パラメータ

次の入力パラメータを使用して、クラスタを展開します。

表 5: クラスタデプロイメントのテンプレートパラメータのリスト

パラメータ	説明	例
プロジェクトの設定		
project_id	GCP プロジェクト ID	test-project-12345
resource_name_prefix	リソースの命名のプレフィックス	democluster
type_of_deployment	デプロイメントのタイプ (north_south または east_west)	north_south
region	GCP リージョン (GCP region)	us-east1
zone1	デプロイメントゾーン	a
zone2	デプロイメントゾーン	b
zone3	デプロイメントゾーン	c
VM の設定		
machine_type	VM インスタンスタイプ	n1-standard-4
source_image_url	ソースイメージの URL	projects/my-project/global/images/my-image
assign_public_ip_to_mgmt	パブリック IP を管理インターフェイスに割り当てるかどうか (true/false)	false
public_key	認証用の SSH 公開キー	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQ...
service_account_mail_id	サービスアカウントの電子メール	service-account-mail@example-project.iam.gserviceaccount.com
asav_password_secret_name	ASAv パスワードのシークレット名	GCP シークレットマネージャの asav_password_secret
asav_en_password_secret_name	ASAv 暗号化パスワードのシークレット名	GCP シークレットマネージャの asav_en_password_secret
vPC の設定		
mgmt_vpc_name	管理 VPC 名	<resource_name_prefix>-mgmt-vpc
mgmt_subnet_name	管理サブネット名	<resource_name_prefix>-mgmt-subnet
inside_vpc_name	内部 VPC 名	<resource_name_prefix>-inside-vpc

パラメータ	説明	例
inside_subnet_name	内部サブネット名	<resource_name_prefix>-inside-subnet
outside_vpc_name	外部 VPC 名	<resource_name_prefix>-outside-vpc
outside_subnet_name	外部サブネット名	<resource_name_prefix>-outside-subnet
ccl_vpc_name	クラスタリング VPC の名前	<resource_name_prefix>-ccl-vpc
ccl_subnet_name	クラスタリングサブネットの名前	<resource_name_prefix>-ccl-subnet
vpc_connector_name	VPC コネクタの名前	my-vpc-connector
ファイアウォールルール		
ccl_firewall_rule_name	クラスタリング ファイアウォール ルールの名前	<resource_name_prefix>-mgmt-firewall-rule
mgmt_firewall_rule_name	管理ファイアウォール ルールの名前	<resource_name_prefix>-mgmt-firewall-rule
inside_firewall_rule_name	内部ファイアウォール ルールの名前	<resource_name_prefix>-in-firewall-rule
outside_firewall_rule_name	外部ファイアウォール ルールの名前	<resource_name_prefix>-out-firewall-rule
クラスタ		
cluster_grp_name	クラスタグループの名前	cluster-group-name
ccl_subnet_firstIp	クラスタリングサブネットの最初の IP アドレス	10.0.0.1
ccl_subnet_lastIp	クラスタリングサブネットの最後の IP アドレス	10.0.0.254
自動スケーリングの設定		
auto_scaling	自動スケーリングの有効化または無効化	true/false
cpu_utilization_target	自動スケーリングのターゲット CPU 使用率 [0.0-1.0]	0.2

パラメータ	説明	例
min_asa_count	自動スケーリンググループ内の Cisco ASA Virtual インスタンスの最小数	1
max_asa_count	自動スケーリンググループ内の Cisco ASA Virtual インスタンスの最大数	2
内部ロードバランサ構成		
ilb_frontend_protocol	フロントエンドプロトコル (TCP/UDP)	TCP
ilb_backend_protocol	バックエンドプロトコル	TCP
ilb_health_check_port	ILB ヘルスチェック バランサ ポート	80
ilb_timeout_sec	ロードバランサのタイムアウト (秒)	5
ilb_check_interval_sec	ILB のヘルスチェックの間隔 (秒)	10
ilb_unhealthy_threshold	内部ロードバランサの異常しきい値	3
外部ロードバランサ構成		
elb_frontend_protocol	外部ロードバランサのフロントエンドプロトコル名 (TCP/UDP)	TCP
elb_backend_protocol	外部ロードバランサのバックエンドプロトコル名 (TCP/UDP/UNSPECIFIED)	TCP
elb_front_end_ports	ELB フロントエンド (リスナー) ポートのリスト	all または [80, 443]
elb_health_check_port	ELB ヘルスチェック ポート	80
elb_timeout_sec	ロードバランサのタイムアウト (秒)	5

