パブリッククラウドでの Threat Defense Virtual のクラスタリング

クラスタリングを利用すると、複数の Firewall Threat Defense Virtual 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。以下のパブリック クラウド プラットフォームを使用して、パブリッククラウドに Firewall Threat Defense Virtual クラスタを展開できます。

- Amazon Web Services（AWS）
- Microsoft Azure
- Google Cloud Platform（GCP）

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。「サポートされていない機能とクラスタリング （84 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 Threat Defense 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、Firewall Threat Defense Virtual はクラスタ制御リンクを介してブロードキャスト/マルチキャストメッセージを送信できます。
- ロードバランサ：パブリッククラウドに応じて、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。Firewall Threat Defense Virtual 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。
  - Azure ゲートウェイロードバランサ

    Azure サービスチェーンでは、Firewall Threat Defense 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。Firewall Threat Defense Virtual は、ペアリングされたプロキシの VXLAN セグメントを利用して、単一の NIC に外部インターフェイスと内部インターフェイスを定義します。
  - 内部および外部のネイティブ GCP ロードバランサ
  - 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、Firewall Threat Defense 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 Firewall Threat Defense 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 Firewall Threat Defense 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。

個々のインターフェイス

クラスターフェイスを個々のインターフェイスとして設定できます。

個別インターフェイスは通常のルーテッドインターフェイスであり、それぞれが専用のローカル IP アドレスを持ちます。インターフェイスの IP アドレスは、DHCP を介して自動的に設定されます。静的 IP 設定はサポートされていません。

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、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。 最初にクラスタを作成するときに、制御ノードにするノードを指定します。これは、クラスタに追加された最初のノードであるため、制御ノードになります。

クラスタ内のすべてのノードは、同一の設定を共有します。最初に制御ノードとして指定したノードは、データノードがクラスタに参加するときにその設定を上書きします。そのため、クラスタを形成する前に制御ノードで初期設定を実行するだけで済みます。

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。

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

VNI インターフェイス

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

ピア VTEP

単一の VTEP ピアを許可するデータインターフェイス用の通常の VXLAN とは異なり、Firewall Threat Defense 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。

- ステート複製。
- 接続所有権クエリおよびデータ パケット転送。

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。

管理ネットワーク

管理インターフェイスを使用して各ノードを管理する必要があります。クラスタリングでは、データインターフェイスからの管理はサポートされていません。

Threat Defense Virtual クラスタリングのライセンス

各 Firewall Threat Defense Virtual クラスタノードには、同じパフォーマンス階層ライセンスが必要です。すべてのメンバーに同じ数の 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。

個別のノードではなく、クラスタ全体に機能ライセンスを割り当てます。ただし、クラスタの各ノードは機能ごとに個別のライセンスを使用します。クラスタリング機能自体にライセンスは必要ありません。

制御ノードを Firewall Management Center に追加する際に、そのクラスタに使用する機能ライセンスを指定できます。クラスタのライセンスは、**[デバイス（Devices）]**>**[デバイス管理（Device Management）]** > **[クラスタ（Cluster）]** > **[ライセンス（License）]** 領域で変更できます。

（注） Firewall Management Center にライセンスを取得する（および評価モードで実行する）前にクラスタを追加した場合、Firewall Management Center にライセンスを取得する際にポリシーの変更をクラスタに展開するとトラフィックの中断が発生することがあります。ライセンスモードを変更したことによって、すべてのデータユニットがクラスタをいったん離れてから再参加することになります。

Threat Defense Virtual クラスタリングの要件および前提条件

モデルの要件

- FTDv5、FTDv10、FTDv20、FTDv30、FTDv50、FTDv100

（注） FTDv5 および FTDv10 は、Amazon Web Services（AWS）ゲートウェイロードバランサ（GWLB）をサポートしていません。

- 以下のパブリッククラウドサービス：
  - Amazon Web Services（AWS）
  - Microsoft Azure
  - Google Cloud Platform（GCP）
- 最大 16 ノード

Cisco Secure Firewall Threat Defense Virtual スタートアップガイド の Firewall Threat Defense Virtual の一般要件も参照してください。

ユーザの役割

- 管理者
- アクセス管理者
- ネットワーク管理者

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ユニット：

- 同じパフォーマンス層内に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。
- Firewall Management Center へのアクセスは管理インターフェイスから行うこと。データインターフェイスの管理はサポートされていません。
- 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。
- クラスタ内のすべてのユニットは、同じ可用性ゾーンに展開する必要があります。
- すべてのユニットのクラスタ制御リンクインターフェイスは、同じサブネット内にある必要があります。

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。クラスタ制御リンクの MTU は、送信元インターフェイスの MTU の + 80 バイトになるように設定する必要があります。

次の表は、クラスタ制御リンク MTU のデフォルト値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*: デフォルト *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
GCP	1554	1400

Threat Defense 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、Firewall Threat Defense 上でのインターフェイスまたはスイッチの有効化または無効化、VSS または vPC または VNet 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のインターフェイス 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。
- ノードでクラスタリングを無効にせずにノードの電源を切らないでください。
- 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 スケーリングはサポートされていません。
- は、AWS にクラスタを展開する場合にステートフル ターゲット フェールオーバーはサポートされません。
- 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）/インスタンス グループ（GCP）/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）/インスタンス グループ（GCP）/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カスタマイズした 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。
- クラウド プラットフォームに Threat Defense 仮想クラスタを展開した後の断続的な ping の失敗など、CCL が不安定になる問題が発生した場合は、CCL の不安定性の原因に対処することをお勧めします。また、CCL が不安定になる問題をある程度軽減するための一時的

な回避策として、保留時間を増やすこともできます。保留時間の変更方法の詳細については、「クラスタの正常性モニタリング設定の編集」を参照してください。

- Management Center Virtual のセキュリティ ファイアウォール ルールまたはセキュリティグループを設定する場合は、Firewall Threat Defense Virtual のプライベート IP アドレスとパブリック IP アドレスの両方を送信元 IP アドレス範囲に含める必要があります。また、Firewall Threat Defense Virtual のセキュリティ ファイアウォール ルールまたはセキュリティグループで、Firewall Management Center Virtual のプライベート IP アドレスとパブリック IP アドレスを指定してください。これは、クラスタリングの展開中にノードを適切に登録するために重要です。

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

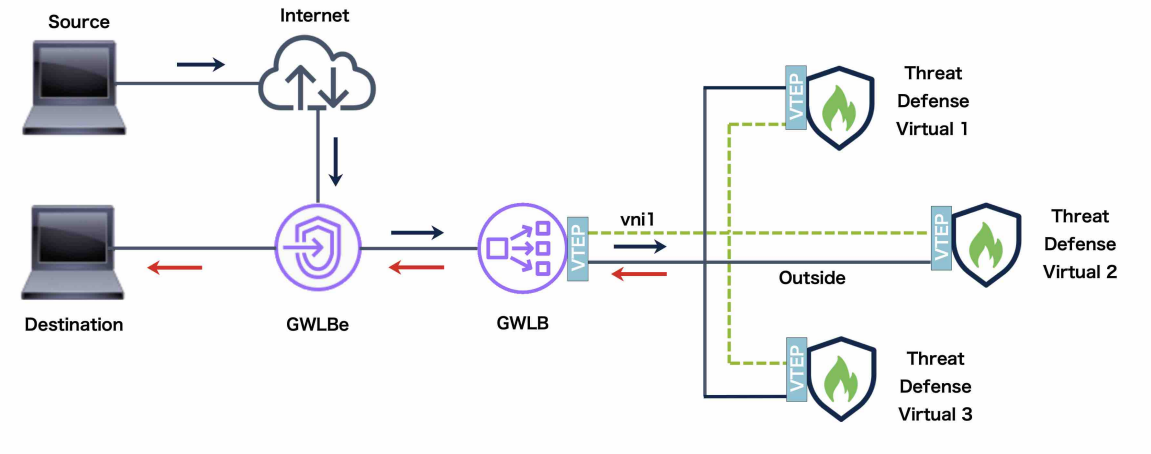
（注） この使用例は、現在サポートされている Geneve インターフェイスの唯一の使用例です。

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。Threat Defense 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 Threat Defense Virtual 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、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。

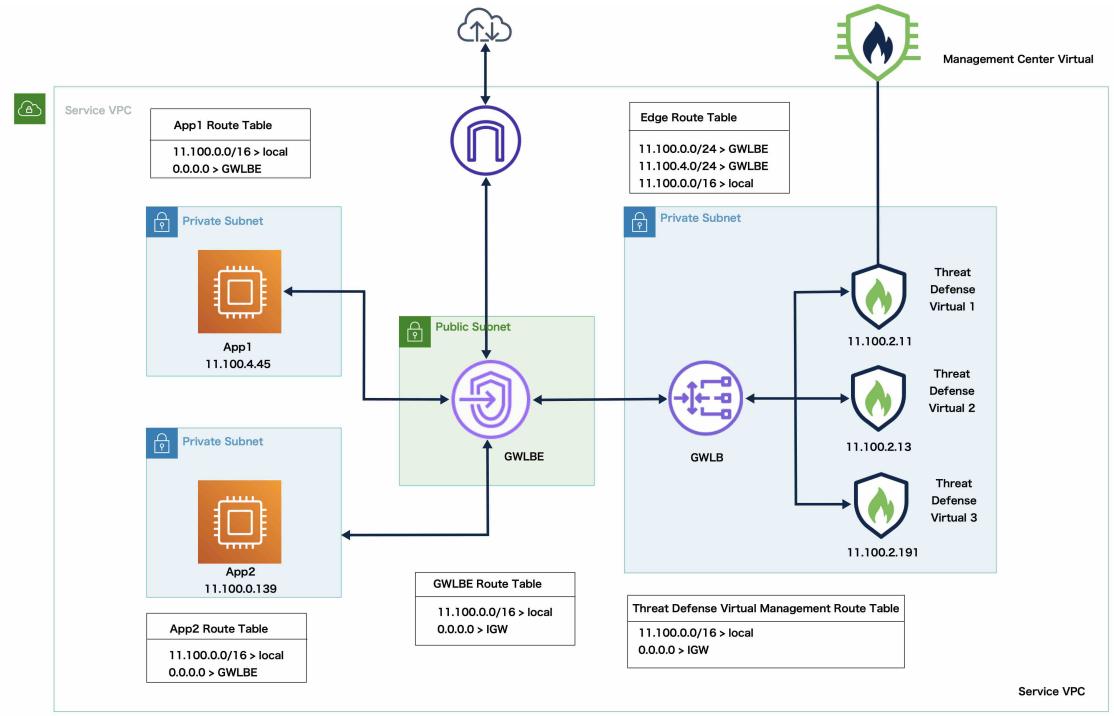
（注） Transport Layer Security（TLS）サーバーアイデンティティ検出は、AWS での Geneve シングルアームセットアップではサポートされていません。

図 *1 : Geneve* シングルアームプロキシ



トポロジの例

次に示すトポロジは、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。GWLB に接続されているクラスタには、3 つの Threat Defense Virtual インスタンスがあります。Management Center Virtual インスタンスは、クラスタの管理に使用されます。



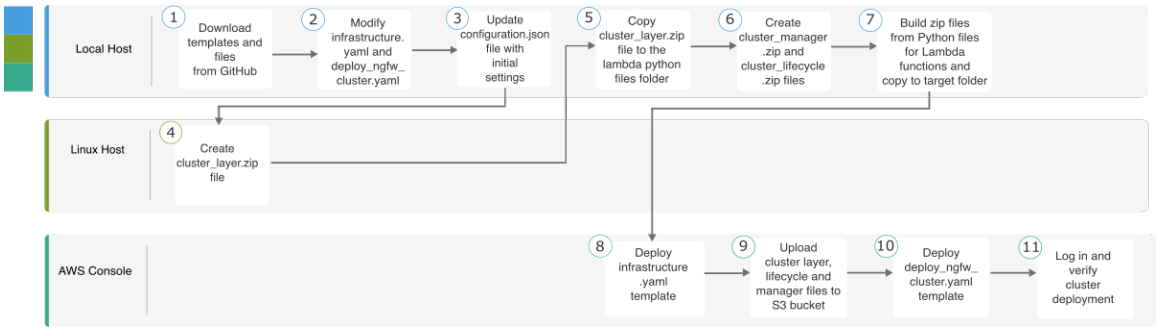
インターネットからの着信トラフィックは、GWLBエンドポイントに送られ、そこからGWLBにトラフィックが送信されます。その後、トラフィックは Threat Defense Virtual クラスタに転送されます。トラフィックは、クラスタ内の Threat Defense Virtual インスタンスによって検査された後、アプリケーション VM App1/App2 に転送されます。

App1/App2からの発信トラフィックは、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。

AWS で Threat Defense 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 Threat Defense 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 からリポジトリを複製します。
2	ローカルホスト	infrastructure.yaml および deploy_ngfw_cluster.yaml テンプレートを変更します。
3	ローカルホスト	Configuration.json ファイルを FMC オブジェクト名で更新します。
4	Linux ホスト	cluster_layer.zip ファイルを作成します。
5	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。
6	ローカルホスト	*cluster_manager.zip*、*custom_metrics_publisher.zip*、および *cluster_lifecycle.zip* ファイルを作成します。
7	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。
8	AWS コンソール	Infrastructure.yaml テンプレートを展開します。
9	AWS コンソール	*cluster_layer.zip*、*cluster_lifecycle.zip*、*custom_metrics_publisher.zip*、および *cluster_manager.zip* を S3 バケットにアップロードします。
10	AWS コンソール	deploy_ngfw_cluster.yaml テンプレートを展開します。
11	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 Threat Defense 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	Threat Defense 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。
7	Management Center	制御ノードを登録します。

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_ngfw_cluster.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_ngfw_cluster.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。

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 Amazon Linux 仮想マシンが必要です。
- クラスタが Firewall Management Center に自動登録されるようにするには、REST API を使用できる管理者権限を持つ *2* つのユーザーを Firewall Management Center で作成する必要があります。Cisco Secure Firewall Management Center アドミニストレーション ガイドを参照してください。
- `Configuration.json` で指定したポリシー名と一致するアクセスポリシーを Firewall Management Center に追加します。

手順

ステップ 1 テンプレートを準備します。

a) 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-ftdv/tree/master/cluster/aws を参照してください。

b) 必要なパラメーターを使用して、**infrastructure.yaml** および **deploy_ngfw_cluster.yaml** を変更します。

c) **cluster/aws/lambda-python-files/Configuration.json** を初期設定に変更します。

次に例を示します。

```
{
  "licenseCaps": ["BASE", "MALWARE", "THREAT"],
  "performanceTier": "FTDv50",
  "fmcIpforDeviceReg": "DONTRESOLVE",
  "RegistrationId": "cisco",
  "NatId": "cisco",
  "fmcAccessPolicyName": "AWS-ACL"
}
```

- fmcIpforDeviceReg 設定は DONTRESOLVE のままにします。
- fmcAccessPolicyName は、Firewall Management Center のアクセスポリシーと一致している必要があります。

（注）
FTDv5 および FTDv10 階層はサポートされていません。

d) **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
zipp
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。

e) 結果の **cluster_layer.zip** ファイルを Lambda Python ファイルフォルダ（`cluster/aws/lambda-python-files`）にコピーします。

f) **cluster_layer.zip**、**custom_metrics_publisher.zip**、**cluster_manger.zip**、および **lifecycle_ftdv.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（`cluster/aws/make.py`）内にあります。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）
Management Center Virtual の登録にプライベート IP アドレスを使用している場合は、`cisco-ftdv/cluster/aws/lambda-python-files/constant.py` ファイルで USE_PUBLIC_IP_FOR_FMC_CONN を False に設定していることを確認します。

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。
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) [次へ（Next）]、[スタックの作成（Create stack）] の順にクリックします。
g) 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

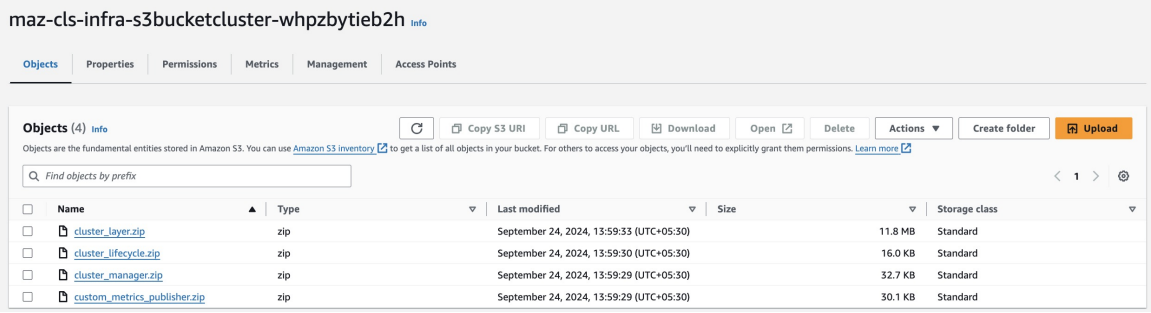
Outputs (13)

Search outputs

Key	Value	Description	Export name
BucketName	maz-cls-infra-s3bucketcluster-whpzbytieb2h	Name of the Amazon S3 bucket	-
BucketUrl	http://maz-cls-infra-s3bucketcluster-whpzbytieb2h.s3-website-us-east-1.amazonaws.com	URL of S3 Bucket Static Website	-
CCLSubnetIds	subnet-0bc04e2cc9e53e5c0,subnet-0d7d046a0fca25615,subnet-03ef42bf527551569	List of CCL subnet IDs (comma seperated)	-
EIPforNATgw	3.218.44.132	EIP reserved for NAT GW	-
FmcInstanceSGID	sg-076880aa64df2db5c	Security Group ID for FMC if user would like to launch in this VPC itself	-
InInterfaceSGId	sg-06ed933d6624fe51b	Security Group ID for Inside Interfaces	-
InsideSubnetIds	subnet-03d12cab8ee0eafff,subnet-0be9158b0970aebab,subnet-0b53c96fceb7c1f4d	List of Inside subnet IDs (comma seperated)	-
InstanceSGId	sg-0680b74be473186aa	Security Group ID for Instances Management Interface	-
LambdaSecurityGroupId	sg-057da2a9954e0d204	Security Group ID for Lambda Functions	-
LambdaSubnetIds	subnet-03439803d989e6bdf,subnet-087488a9d6ffc95cd	List of lambda subnet IDs (comma seperated)	-
ListOfAZs	us-east-1a,us-east-1b,us-east-1c	Availability zones for NGFWv instances	-
MgmtSubnetIds	subnet-06f0bbbd3f207a504,subnet-0c339dc43688cddc9,subnet-0a67629632a655de7	List of Mangement subnet IDs (comma seperated)	-
VpcName	vpc-09c2b0ad995e2fb24	Name of the VPC created	-

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

図 *3 : S3* バケット



（注）
Lambda NAT ゲートウェイのエラスチック IP アドレスが Management Center Virtual に関連付けられたセキュリティグループに追加されていることを確認してください。

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

a) [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 **deploy_ngfw_cluster.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。

d) 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。
ClusterNumber	文字列	これはクラスタ番号であり、クラスタ名（msa-ftdv-infra）のサフィックスとして追加されます。たとえば、この値が「1」の場合、グループ名は msa-ftdv-infra-1 になります。 1 桁以上 3 桁以下である必要があります。デフォルト：1。
ClusterSize	番号	これは、クラスタ内の Firewall Threat Defense Virtual ノードの総数です。 最小値：1

パラメータ	使用できる値/タイプ	説明
		最大値：16
インフラストラクチャの詳細		
NoOfAZs	文字列	これは、Firewall Threat Defense 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
S3BktName	文字列	アップロードされた Lambda zip ファイルを含む S3 バケット。正しいバケット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、Firewall Threat

パラメータ	使用できる値/タイプ	説明
		Defense 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、Firewall Threat Defense 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	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、Firewall Threat Defense 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>
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 アドレスプールから Firewall Threat Defense Virtual インスタンスの CCL インターフェイスに割り当てられます。
MgmtInterfaceSG	リスト	Firewall Threat Defense 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	Firewall Threat Defense Virtual 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。 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	Firewall Threat Defense 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。
GWLBESubnetId	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。 [再調整（rebalance）] は、Firewall Threat Defense Virtual バージョン 7.4.1 以降でサポートされています。
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 Firewall Threat Defense 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 Firewall Threat Defense 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 Firewall Threat Defense Virtual のパブリック IP アドレスを割り当てるには、値を [はい（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。

パラメータ	使用できる値/タイプ	説明
		Firewall Threat Defense Virtual 7.2 以降ではクラスタリングがサポートされ、Firewall Threat Defense Virtual バージョン 7.6 以降では自動スケーリングと複数の可用性ゾーンの機能拡張がサポートされています。 タイプ：AWS::EC2::Image::Id
ngfwPassword	文字列	Firewall Threat Defense Virtual インスタンスのパスワード。 すべての Firewall Threat Defense Virtual インスタンスには、起動テンプレート（クラスタグループ）の [ユーザーデータ（Userdata）] フィールドにあるデフォルトのパスワードが設定されています。 Firewall Threat Defense Virtual にアクセスできるようになると、パスワードがアクティブになります。 文字数は 8 文字以上にする必要があります。パスワードには、プレーンテキストのパスワードまたはKMS暗号化パスワードを使用できます。
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。 このフィールドに値を指定する場合、Firewall Threat Defense 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："*aws kms encrypt --key-id <KMS ARN> --plaintext <password>* " 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。
FMC 自動化の設定		
fmcDeviceGrpName	文字列	Management Center でクラスタグループの一意の名前を入力します。
fmcPublishMetrics	文字列	Management Center をポーリングし、特定のデバイスグループメトリックを AWS CloudWatch にパブリッシュする Lambda 関数を作成するには、**`true`** を選択します。 使用可能な値： • true

パラメータ	使用できる値/タイプ	説明
		• false デフォルトでは、この値は **true** に設定されています。
fmcMetricsUsername	文字列	Management Center からメモリメトリックをポーリングするための一意の内部ユーザー名を入力します。 ユーザーには、**ネットワーク管理者**および**メンテナンスユーザー**以上の権限が必要です。
fmcMetricsPassword	文字列	パスワードを入力します。 KMS マスターキー ARN パラメータを指定した場合は、必ず、暗号化されたパスワードを入力してください。 間違ったパスワードを入力するとメトリック収集が失敗する可能性があるため、必ず、正しいパスワードを入力してください。
fmcServer	文字列	IP アドレスには、外部 IP アドレス、または VPC の Firewall Threat Defense Virtual 管理サブネットで到達可能な IP アドレスを指定できます。 最小長：7 最大長：15
fmcOperationsUsername	文字列	CloudWatch 用の Firewall Management Center Virtual に使用する一意の内部ユーザー名を入力します。 ユーザーには**管理者**権限が必要です。
fmcOperationsPassword	文字列	パスワードを入力します。 KMS マスターキー ARN パラメータを指定した場合は、必ず、暗号化されたパスワードを入力してください。
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。

パラメータ	使用できる値/タイプ	説明
MemoryThresholds	カンマ区切りリスト	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、メモリ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。

e) [次へ（Next）]、[スタックの作成（Create stack）] の順にクリックします。

Lambda 関数が残りのプロセスを管理し、Firewall Threat Defense Virtual が自動的に Firewall Management Center に登録されます。