パラメータ	説明	例
elb_unhealthy_threshold	外部ロードバランサの異常しきい値	2
elb_check_interval_sec	ELB のヘルスチェックの間隔 (秒)	10
ライセンシング		
license_throughput	ライセンスのスループット	10G
license_id_token	ライセンス ID トークン	my-license-id-token

GCP での Terraform を使用したクラスタの展開

GCP Infrastructure Manager は、スタックとしての Terraform の展開を促進します。このインターフェイスから、リソースの管理やクリーンアップを実行できます。

手順

ステップ 1 [GCP コンソール](#) に移動し、クレデンシャルを使用してログインします。

ステップ 2 Cloud シェルまたはローカル端末を開きます。

ローカル端末を使用している場合は、システムで GCP CLI が構成されているか確認します。

ステップ 3 新しいディレクトリを作成します。

例：

コマンド **mkdir cluster** を使用して、Cloud シェルまたはローカル端末で **cluster** という名前のディレクトリを作成します。

ステップ 4 新しく作成したディレクトリに移動します。

例：

cd cluster コマンドを使用して新しいディレクトリに移動します。

ステップ 5 Cisco GitHub リポジトリから **cluster_params.tfvars** ファイルをコピーまたはダウンロードし、新しい **cluster** フォルダに保存します。

ステップ 6 **cluster_params.tfvars** ファイルを開き、必要に応じてすべてのプレースホルダ値を更新します。[クラスタの展開パラメータ \(75 ページ\)](#) を参照してください。

ステップ 7 プレースホルダ値を更新した後、以下のコマンドを使用して、展開を開始します。

```
gcloud infra-manager deployments \
  apply "projects/<project_id>/locations/<region>/deployments/<deployment_name>" \
  --location="<region>" \
  --git-source-repo="<repo name>" \
  --git-source-directory="cluster_deployment" \
```

```
--git-source-ref="<branch name>" \
--service-account="projects/<project_id>/serviceAccounts/<service_account_name>" \
--artifacts-gcs-bucket="gs://<bucket_name>/artifacts" \
--inputs-file="/path/to/cluster_params.tfvars"
```

表 6: パラメータの説明

パラメータ	説明
deployment_name	GCP Infrastructure Manager に表示される Terraform 展開スタックの名前。
location	デプロイメントが適用される場所またはリージョン。
git-source-repo	ソースコードを含む Git リポジトリの名前 (Cisco GitHub リンク)。
git-source-directory	インフラストラクチャコードが配置されている Git リポジトリ内のディレクトリ。
git-source-ref	デプロイメントに使用する Git リポジトリ内のブランチ名。
service-account	サービスアカウントの名前。
artifacts-gcs-bucket	デプロイメントに関連するアーティファクトを保存する Google Cloud Storage バケット (Terraform 状態ファイルなど)。
inputs-file	クラスタデプロイメントの入力パラメータへのファイルパス。

クラスタ展開後の展開の構成

Google Cloud Shell の内部ロードバランサを介して必要なトラフィックを転送するためのルートを VPC 内に作成するには、次の手順に従います。

```
gcloud compute routes create <ilb-route-name>
--network=<inside-vpc-name> --priority=1000 --destination-range=0.0.0.0/0
--next-hop-ilb=<ilb-forwarding-rule-name> --next-hop-ilb-region=<region>
```



(注) ルートはGCP UIからも作成できます。GCP UIからルートを作成するには、次の手順を踏みます。

1. **GCP UI** で、[ルート (Routes)] > [ルート管理 (Route Management)] > [ルートの作成 (Create route)] に移動します。
2. [ルートの作成 (Create a route)] ダイアログボックスに入力します。
 1. [名前 (Name)] : ルート名を指定します。
 2. [ネットワーク (Network)] : **<inside-vpc-name>** を選択します。
 3. [ルートタイプ (Route type)] : [スタティックルート (Static Route)] を選択します。
 4. [接続先IPv4範囲 (Destination IPv4 range)] : **0.0.0.0/0** と入力します。
 5. [優先順位 (Priority)] : **1000** に設定します。
 6. [次のホップ (Next Hop)] : [内部パススルーネットワークロードバランサの転送ルールを指定する (Specify a forwarding rule of internal passthrough Network Load Balancer)] を選択します。
 7. [転送ルール名 (Forwarding rule name)] : **<forwarding-rule-for-ilb>** を選択します。
3. [作成 (Create)] をクリックして、ルートを作成します。