図 *4 :* 展開されたリソース

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** コマンドを使用して、クラスタの展開を確認します。

図 *5:* クラスタ ノード

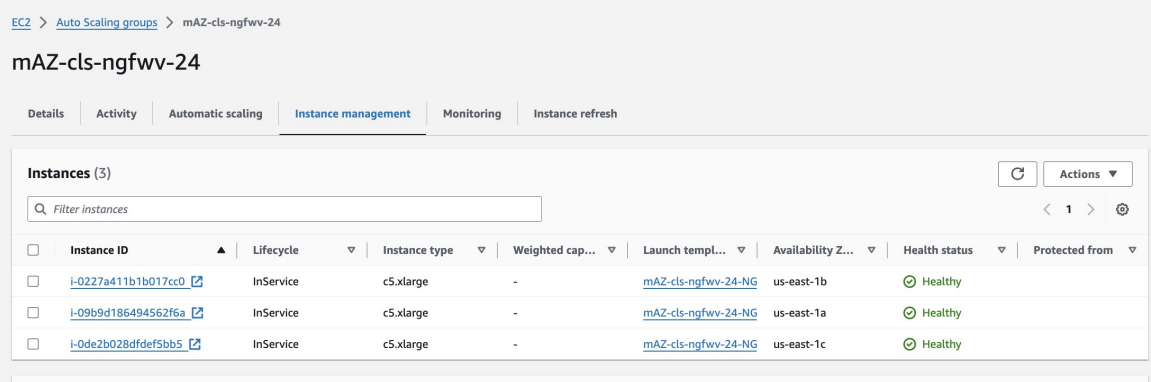
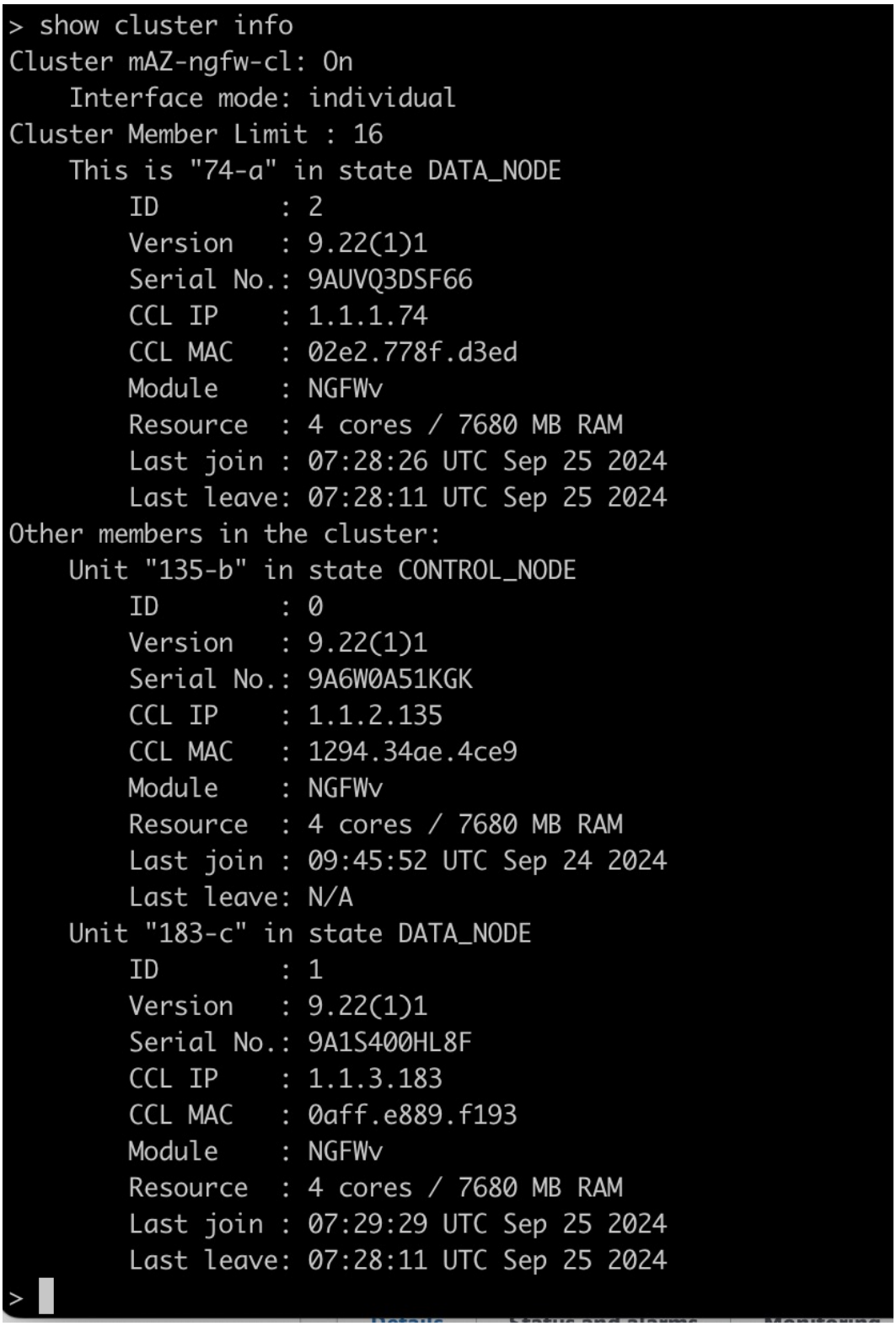


図 *6 : show cluster info*



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、各ノードを展開してから制御ノードを Firewall Management Center に追加します。

AWS 向け Day 0 構成の作成

固定構成またはカスタマイズ構成のいずれかを使用できます。固定構成の使用をお勧めします。

AWS 向け固定構成を使用した Day 0 構成の作成

固定構成により、クラスタのブートストラップ構成が自動生成されます。

単一の可用性ゾーン：AWS 向け固定構成を使用した Day 0 構成

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "Cluster": {
        "CclSubnetRange": "ip_address_start ip_address_end",
        "ClusterGroupName": "cluster_name",
        [For Gateway Load Balancer] "Geneve": "{Yes | No}",
        [For Gateway Load Balancer] "HealthProbePort": "port"
    }
}
```

次に例を示します。

```
{
 "AdminPassword": "Sup3rnatural",
 "Hostname": "ciscoftdv",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "Cluster": {
             "CclSubnetRange": "10.5.90.4 10.5.90.30",
  "ClusterGroupName": "ftdv-cluster",
  "Geneve": "Yes",
  "HealthProbePort": "7777"
 }
}
```

CclSubnetRange 変数には、x.x.x.4 から始まる IP アドレスの範囲を指定します。クラスタリングに使用可能な IP アドレスが 16 個以上あることを確認します。開始（`ip_address_start`）および終了（`ip_address_end`）IP アドレスの例を以下に示します。

表 *2:* 開始 *IP* アドレスと終了 *IP* アドレスの例

CIDR	開始 IP アドレス	終了 IP アドレス
10.1.1.0/27	10.1.1.4	10.1.1.30

CIDR	開始 IP アドレス	終了 IP アドレス
10.1.1.32/27	10.1.1.36	10.1.1.62
10.1.1.64/27	10.1.1.68	10.1.1.94
10.1.1.96/27	10.1.1.100	10.1.1.126
10.1.1.128/27	10.1.1.132	10.1.1.158
10.1.1.160/27	10.1.1.164	10.1.1.190
10.1.1.192/27	10.1.1.196	10.1.1.222
10.1.1.224/27	10.1.1.228	10.1.1.254
10.1.1.0/24	10.1.1.4	10.1.1.254

クラスタノードの展開

クラスタが形成されるよう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 4 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 5 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、Threat Defense 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 4 つのインターフェイス：管理、診断、内部、クラスタ制御リンク。
- 非ネイティブロードバランサの 5 つのインターフェイス：管理、診断、内部、外部、クラスタ制御リンク。

AWS での Threat Defense Virtual の展開の詳細については、「Deploy the Threat Defense 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 Threat Defense 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。

ステップ 5 Management Center に制御ノードを追加します。「Management Center へのクラスタの追加（手動展開） （60 ページ）」を参照してください。

Azure でクラスタを展開する

Azure Gateway Load Balancer（GWLB）、または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。Azure でクラスタを展開するには、Azure Resource Manager（ARM）テンプレートを使用して仮想マシンスケールセットを展開します。

GWLB ベースのクラスタ展開のサンプルトポロジ

図 *7: GWLB* を使用する着信トラフィックの導入例とトポロジ

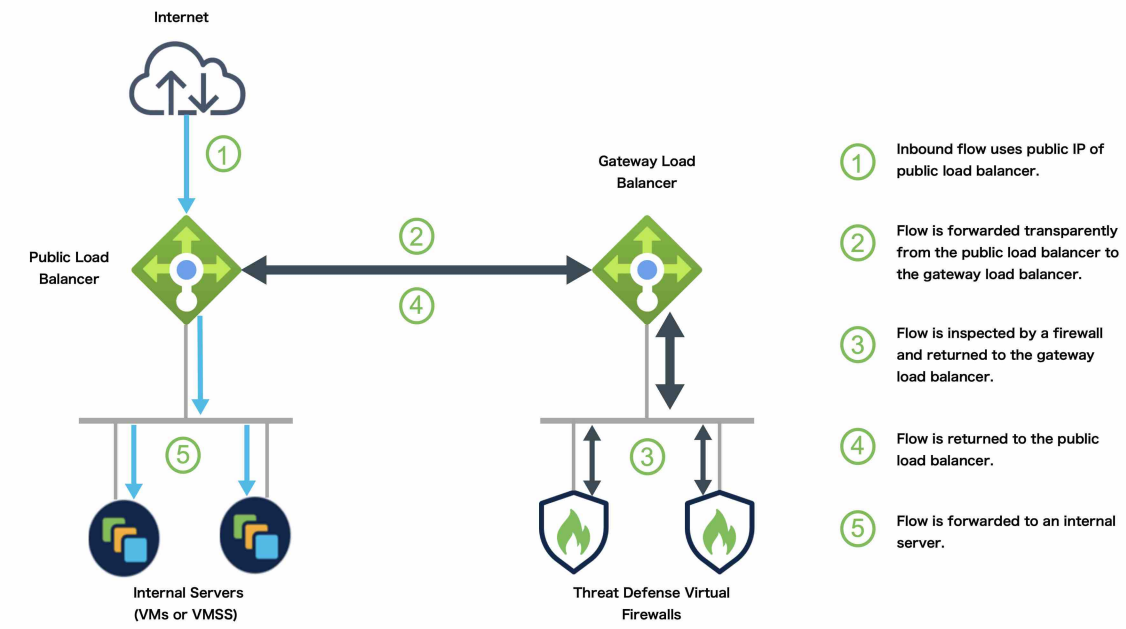
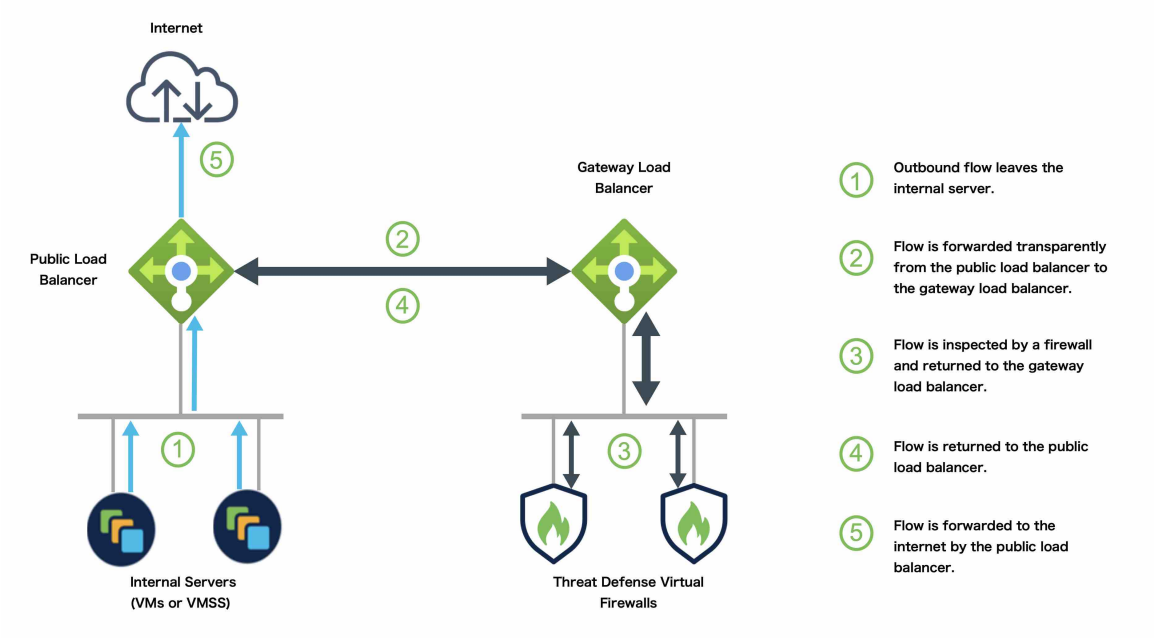


図 *8 : GWLB* を使用する発信トラフィックの導入例とトポロジ

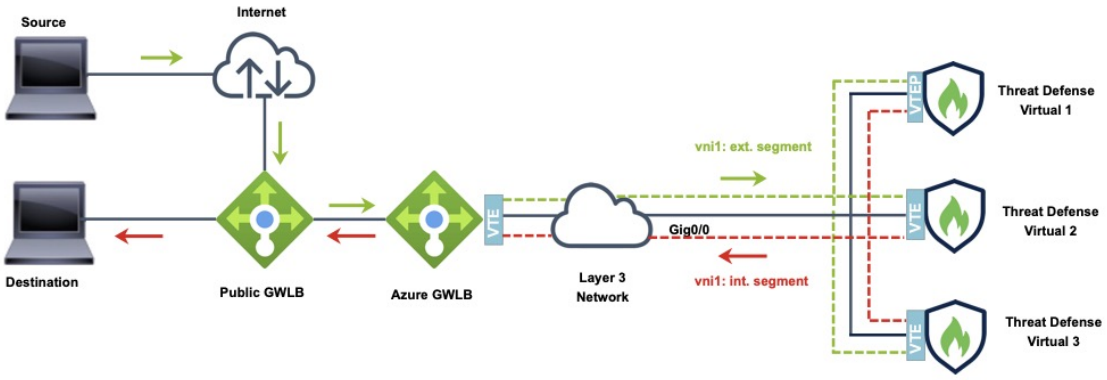


Azure ゲートウェイロードバランサおよびペアプロキシ

Azure サービスチェーンでは、Threat Defense 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。Threat Defense Virtual は、ペアプロキシの VXLAN セグメントを利用して、単一の NIC に外部インターフェイスと内部インターフェイスを定義します。

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 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 Threat Defense Virtual 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 VXLAN セグメント上のゲートウェイロードバランサに送り返す前に検査します。Azure ゲートウェイロードバランサは、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。

図 *9*: ペアリングされたプロキシを使用した *Azure Gateway* ロードバランサ

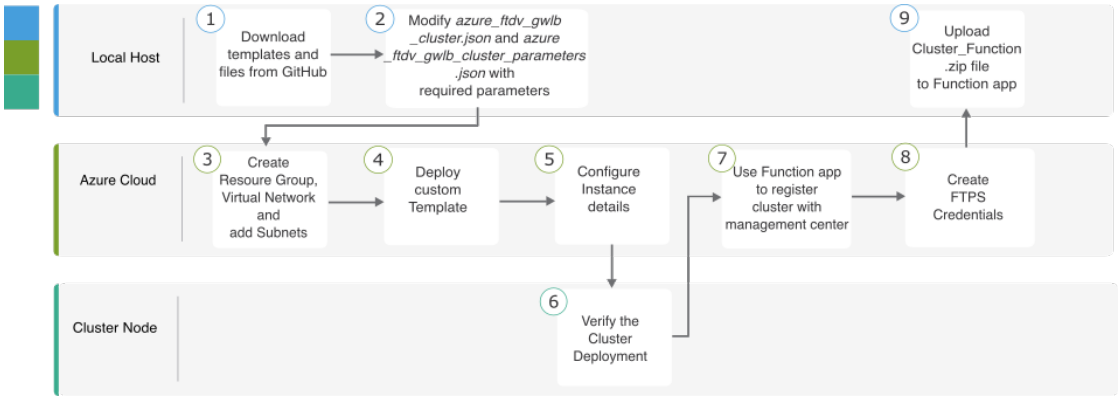


Traffic flow between GWLBe to GWLB (Geneve Single-Arm Proxy) in Azure

GWLB を使用して Azure で Threat Defense 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、GWLB を使用した Azure での Threat Defense 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。

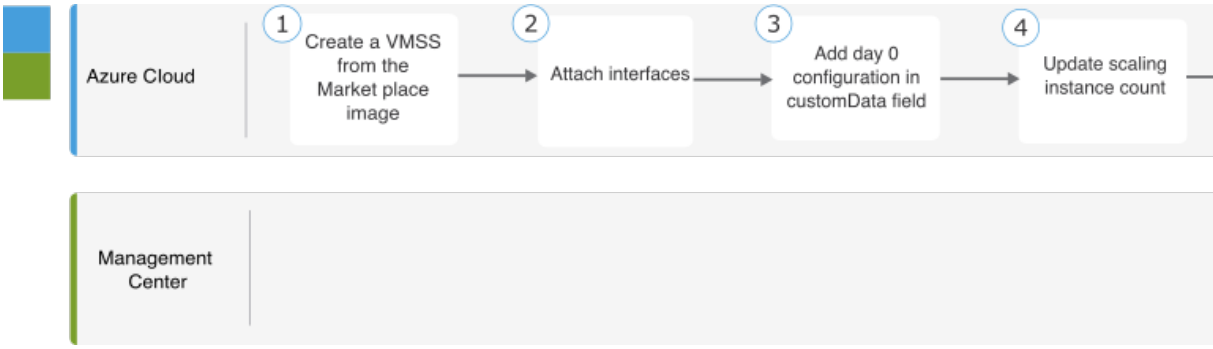


	ワークスペース	手順
1	ローカルホスト	GitHub からテンプレートとファイルをダウンロードします。
2	ローカルホスト	azure_ftdv_gwlb_cluster.json と azure_ftdv_gwlb_cluster_parameters.json を必要なパラメータで変更します。

	ワークスペース	手順
3	Azure Cloud	リソースグループ、仮想ネットワーク、およびサブネットを作成します。
4	Azure Cloud	カスタムテンプレートを展開します。
5	Azure Cloud	インスタンスの詳細を設定します。
6	クラスタノード	クラスタの展開を確認します。
7	Azure Cloud	Function アプリを使用して Management Center にクラスタを登録します。
8	Azure Cloud	FTPS のログイン情報を作成します。
9	ローカルホスト	Cluster_Function.zip ファイルを Function アプリにアップロードします。

手動展開

次のフローチャートは、GWLB を使用した Azure での Threat Defense 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	Marketplace イメージから VMSS を作成します。
2	ローカルホスト	インターフェイスを接続します。
3	ローカルホスト	[customData] フィールドに Day 0 構成を追加します。
4	ローカルホスト	スケーリングインスタンス数を更新します。
5	ローカルホスト	GWLB を設定します。

	ワークスペース	手順
6	Management Center	制御ノードを追加します。

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、および値であり、自明です。

- azure_ftdv_gwlb_cluster_parameters.json：GWLB を使用する Firewall Threat Defense Virtual クラスタのパラメータを入力するためのテンプレート。
- azure_ftdv_gwlb_cluster.json：GWLB を使用する Firewall Threat Defense Virtual クラスタを展開するためのテンプレート。

前提条件

- クラスタが Management Center に自動登録できるようにするには、Management Center でネットワーク管理者およびメンテナンスのユーザー権限を持つユーザーを作成します。これらの権限を持つユーザーは、REST API を使用できます。『Cisco Secure Firewall Management Center Administration Guide』を参照してください。
- テンプレートの展開時に指定するポリシー名と一致するアクセスポリシーを Management Center に追加します。
- Management Center Virtual が適切にライセンスされていることを確認します。
- クラスタが Management Center Virtual に追加されたら、次の手順を実行します。

1. Management Center のプラットフォーム設定でヘルスチェックのポート番号を設定します。この設定の詳細については、「Platform Settings」を参照してください。
2. データトラフィックのスタティックルートを作成します。スタティックルートの作成の詳細については、「Add a Static Route」を参照してください。

   スタティックルートの設定例：