GCP でのクラスタの手動展開

GCP 向け Day 0 構成の作成 :

クラスタを手動で展開するには、day0 構成を準備します。

```
{
  "AdminPassword": "password",
  "Hostname": "hostname",
  "FirewallMode": "Routed",
  "ManageLocally": "No",
  "Diagnostic": "OFF",          //Optional user input from version 7.4.1
  "Cluster": {
    "CclSubnetRange": "ip_address_start ip_address_end",
    "ClusterGroupName": "cluster_name"
  }
}
```

次に例を示します。

```
{
  "AdminPassword": "C15co@!23",
  "Hostname": "cisco-asa",
  "FirewallMode": "routed",
  "Diagnostic": "OFF",
  "ManageLocally": "No",
  "Cluster": {
    "CclSubnetRange": "10.10.65.2 10.10.65.29",
```

```
"ClusterGroupName": "gcp-cls-asa"
}
```

自動スケールによる GCP クラスターリングのトラブルシューティング

問題：クラスタが形成されていない。

回避策：

- NVE 専用のクラスターインターフェイスの IP アドレスをチェックします。他のノードの NVE 専用のクラスターインターフェイスに ping を実行できることを確認してください。
- NVE 専用のクラスターインターフェイスの IP アドレスが、オブジェクトグループの一部であることを確認します。
- NVE（ネットワーク仮想化エッジ）インターフェイスが対応するオブジェクトグループで設定されていることを確認します。
- クラスターグループのクラスターインターフェイスに正しい VNI（仮想ネットワーク識別子）インターフェイスがあることを確認します。この VNI インターフェイスには、対応するオブジェクトグループを持つ NVE があるはずです。
- 各ノードには、独自のクラスターインターフェイスの IP アドレスがあります。ノードで相互に ping を実行して接続を確認できることを確認します。

クラスターリング動作のカスタマイズ

Day0 設定の一環として、またはクラスターの展開後に、クラスター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 クラスターの基本パラメータの設定

制御ノード上のクラスター設定をカスタマイズできます。

手順

ステップ 1 クラスター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)

ステップ4 フローオーナーからディレクタおよびバックアップオーナーへのフロー状態更新メッセージ (clu_heartbeat および clu_update メッセージ) のキープアライブ間隔を設定します。

clu-keepalive-interval seconds

- **seconds** : 15 ~ 55。デフォルトは 15 です。

クラスタ制御リンクのトラフィック量を減らすために、デフォルトよりも長い間隔を設定することもできます。

ステップ5 同時参加を有効にします。

concurrent-join

この機能は、デフォルトでイネーブルにされています。有効にすると、クラスタノードは順番ではなく同時に参加します。NAT および VPN 分散モードが有効になっている場合、同時結合は使用できません。互換性のない構成を表示するには、**show cluster info concurrent-join incompatible-config** を入力します。

例 :

```
ciscoasa(cfg-cluster)# concurrent-join
ciscoasa(cfg-cluster)# show cluster info concurrent-join incompatible-config

INFO: Clustering is not compatible with following commands. User must
manually remove them to activate the cluster concurrent-join feature.
nat (gre_outside,gre_inside) source static any interface destination static interface
any unidirectional

ciscoasa(cfg-cluster)# no nat (gre_outside,gre_inside) source static any interface
destination static interface any unidirectional
ciscoasa(cfg-cluster)# show cluster info concurrent-join incompatible-config

INFO: No concurrent-join incompatible config found.
```

ヘルスモ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モ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モニタリングをディセーブルにすることができます。ヘルスモニタリングは 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 クラスタ制御リンクの正常性チェックを一時停止するように CPU 使用率のしきい値を設定します。

cpu-healthcheck-threshold percent

percent : 70 ～ 100。デフォルトは 90 です。

クラスタノードの CPU 使用率が高い場合、正常性チェックは一時停止され、ノードは異常とはマークされません。

ステップ 4 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 モ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。ヘルスモニタリングは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が無効（**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 5 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 6 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 7 (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]

- **frequency 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。デフォルトは 20 秒です。
- **intervals intervals** : 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。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
```

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 unit*node_name*

例：

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

メイン クラスター 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。

クラスター全体でのコマンドの実行

コマンドをクラスター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、クラスター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。

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.: P30000000025
    CCL IP   : 10.0.0.3
    CCL MAC   : 000b.fcfc8.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.fcf8.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.fcf8.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、データパ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グgingsの設定

クラスタリングのログgings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_KEEPA_LIVE from 80-1 at
CONTROL_NODE
```

たとえば、同じローカルユニット名を持つ2つのノードが制御ノードとして機能していることを示す次のメッセージが表示された場合は、2つのノードのローカルユニット名が同じである（設定を確認してください）か、1つのノードが自身のブロードキャストメッセージを受信している（ネットワークを確認してください）可能性があります。

```
ciscoasa# show cluster info trace
```

```

May 23 07:27:23.113 [CRIT]Received datapath event 'multi control_nodes' with parameter
1.
May 23 07:27:23.113 [CRIT]Found both unit-9-1 and unit-9-1 as control_node units.
Control_node role retained by unit-9-1, unit-9-1 will leave then join as a Data_node

May 23 07:27:23.113 [DEBUG]Send event (DISABLE, RESTART | INTERNAL-EVENT, 5000 msec,
Detected another Control_node, leave and re-join as Data_node) to FSM. Current state
CONTROL_NODE
May 23 07:27:23.113 [INFO]State machine changed from state CONTROL_NODE to DISABLED

```

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る場合があります。

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す。
- VPN ロード バランシング
- Azure でのフェールオーバー
- 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。アカウンティングは非集中型です。
- 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、出力に対してポリシーを設定する場合は、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。

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、アカウンティングの 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。

アカウンティングは、クラスタ内の分散型機能として実装されています。アカウンティング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ティング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ティング開始と停止のメッセージを AAA サーバーに送信します。