```
Network: any-ipv4
Interface: vxlan_tunnel
Leaked from Virtual Router: Global
Gateway: vxlan_tunnel_gw
Tunneled: false
Metric: 2
```

（注） *vxlan_tunnel_gw* は、データサブネットのゲートウェイ IP アドレスです。

Azure Resource Manager テンプレートを使用した Azure と GWLB でのクラスタの展開

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します。

手順

ステップ 1 テンプレートを準備します。

a) 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-ftdv/tree/master/cluster/azure を参照してください。

b) azure_ftdv_gwlb_cluster.json と azure_ftdv_gwlb_cluster_parameters.json を必要なパラメータで変更します。

ステップ 2 Azure ポータルにログイン：https://portal.azure.com。

ステップ 3 リソース グループを作成します。

a) [基本（Basics）] タブで、ドロップダウンリストから [サブスクリプション（Subscription）] および [リソースグループ（Resource group）] を選択します。

b) 必須の [リージョン（Region）] を選択します。

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

a) 仮想ネットワークを作成します。

1. [基本（Basics）] タブで、ドロップダウンリストから [サブスクリプション（Subscription）] および [リソースグループ（Resource group）] を選択します。
2. 必須の [リージョン（Region）] を選択します。[次へ：IPアドレス（Next: IP addresses）] をクリックします。

[IPアドレス（IP Addresses）] タブで、[サブネットの追加（Add subnet）] をクリックし、管理、診断、データ、およびクラスタ制御リンクのサブネットを追加します。

b) サブネットを追加します。

ステップ 5 カスタムテンプレートを展開します。

a) **[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。

b) [エディタで独自のテンプレートを構築する（Build your own template in the editor）] をクリックします。

c) [ファイルのロード（Load File）] をクリックし、**azure_ftdv_gwlb_cluster.json**

d) [保存（Save）] をクリックします。

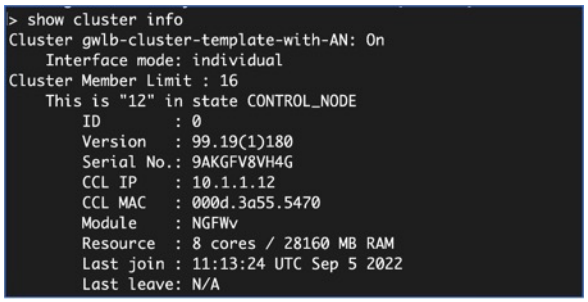
ステップ 6 インスタンスの詳細を設定します。

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

b) 検証に合格したら、[作成（Create）] をクリックします。

ステップ 7 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。

図 *10 : show cluster info*



ステップ 8 Azure ポータルで、Function アプリをクリックしてクラスタを Firewall Management Center に登録します。

（注）
Function アプリを使用しない場合は、**[追加（Add）]** > **[デバイス（Device）]**（ **[追加（Add）]** > **[クラスタ（Cluster）]** ではない）を使用して、制御ノードを Firewall Management Center に直接登録することもできます。その他のクラスタノードは自動的に登録されます。

ステップ 9 **[展開センター（Deployment Center）]** > **[FTPSのログイン情報（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 のログイン情報を作成し、[保存（Save）] をクリックします。

ステップ 10 ローカルの端末で次の **curl** コマンドを実行し、Cluster_Function.zip ファイルを Function アプリにアップロードします。

curl -X POST -u ユーザー名 **--data-binary @"Cluster_Function.zip" https://** *Function_App_Name***.scm.azurewebsites.net/api/zipdeploy**

（注）
curl コマンドは、実行が完了するまでに数分（2 分未満～ 3 分）かかる場合があります。

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。Management Center へのデバイス登録が開始されます。

図 *11* : 機能

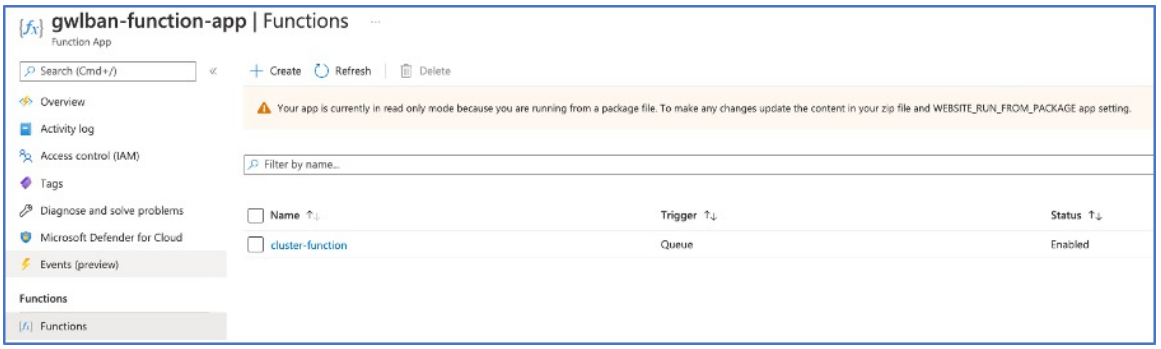


図 *12* : キュー

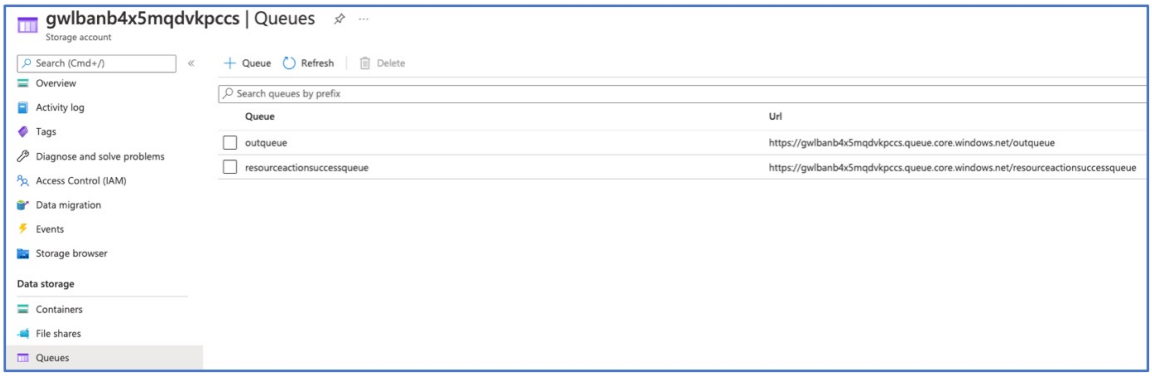
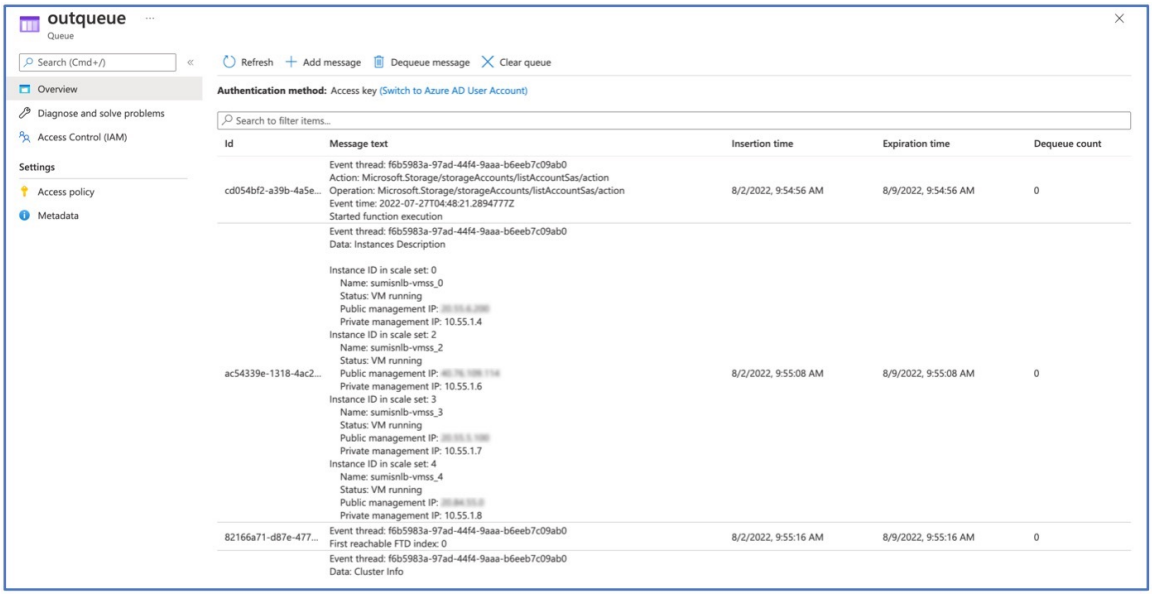
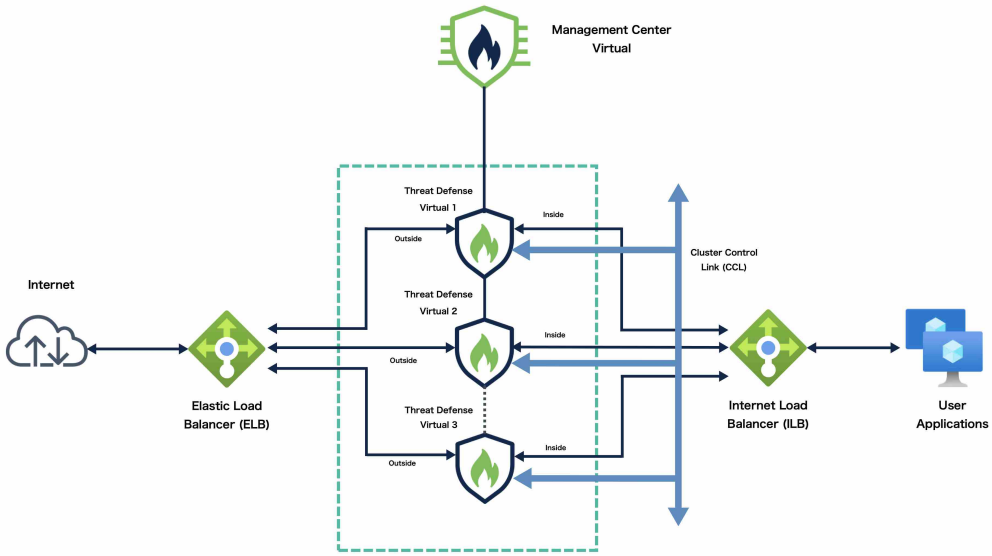


図 *13* : アウトキュー



NLB ベースのクラスタ展開のサンプルトポロジ



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。Threat Defense 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挟まれています。Management Center Virtual インスタンスは、クラスタの管理に使用されます。

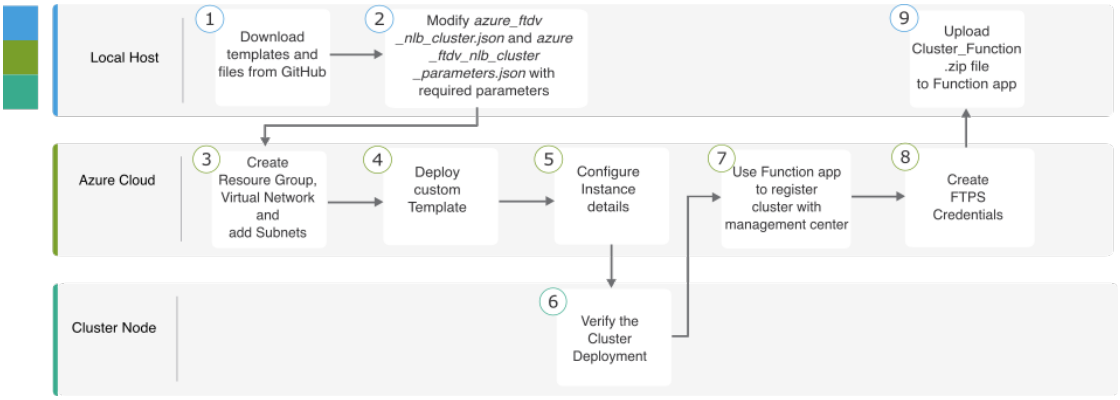
インターネットからの着信トラフィックは、外部ロードバランサに送られ、そこから Threat Defense Virtual クラスタにトラフィックが送信されます。トラフィックは、クラスタ内の Threat Defense Virtual インスタンスによって検査された後、アプリケーション VM に転送されます。

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。その後、トラフィックは Threat Defense Virtual クラスタに転送され、インターネットに送信されます。

NLB を使用して Azure で Threat Defense 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

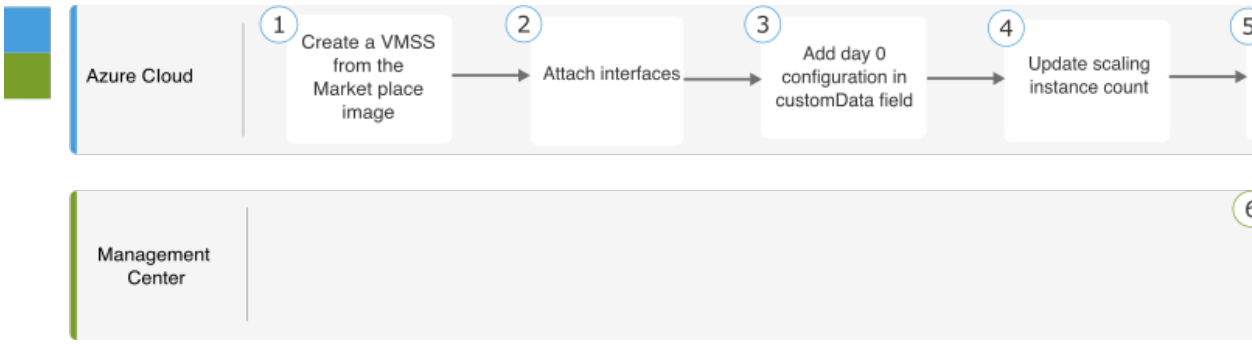
次のフローチャートは、NLB を使用した Azure での Threat Defense Virtual クラスタのテンプレートベース展開のワークフローを示しています。



	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	azure_ftdv_nlb_cluster.json と azure_ftdv_nlb_cluster_parameters.json を必要なパラメータで変更します。
3	Azure Cloud	リソースグループ、仮想ネットワーク、およびサブネットを作成します。
4	Azure Cloud	カスタムテンプレートを展開します。
5	Azure Cloud	インスタンスの詳細を設定します。
6	クラスタノード	クラスタの展開を確認します。
7	Azure Cloud	Function アプリを使用して Management Center にクラスタを登録します。
8	Azure Cloud	FTPS のログイン情報を作成します。
9	ローカルホスト	Cluster_Function.zip ファイルを Function アプリにアップロードします。

手動展開

次のフローチャートは、NLB を使用した Azure での Threat Defense 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	Marketplace イメージから VMSS を作成します。
2	ローカルホスト	インターフェイスを接続します。
3	ローカルホスト	[customData] フィールドに Day 0 構成を追加します。
4	ローカルホスト	スケーリングインスタンス数を更新します。
5	ローカルホスト	NLB を設定します。
6	Management Center	制御ノードを追加します。

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、および値であり、自明です。

- azure_ftdv_nlb_cluster_parameters.json：NLB を使用して Threat Defense Virtual クラスタのパラメータを入力するためのテンプレート。
- azure_ftdv_nlb_cluster.json：NLB を使用して Threat Defense Virtual クラスタを展開するためのテンプレート。

前提条件

- クラスタが Management Center に自動登録できるようにするには、Management Center でネットワーク管理者およびメンテナンスのユーザー権限を持つユーザーを作成します。これらの権限を持つユーザーは、REST API を使用できます。『Cisco Secure Firewall Management Center Administration Guide』を参照してください。
- テンプレートの展開時に指定するポリシー名と一致するアクセスポリシーを Management Center に追加します。

- Management Center Virtual が適切にライセンスされていることを確認します。
- クラスタが Management Center Virtual に追加されたら、次の手順を実行します。

1. Management Center のプラットフォーム設定でヘルスチェックのポート番号を設定します。この設定の詳細については、「Platform Settings」を参照してください。
2. 外部および内部インターフェイスからのトラフィックのスタティックルートを作成します。スタティックルートの作成の詳細については、「Add a Static Route」を参照してください。

   外部インターフェイスのスタティックルートの設定例：