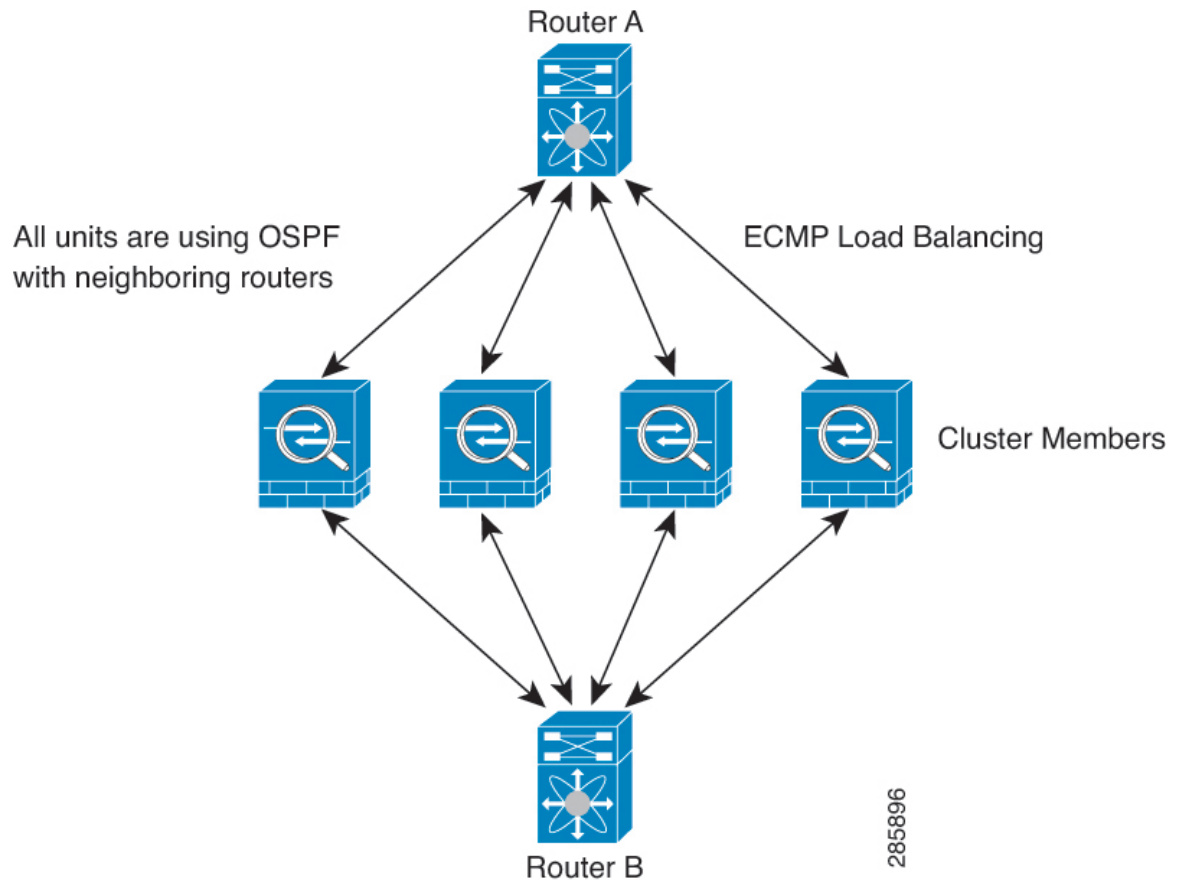
接続設定とクラスタリング

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。ルート学習は、各ノードが個別に行います。

図 13: 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 つのノードを通過します。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。各ノードは、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ノードインターフェイスをまとめます。[トラフィックゾーンの設定](#)を参照してください。

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 インスペク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 エラーインスペクションが有効かどうかによって異なります。ICMP インスペク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 インスペク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

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。

- 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、クラスターリングではサポートされません。
- 拡張 PAT なし：拡張 PAT はクラスター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 がテーブル内にない場合、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：クラスター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。クラスターリングの場合は、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。たとえば、TCP/443 の HTTPS TLS と比較してずっと優れたパフォーマンスを発揮する代替手段として、UDP/443 を用いる Quick プロトコルの使用が増加している場合、UDP/443 に対し per-session PAT を有効にする必要があります。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、ノードがクラスター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 インスペク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とき、STUN インスペク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 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。

クラスタ内のハイアベイラビリティ

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ーし、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。

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。

ヘルスモ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

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 name** と入力してから **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ルにします。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す。

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 7: 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	対応	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	対応	—
MAC アドレス テーブル	対応	—
ユーザ アイデンティティ	対応	AAA ルール (uauth) が含まれます。
IPv6 ネイバーデータベース	対応	—
ダイナミック ルーティング	対応	—
SNMP エンジン ID	非対応	—
Firepower 4100/9300 の分散型 VPN (サイト間)	対応	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。

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。接続のロールにより、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。

接続のロール

接続ごとに定義された次のロ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、フォワーダーになることもできます。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、フォワーダー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する 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。

ポートアドレス変換接続

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。

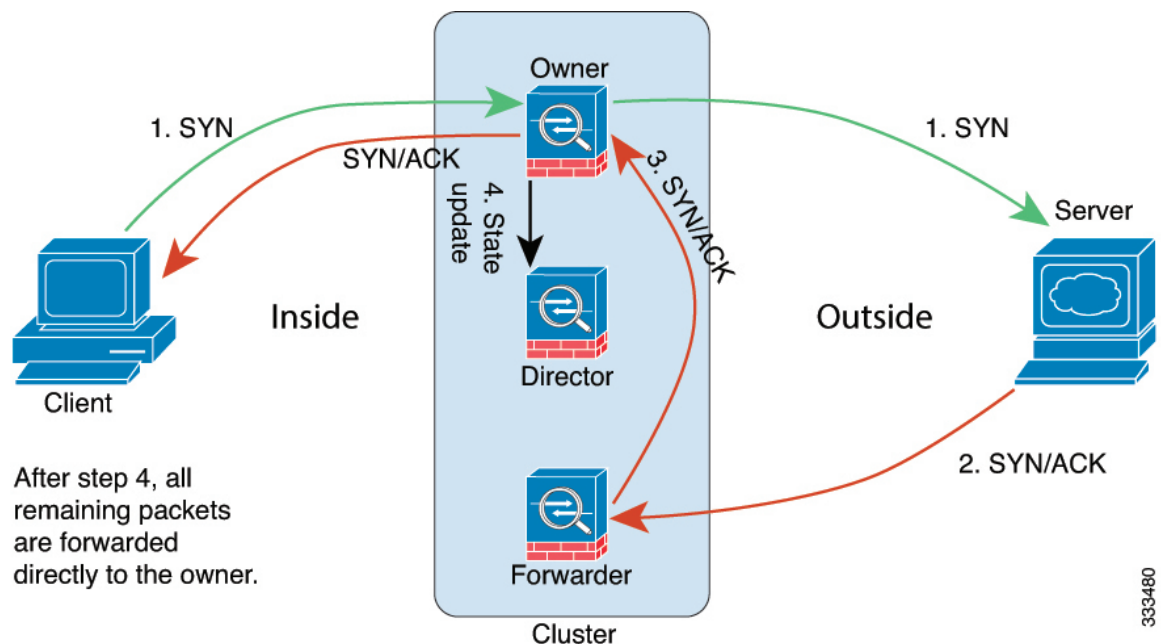
TCP および UDP の `per-session PAT` デフォルトを変更できるので、これらのプロトコルの接続は、その設定に応じて `per-session` または `multi-session` で処理されます。ICMP の場合は、デフォルトの `multi-session PAT` から変更することはできません。 `per-session PAT` の詳細については、『ファイアウォールの構成ガイド』を参照してください。