```
Network: any-ipv4
Interface: outside
Leaked from Virtual Router: Global
Gateway: ftdv-cluster-outside
Tunneled: false
Metric: 10
```

（注） *ftdv-cluster-outside* は、外部サブネットのゲートウェイ IP アドレスです。

   内部インターフェイスのスタティックルートの設定例：

```
Network: any-ipv4
Interface: inside
Leaked from Virtual Router: Global
Gateway: ftdv-cluster-inside-gw
Tunneled: false
Metric: 11
```

（注） *ftdv-cluster-inside-gw* は、内部サブネットのゲートウェイ IP アドレスです。

3. データトラフィックの NAT ルールを設定します。NAT ルールの設定の詳細については、「Network Address Translation」を参照してください。

Azure Resource Manager テンプレートを使用した Azure と NLB でのクラスタの展開

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 NLB のクラスタを展開します。

手順

ステップ 1 テンプレートを準備します。

a) 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-ftdv/tree/master/cluster/azure を参照してください。

b) azure_ftdv_nlb_cluster.json と azure_ftdv_nlb_cluster_parameters.json を必要なパラメータで変更します。

ステップ 2 Azure ポータルにログイン：https://portal.azure.com。

ステップ 3 リソース グループを作成します。

a) [基本（Basics）] タブで、ドロップダウンリストから [サブスクリプション（Subscription）] および [リソースグループ（Resource group）] を選択します。

b) 必須の [リージョン（Region）] を選択します。

ステップ 4 管理、診断、内部、外部、クラスタ制御リンクの 5 つのサブネットを持つ仮想ネットワークを作成します。

a) 仮想ネットワークを作成します。

1. [基本（Basics）] タブで、ドロップダウンリストから [サブスクリプション（Subscription）] および [リソースグループ（Resource group）] を選択します。
2. b）必須の [リージョン（Region）] を選択します。[次へ：IPアドレス（Next: IP addresses）] をクリックします。

b) サブネットを追加します。

[IPアドレス（IP Addresses）] タブで、[サブネットの追加（Add subnet）] をクリックし、管理、診断、内部、外部、およびクラスタ制御リンクのサブネットを追加します。

ステップ 5 カスタムテンプレートを展開します。

a) **[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。

b) [エディタで独自のテンプレートを構築する（Build your own template in the editor）] をクリックします。

c) [ファイルのロード（Load File）] をクリックし、**azure_ftdv_nlb_cluster.json** 。

d) [保存（Save）] をクリックします。

ステップ 6 インスタンスの詳細を設定します。

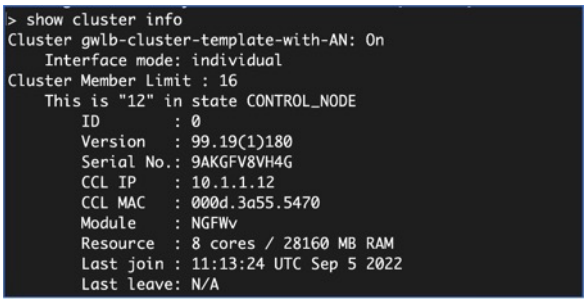
a) 必要な値を入力し、[確認して作成（Review + create）] をクリックします。

（注）
クラスタ制御リンクの開始アドレスと終了アドレスは、必要な数だけ指定してください（最大 16 個）。範囲を大きくすると、パフォーマンスに影響する可能性があります。

b) 検証に合格したら、[作成（Create）] をクリックします。

ステップ 7 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** コマンドを使用して、クラスタの展開を確認します。

図 *14 : show cluster info*



ステップ 8 Azure ポータルで、Function アプリをクリックしてクラスタを Firewall Management Center に登録します。

（注）
Function アプリを使用しない場合は、[追加（Add）] > [デバイス（Device）]（[追加（Add）] > [クラスタ（Cluster）] ではない）を使用して、制御ノードを Management Center に直接登録することもできます。その他のクラスタノードは自動的に登録されます。

ステップ 9 **[展開センター（Deployment Center）] > [FTPSのログイン情報（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 のログイン情報を作成し、[保存（Save）] をクリックします。

ステップ 10 ローカルの端末で次の **curl** コマンドを実行し、Cluster_Function.zip ファイルを Function アプリにアップロードします。

curl -X POST -u ユーザー名 **--data-binary @"Cluster_Function.zip" https://** *Function_App_Name***.scm.azurewebsites.net/api/zipdeploy**

（注）
curl コマンドは、実行が完了するまでに数分（2 分未満～ 3 分）かかる場合があります。

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。Management Center へのデバイス登録が開始されます。

Azure 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、Day0 構成を準備し、各ノードを展開してから制御ノードを Firewall Management Center に追加します。

Azure 向け Day 0 構成の作成

固定構成またはカスタマイズ構成のいずれかを使用できます。

Azure 向け固定構成を使用した Day 0 構成の作成

固定構成により、クラスタのブートストラップ構成が自動生成されます。

```
{
 "AdminPassword": "password",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "Diagnostic": "OFF",    //For deployment of version 7.4.1 and later without Diagnostics
 template, set this parameter to OFF.
. "FmcIp": "<FMC_IP>",
 "FmcRegKey": "<REGISTRATION_KEY>",
 "FmcNatId": "<NAT_ID>",
 "Cluster": {
  "CclSubnetRange": "ip_address_start ip_address_end",
  "ClusterGroupName": "cluster_name",
  "HealthProbePort": "port_number",
  "GatewayLoadBalancerIP": "ip_address",
  "EncapsulationType": "vxlan",
  "InternalPort": "internal_port_number",
  "ExternalPort": "external_port_number",
  "InternalSegId": "internal_segment_id",
  "ExternalSegId": "external_segment_id"
 }
}
```

例

次に、Day 0 構成の例を示します。

```
{
 "AdminPassword": "password",
 "FirewallMode": "routed",
 "ManageLocally": "No",
 "Diagnostic": "OFF",    //For deployment of version 7.4.1 and later without Diagnostics
 template, set this parameter to OFF.
 "FmcIp":"<FMC_IP>",
 "FmcRegKey":"<REGISTRATION_KEY>",
 "FmcNatId":"<NAT_ID>",
 "Cluster": {
  "CclSubnetRange": "10.45.3.4 10.45.3.30",     //mandatory user input
  "ClusterGroupName": "ngfwv-cluster",          //mandatory user input
  "HealthProbePort": "7777",                    //mandatory user input
  "GatewayLoadBalancerIP": "10.45.2.4",          //mandatory user input
  "EncapsulationType": "vxlan",
  "InternalPort": "2000",
  "ExternalPort": "2001",
  "InternalSegId": "800",
  "ExternalSegId": "801"
 }
}
```

（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。

Azureヘルスチェックの設定では、ここで設定した**HealthProbePort**を必ず指定してください。

CclSubnetRange 変数には、x.x.x.4 から始まる IP アドレスの範囲を指定します。クラスタリングに使用可能な IP アドレスが 16 個以上あることを確認します。開始 IP アドレスと終了 IP アドレスの例を次に示します。

表 *3*: 開始 *IP* アドレスと終了 *IP* アドレスの例

CIDR	開始 IP アドレス	終了 IP アドレス
10.1.1.0/27	10.1.1.4	10.1.1.30
10.1.1.32/27	10.1.1.36	10.1.1.62
10.1.1.64/27	10.1.1.68	10.1.1.94
10.1.1.96/27	10.1.1.100	10.1.1.126
10.1.1.128/27	10.1.1.132	10.1.1.158
10.1.1.160/27	10.1.1.164	10.1.1.190
10.1.1.192/27	10.1.1.196	10.1.1.222
10.1.1.224/27	10.1.1.228	10.1.1.254

Azure 向けカスタマイズ構成を使用した Day 0 構成の作成

コマンドを使用して、クラスタのブートストラップ設定をすべて入力できます。

```
{
 "AdminPassword": "password",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "Diagnostic": "OFF",    //For deployment of version 7.4.1 and later without Diagnostics
 template, set this parameter to OFF.
 "FmcIp": "<FMC_IP>",
 "FmcRegKey": "<REGISTRATION_KEY>",
 "FmcNatId": "<NAT_ID>",
 "Cluster": {
  "CclSubnetRange": "ip_address_start ip_address_end",
  "ClusterGroupName": "cluster_name",
  "HealthProbePort": "port_number",
  "GatewayLoadBalancerIP": "ip_address",
  "EncapsulationType": "vxlan",
  "InternalPort": "internal_port_number",
  "ExternalPort": "external_port_number",
  "InternalSegId": "internal_segment_id",
  "ExternalSegId": "external_segment_id"
 }
}
```

例

以下に、**バージョン 7.4 以降**の Day 0 構成の例を示します。

```
{
 "AdminPassword": "Sup3rnatural",
 "Hostname": "clusterftdv",
 "FirewallMode": "routed",
 "ManageLocally": "No",
```

```
"Diagnostic": "OFF",    //For deployment of version 7.4.1 and later without Diagnostics
template, set this parameter to OFF.
"FmcIp": "<FMC_IP>",
"FmcRegKey": "<REGISTRATION_KEY>",
"FmcNatId": "<NAT_ID>",
"run_config": [
 "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 Management0/0",
 "management-only",
 "nameif management",
 "security-level 0",
 "ip address dhcp",
 "interface GigabitEthernet0/0",
 "no shutdown",
 "nameif vxlan_tunnel",
 "security-level 0",
 "ip address dhcp",
 "interface GigabitEthernet0/1",
 "no shutdown",
 "nve-only cluster",
 "nameif ccl_link",
 "security-level 0",
 "ip address dhcp",
 "interface vni1",
 "description Clustering Interface",
 "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 10.45.3.4 10.45.3.30",                          //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",
 "nve 2 ",
 "encapsulation vxlan",
 "source-interface vxlan_tunnel",
 "peer ip <GatewayLoadbalancerIP>",
 "cluster group ftdv-cluster",                          //mandatory user input
 "local-unit 1",
 "cluster-interface vni1 ip 1.1.1.1 255.255.255.0",
 "priority 1",
 "enable",
 "mtu vxlan_tunnel 1454",
 "mtu ccl_link 1454"
]
}
```

以下に、**バージョン 7.3 以前**の Day 0 構成の例を示します。

```
{
 "AdminPassword": "Sup3rnatural",
 "Hostname": "clusterftdv",
 "FirewallMode": "routed",
 "ManageLocally": "No",
 "FmcIp": "<FMC_IP>",
 "FmcRegKey": "<REGISTRATION_KEY>",
 "FmcNatId": "<NAT_ID>",
 "run_config": [
  "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 Management0/0",
  "management-only",
  "nameif management",
  "security-level 0",
  "ip address dhcp",
  "interface GigabitEthernet0/0",
  "no shutdown",
  "nameif vxlan_tunnel",
  "security-level 0",
  "ip address dhcp",
  "interface GigabitEthernet0/1",
  "no shutdown",
  "nve-only cluster",
  "nameif ccl_link",
  "security-level 0",
  "ip address dhcp",
  "interface vni1",
  "description Clustering Interface",
  "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 10.45.3.4 10.45.3.30",                          //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",
  "nve 2 ",
  "encapsulation vxlan",
  "source-interface vxlan_tunnel",
  "peer ip <GatewayLoadbalancerIP>",
  "cluster group ftdv-cluster",                          //mandatory user input
  "local-unit 1",
  "cluster-interface vni1 ip 1.1.1.1 255.255.255.0",
  "priority 1",
```

```
  "enable",
  "mtu vxlan_tunnel 1454",
  "mtu ccl_link 1554"
 ]
}
```

（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。

クラスタノードの手動展開：GWLB ベースの展開

クラスタが形成されるようにクラスタノードを展開します。

手順

ステップ 1 **az vmss create** CLI を使用して、インスタンス数が 0 の Marketplace イメージから仮想マシンスケールセットを作成します。

az vmss create --resource-group <*ResourceGroupName*> **--name** <*VMSSName*> **--vm-sku** <*InstanceSize*> **--image** <*FTDvImage*> **--instance-count 0 --admin-username** <*AdminUserName*> **--admin-password** <*AdminPassword*> **--plan-name** <*ftdv-azure-byol/ftdv-azure-payg*> **--plan-publisher cisco --plan-product cisco-ftdv --plan-promotion-code** <*ftdv-azure-byol/ftdv-azure-payg*> **--vnet-name** <*VirtualNetworkName*> **--subnet** <*MgmtSubnetName*>

ステップ 2 3 つのインターフェイス（診断、データ、およびクラスタ制御リンク）を接続します。

ステップ 3 作成した仮想マシンスケールセットに移動し、次の手順を実行します。

a) [オペレーティングシステム（Operating system）] セクションで、[customData] フィールドに Day 0 構成を追加します。

b) [保存（Save）] をクリックします。

c) [スケーリング（Scaling）] セクションで、インスタンス数を必要なクラスタノードで更新します。インスタンス数は、最小 1、最大 16 の範囲に設定できます。

ステップ 4 Azure ゲートウェイロードバランサを設定します。詳細については、「Azure ゲートウェイロードバランサを使用した Auto Scale の導入例」を参照してください。

ステップ 5 Firewall Management Center に制御ノードを追加します。「Management Center へのクラスタの追加（手動展開） （60 ページ）」を参照してください。

クラスタノードの手動展開：NLB ベースの展開

クラスタが形成されるようにクラスタノードを展開します。

手順

ステップ 1 **az vmss create** CLI を使用して、インスタンス数が 0 の Marketplace イメージから仮想マシンスケールセットを作成します。

az vmss create --resource-group <*ResourceGroupName*> **--name** <*VMSSName*> **--vm-sku** <*InstanceSize*> **--image** <*FTDvImage*> **--instance-count 0 --admin-username** <*AdminUserName*> **--admin-password** <*AdminPassword*> **--plan-name** <*ftdv-azure-byol/ftdv-azure-payg*> **--plan-publisher cisco --plan-product cisco-ftdv --plan-promotion-code** <*ftdv-azure-byol/ftdv-azure-payg*> **--vnet-name** <*VirtualNetworkName*> **--subnet** <*MgmtSubnetName*>

ステップ 2 4 つのインターフェイス（診断、内部、外部、およびクラスタ制御リンク）を接続します。

ステップ 3 作成した仮想マシンスケールセットに移動し、次の手順を実行します。

a) [オペレーティングシステム（Operating system）] セクションで、[customData] フィールドに Day 0 構成を追加します。

b) [保存（Save）] をクリックします。

c) [スケーリング（Scaling）] セクションで、インスタンス数を必要なクラスタノードで更新します。インスタンス数は、最小 1、最大 16 の範囲に設定できます。

ステップ 4 Management Center に制御ノードを追加します。「Management Center へのクラスタの追加（手動展開） （60 ページ）」を参照してください。

Azure でのトラブルシューティング クラスタ展開

- 問題：トラフィックフローがない

  トラブルシューティング：

  - GWLB で展開された Threat Defense Virtual インスタンスの正常性プローブステータスが正常かどうかを確認します。
  - Threat Defense Virtual インスタンスの正常性プローブステータスが異常である場合：
    - Management Center Virtual でスタティックルートが設定されているかどうかを確認します。
    - 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 であるかどうかを確認します。
    - Threat Defense 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かどうかを確認します。
    - Management Center Virtual で設定されたアクセスリストが正常性プローブトラフィックを許可しているかどうかを確認します。

- 問題：クラスタが形成されていない

トラブルシューティング：

- nve-only クラスタインターフェイスの IP アドレスを確認します。他のノードの nve-only のクラスタインターフェイスにピン可能であ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 インターフェイス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適切な VNI 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 があります。
- ノードが相互にピン可能であることを確認します。各ノードに独自のクラスタインターフェイス IP があるため、これらは相互にピン可能である必要があります。
- テンプレート展開中に指定された CCL サブネットの開始アドレスと終了アドレスが正しいかどうかを確認します。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 192.168.1.0/24 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）。
- Management Center Virtual に有効なライセンスがあるかどうかを確認します。

- 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する。

  トラブルシューティング：端末で次のコマンドを使用して、以下のロールを削除します。

  エラー メッセージ：

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

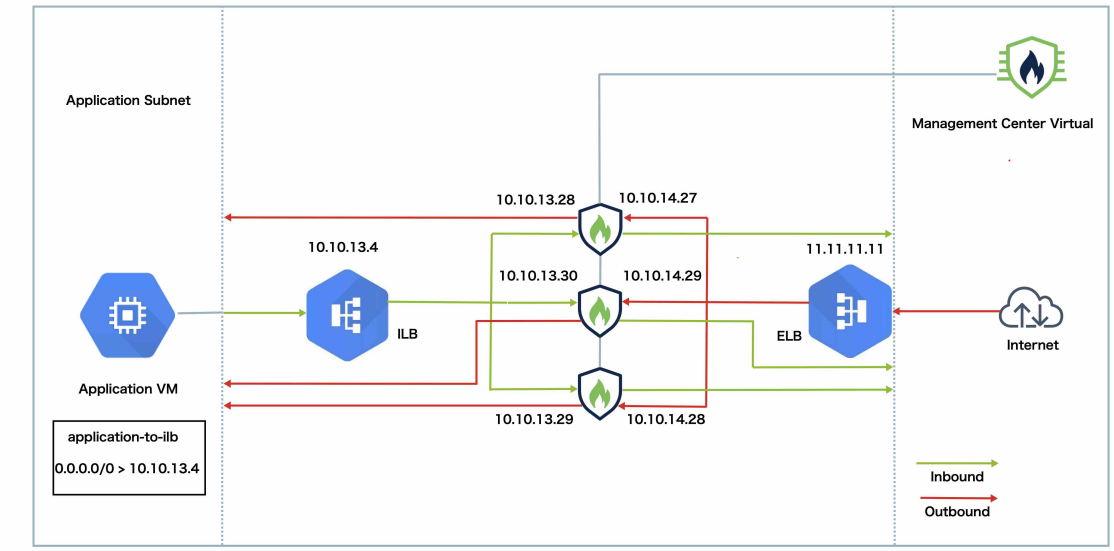
  - **az role assignment delete --resource-group** <リソースグループ名> **--role "Storage Queue Data Contributor"**
  - **az role assignment delete --resource-group** <リソースグループ名> **--role "Contributor"**

GCP でのクラスタの展開

クラスタを GCP で展開するには、手動で展開するか、インスタンステンプレートを使用してインスタンスグループを展開します。GCP 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。

（注） 発信トラフィックはインターフェイス NAT が必要であり、64K 接続に制限されています。

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。Threat Defense 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挟まれています。Management Center Virtual インスタンスは、クラスタの管理に使用されます。

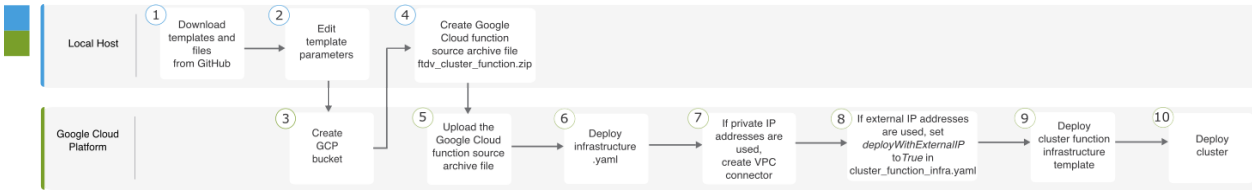
インターネットからの着信トラフィックは、外部ロードバランサに送られ、そこから Threat Defense Virtual クラスタにトラフィックが送信されます。トラフィックは、クラスタ内の Threat Defense Virtual インスタンスによって検査された後、アプリケーション VM に転送されます。

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。その後、トラフィックは Threat Defense Virtual クラスタに転送され、インターネットに送信されます。

GCP で Threat Defense 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

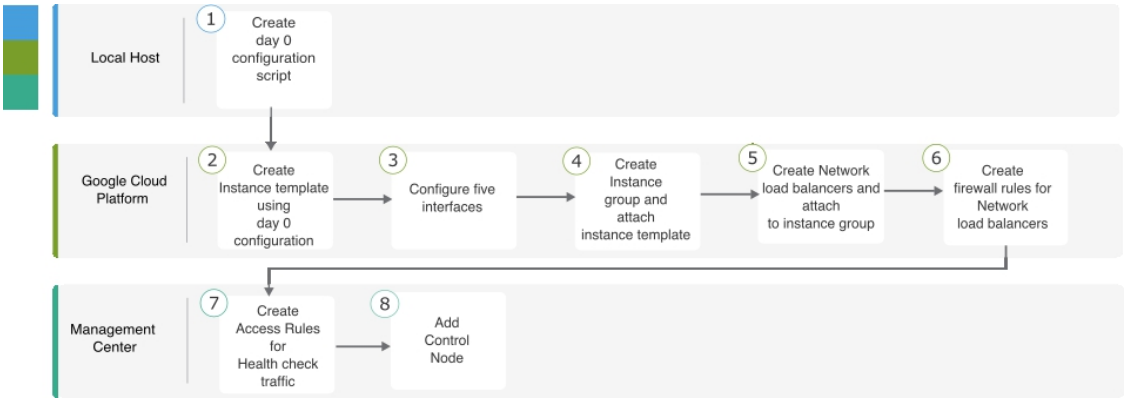
次のフローチャートは、GCP での Threat Defense Virtual クラスタのテンプレートベース展開のワークフローを示しています。



	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	テンプレートパラメータを編集します。
3	Google Cloud Platform	GCP バケットを作成します。
4	ローカルホスト	Google Cloud 関数ソースアーカイブファイル ftdv_cluster_function.zip を作成します。
5	Google Cloud Platform	Google 関数ソースアーカイブファイルをアップロードします。
6	Google Cloud Platform	infrastructure.yaml を展開します。
7	Google Cloud Platform	プライベート IP アドレスが使用されている場合は、VPC コネクタを作成します。
8	Google Cloud Platform	外部 IP アドレスが使用されている場合は、cluster_function_infra.yaml で deployWithExternalIP を True に設定します。
9	Google Cloud Platform	クラスタ機能インフラストラクチャ テンプレートを展開します。
10	Google Cloud Platform	クラスタを展開します。

手動展開

次のフローチャートは、GCP での Threat Defense 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	Google Cloud Platform	Day 0 構成を使用してインスタンステンプレートを作成します。
3	Google Cloud Platform	インターフェイスを設定します。
4	Google Cloud Platform	インスタンスグループを作成し、インスタンステンプレートを割り当てます。
5	Google Cloud Platform	NLB を作成し、インスタンスグループにアタッチします。
6	Google Cloud Platform	NLB のファイアウォールルールを作成します。
7	Management Center	ヘルスチェックトラフィックのアクセスルールを作成します。
8	Management Center	制御ノードを追加します。

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、および値であり、自明です。

- East-West トラフィック用のクラスタ展開テンプレート：deploy_ngfw_cluster_yaml
- North-South トラフィック用のクラスタ展開テンプレート：deploy_ngfw_cluster.yaml

インスタンステンプレートを使用したGCPでのインスタンスグループの展開

インスタンステンプレートを使用して、GCP にインスタンスグループを展開します。

始める前に

- 展開には Google Cloud Shell を使用します。または、任意の macOS/Linux/Windows マシンで Google SDK を使用できます。
- クラスタが Management Center に自動登録されるようにするには、REST API を使用できる管理者権限を持つユーザーを Management Center で作成する必要があります。Cisco Secure Firewall Management Center アドミニストレーション ガイドを参照してください。
- cluster_function_infra.yaml で指定したポリシー名と一致するアクセスポリシーを Management Center に追加します。

手順

ステップ 1 テンプレートを GitHub からローカルフォルダにダウンロードします。

ステップ 2 必要な resourceNamePrefix パラメータ（ngfwvcls など）と他の必要なユーザー入力を使用して、**infrastructure.yaml**、**cluster_function_infra.yaml**、および **deploy_ngfw_cluster.yaml** を編集します。

deploy_ngfw_cluster.yaml ファイルは、GitHub で **east-west** フォルダと **north-south** フォルダの両方にあることに注意してください。トラフィックフローの要件に従って、適切なテンプレートをダウンロードします。

ステップ 3 Google Cloud Shell を使用してバケットを作成し、Google Cloud 関数ソースアーカイブファイル ftdv_cluster_function.zip をアップロードします。

gsutil mb --pap enforced gs://*resourceNamePrefix***-ftdv-cluster-bucket/**

ここでの *resourceNamePrefix* 変数が **cluster_function_infra.yaml** で指定した resourceNamePrefix 変数と一致していることを確認します。

ステップ 4 クラスタ インフラストラクチャのアーカイブファイルを作成します。

例：