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。

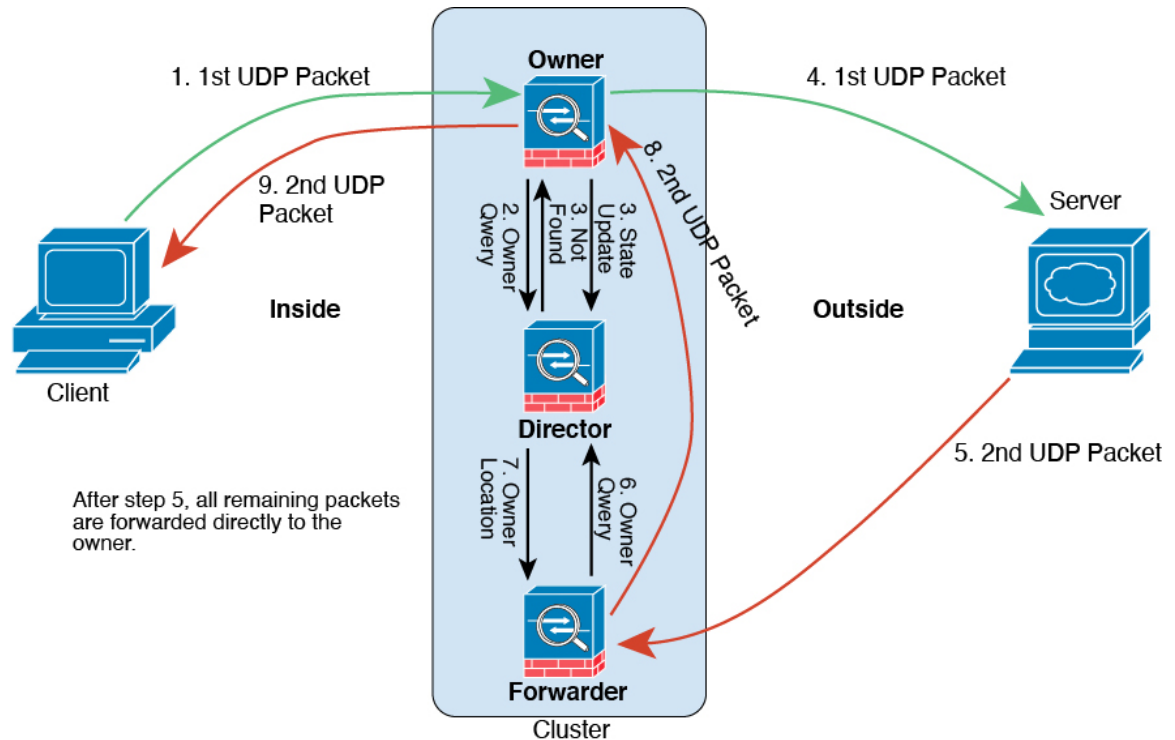


1. SYN パケットがクライアントから発信され、ASA の 1 つ（ロードバランシング方法に基づく）に配信されます。これがオーナーとなります。オーナーはフローを作成し、オーナー情報をエンコードして SYN Cookie を生成し、パケットをサーバに転送します。
2. SYN-ACK パケットがサーバから発信され、別の ASA（ロードバランシング方法に基づく）に配信されます。この ASA はフォワーダーです。
3. フォワーダーはこの接続を所有してはいないので、オーナー情報を SYN Cookie からデコードし、オーナーへの転送フローを作成し、SYN-ACK をオーナーに転送します。
4. 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は、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 14: 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. 最初のパケット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。

パブリッククラウドにおける ASA Virtual クラスターリングの履歴

機能	バージョン	詳細
データノードはクラスターに同時に参加できます	9.24(1)	以前は、制御ノードで一度に1つのデータノードのみがクラスターに参加できました。設定の同期に時間がかかる場合、データノードの結合に時間がかかることがあります。同時結合はデフォルトで有効になっています。NAT および VPN 分散モードが有効になっている場合、同時結合は使用できません。 次のコマンドが追加/変更されました。concurrent-join、show cluster info concurrent-join incompatible-config
クラスターノード結合での MTU ping テストでは、MTU を小さくすることでより多くの情報が提供されます。	9.24(1)	クラスターに参加したノードは、クラスター制御リンク MTU と一致するパケットサイズで制御ノードに ping を送信することで MTU の互換性をチェックします。ping が失敗した場合は、MTU を 2 で割った値を試し、MTU ping が成功するまで 2 で割った値を返しません。通知が生成されるため、MTU を適切な値に修正して再試行できます。スイッチ MTU サイズを推奨値に増やすことを推奨しますが、スイッチ設定を変更できない場合は、クラスター制御リンクの有効な値を使用してクラスターを形成できます。 次のコマンドが追加/変更されました。show cluster history
CPU 使用率が高いクラスター制御リンクの正常性チェックの改善	9.24(1)	クラスターノードの CPU 使用率が高い場合、正常性チェックは一時停止され、ノードは異常とはマークされません。正常性チェックを一時停止する CPU 使用率のしきい値を設定できます。 次のコマンドが追加/変更されました。cpu-healthcheck-threshold
ノード参加時の MTU ping テスト	9.22(1)	クラスターに参加したノードは、クラスター制御リンク MTU と一致するパケットサイズで制御ノードに ping を送信することで MTU の互換性をチェックします。ping が失敗すると、通知が生成されるため、接続スイッチの MTU 不一致を修正して再試行することができます。
Azure 上の Cisco ASA Virtual : ゲートウェイロードバランシングによるクラスターリング	9.20(2)	Azure Resource Manager (ARM) テンプレートを使用した Azure での Cisco ASA Virtual のクラスターリングの展開がサポートされるようになり、ネットワークトラフィックのロードバランシングにゲートウェイロードバランサ (GWLB) を使用するように ASA Virtual クラスターが設定されています。

機能	バージョン	詳細
AWS における GWLB を使用した ASA Virtual クラスタリングのターゲットフェールオーバーの設定	9.20(2)	AWS でターゲットフェールオーバー機能を使用すると、計画メンテナンス中またはターゲットノードの障害時にノードの登録が解除された場合に、GWLB がネットワークパケットを正常なターゲットノードにリダイレクトできるようになります。この機能ではクラスタのステートフルフェールオーバーを利用しています。
フローステータスの設定可能なクラスタキープアライブ間隔	9.20(2)	フローオーナーは、キープアライブ (clu_keepalive メッセージ) と更新 (clu_update メッセージ) をディレクタおよびバックアップオーナーに送信して、フローの状態を更新します。キープアライブ間隔を設定できるようになりました。デフォルトは 15 秒で、15 ～ 55 秒の範囲で間隔を設定できます。クラスタ制御リンクのトラフィック量を減らすために長い間隔を設定できます。 新規/変更されたコマンド: clu-keepalive-interval
Cisco ASA Virtual Amazon Web Services (AWS) クラスタリング	9.19(1)	Cisco ASA Virtual は AWS で最大 16 ノードの個別インターフェイスのクラスタリングをサポートします。AWS ゲートウェイロードバランサの有無にかかわらず、クラスタリングを使用できます。
Amazon Web Services (AWS) クラスタリングでの ASA virtual IMDSv2 サポート	9.22	Cisco ASA Virtual は、AWS での IMDSv2 をサポートしています。スタックを更新することで、IMDSv2 必須モードを有効にすることができます。

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。

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。

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。

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。