```
zip -j ftdv_cluster_function.zip ./cluster-function/*
```

ステップ 5 前に作成した Google ソースアーカイブをアップロードします。

gsutil cp ftdv_cluster_function.zip gs://*resourceNamePrefix***-ftdv-cluster-bucket/**

ステップ 6 クラスタのインフラストラクチャを展開します。

gcloud deployment-manager deployments create *cluster_name* **--config infrastructure.yaml**

ステップ 7 プライベート IP アドレスを使用している場合は、次の手順を実行します。

a) Threat Defense Virtual 管理 VPC を使用して、Management Center Virtual を起動してセットアップします。

b) VPC コネクタを作成して、Google Cloud 関数を Threat Defense Virtual 管理 VPC に接続します。

gcloud compute networks vpc-access connectors create *vpc-connector-name* **--region us-central1 --subnet** *resourceNamePrefix*-**ftdv-mgmt-subnet28**

ステップ 8 Management Center が Threat Defense Virtual からリモートに配置され、Threat Defense Virtual に外部 IP アドレスが必要な場合は、必ず **cluster_function_infra.yaml** で **deployWithExternalIP** を **True** に設定してください。

ステップ 9 クラスタ機能インフラストラクチャを展開します。

gcloud deployment-manager deployments create *cluster_name* **--config cluster_function_infra.yaml**

ステップ 10 クラスタを展開します。

1. North-South トポロジ展開の場合：

   gcloud deployment-manager deployments create *cluster_name* **--config north-south/deploy_ngfw_cluster.yaml**

2. East-West トポロジ展開の場合：

   gcloud deployment-manager deployments create *cluster_name* **--config east-west/deploy_ngfw_cluster.yaml**

GCP 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、Day0 構成を準備し、各ノードを展開してから制御ノードを Firewall Management Center に追加します。

GCP 向け Day 0 構成の作成

固定構成またはカスタマイズ構成のいずれかを使用できます。

GCP 向け固定構成を使用した Day 0 構成の作成

固定構成により、クラスタのブートストラップ構成が自動生成されます。

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
                "Diagnostic": "OFF",
    "Cluster": {
        "CclSubnetRange": "ip_address_start ip_address_end",
        "ClusterGroupName": "cluster_name"
```

```
    }
}
```

次に例を示します。

```
{
    "AdminPassword": "DeanW1nche$ter",
    "Hostname": "ciscoftdv",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "Cluster": {
        "CclSubnetRange": "10.10.55.2 10.10.55.253",      //mandatory user input
        "ClusterGroupName": "ftdv-cluster"                //mandatory user input
    }
}
```

（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。

CclSubnetRange 変数では、サブネット内の最初の 2 つの IP アドレスと最後の 2 つの IP アドレスを使用できないことに注意してください。詳細については、「Reserved IP addresses in IPv4 subnets」を参照してください。クラスタリングに使用可能な IP アドレスが 16 個以上あることを確認します。開始 IP アドレスと終了 IP アドレスの例を次に示します。

表 *4*: 開始 *IP* アドレスと終了 *IP* アドレスの例

CIDR	開始 IP アドレス	終了 IP アドレス
10.1.1.0/27	10.1.1.2	10.1.1.29
10.1.1.32/27	10.1.1.34	10.1.1.61
10.1.1.64/27	10.1.1.66	10.1.1.93
10.1.1.96/27	10.1.1.98	10.1.1.125
10.1.1.128/27	10.1.1.130	10.1.1.157
10.1.1.160/27	10.1.1.162	10.1.1.189
10.1.1.192/27	10.1.1.194	10.1.1.221
10.1.1.224/27	10.1.1.226	10.1.1.253
10.1.1.0/24	10.1.1.2	10.1.1.253

GCP 向けカスタマイズ構成を使用した Day 0 構成の作成

コマンドを使用して、クラスタのブートストラップ設定をすべて入力できます。

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
```

```
    "run_config": [comma_separated_threat_defense_configuration]
    }
```

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構成を作成します。太字の値はノードごとに一意である必要があることに注意してください。

```
{
 "AdminPassword": "W1nch3sterBr0s",
 "Hostname": "ftdv1",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "run_config": [
  "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 vni1",
  "description Clustering Interface",
  "segment-id 1",
  "vtep-nve 1",
  "object network ccl#link",
  "range 10.1.90.2 10.1.90.17",
  "object-group network cluster#group",
  "network-object object ccl#link",
  "nve 1",
  "encapsulation vxlan",
  "source-interface ccl_link",
  "peer-group cluster#group",
  "cluster group ftdv-cluster",
  "local-unit 1",
  "cluster-interface vni1 ip 10.1.1.1 255.255.255.0",
  "priority 1",
  "enable",
  "mtu outside 1400",
  "mtu inside 1400"
 ]
}
```

（注） クラスタ制御リンク ネットワーク オブジェクトには、アドレスを必要な数だけ指定します（最大 16個）。範囲を大きくすると、パフォーマンスに影響する可能性があります。

クラスタノードの手動展開

クラスタが形成されるようにクラスタノードを展開します。GCP でのクラスタリングの場合、4 vCPU マシンタイプは使用できません。4 vCPU マシンタイプがサポートするインターフェイスは 4 つのみですが、インターフェイスは 5 つ必要です。c2-standard-8 など、5 つのインターフェイスがサポートされるマシンタイプを使用します。

手順

ステップ 1 5 つのインターフェイス（外部、内部、管理、診断、クラスタ制御リンク）を備えた Day 0 構成を使用して、インスタンステンプレートを作成します（[メタデータ（Metadata）] > [スタートアップスクリプト（Startup Script）] セクション）。

Cisco Secure Firewall Threat Defense Virtual スタートアップガイド を参照してください。

ステップ 2 インスタンスグループを作成し、インスタンステンプレートを割り当てます。

ステップ 3 GCP ネットワークロードバランサ（内部および外部）を作成し、インスタンスグループを割り当てます。

ステップ 4 GCP ネットワークロードバランサの場合、Management Center のセキュリティポリシーでヘルスチェックを許可します。GCP ネットワークロードバランサのヘルスチェックの許可 （58 ページ） を参照してください。

ステップ 5 Management Center に制御ノードを追加します。「Management Center へのクラスタの追加（手動展開） （60 ページ）」を参照してください。

GCP ネットワークロードバランサのヘルスチェックの許可

Google Cloud は、バックエンドがトラフィックに応答するかどうかを判断するヘルスチェック機能を提供します。

ネットワークロードバランサのファイアウォール ルールを作成するには、「https://cloud.google.com/load-balancing/docs/health-checks」を参照してください。次に、Firewall Management Center でヘルスチェックトラフィックを許可するアクセスルールを作成します。必要なネットワーク範囲については、「https://cloud.google.com/load-balancing/docs/health-check-concepts」を参照してください。 アクセス コントロール ルールを参照してください。

また、動的な手動 NAT ルールを設定して、ヘルスチェックトラフィックを 169.254.169.254 の Google メタデータサーバーにリダイレクトする必要もあります。 ダイナミック手動 NAT の設定を参照してください。

正常性プローブの構成に使用されるすべてのインターフェイスに対する GCP 正常性チェックのルートを設定できます。これは、GCP 正常性チェック用のルートがまだ利用可能になっていないインターフェイスで、より高いメトリックを持つルートを作成することで実現できます。

North-South NAT ルールの設定例

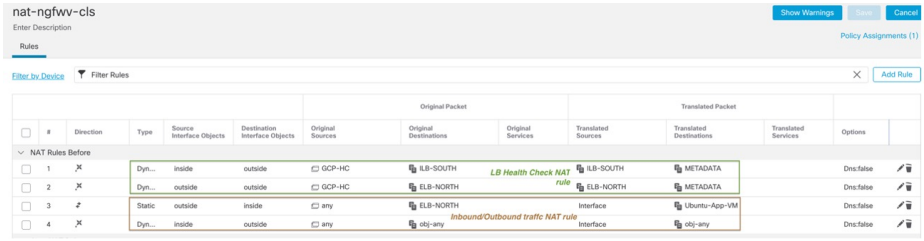
```
nat (inside,outside) source dynamic GCP-HC ILB-SOUTH destination static ILB-SOUTH METADATA
nat (outside,outside) source dynamic GCP-HC ELB-NORTH destination static ELB-NORTH
METADATA

nat (outside,inside) source static any interface destination static ELB-NORTH Ubuntu-App-VM
nat (inside,outside) source dynamic any interface destination static obj-any obj-any

object network Metadata
 host 169.254.169.254

object network ILB-SOUTH
host <ILB_IP>
object network ELB-NORTH
host <ELB_IP>

object-group network GCP-HC
 network-object 35.191.0.0 255.255.0.0
 network-object 130.211.0.0 255.255.252.0
 network-object 209.85.204.0 255.255.252.0
 network-object 209.85.152.0 255.255.252.0
```



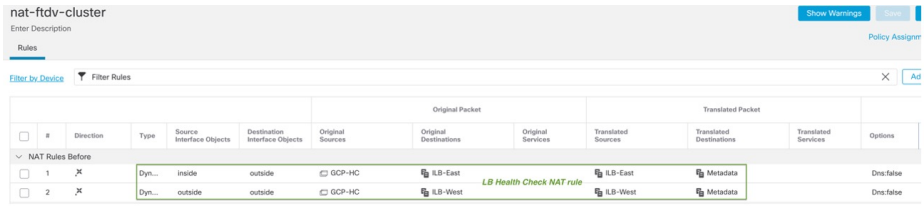
East-West NAT ルールの設定例

```
nat (inside,outside) source dynamic GCP-HC ILB-East destination static ILB-East Metadata
nat (outside,outside) source dynamic GCP-HC ILB-West destination static ILB-West Metadata

object network Metadata
 host 169.254.169.254

object network ILB-East
host <ILB_East_IP>
object network ILB-West
host <ILB_West_IP>

object-group network GCP-HC
 network-object 35.191.0.0 255.255.0.0
 network-object 130.211.0.0 255.255.252.0
 network-object 209.85.204.0 255.255.252.0
 network-object 209.85.152.0 255.255.252.0
```



ノースサウスおよびイーストウェスト トラフィック ルーティングの設定例

```
route outside 0.0.0.0 0.0.0.0 <Outside_Gateway> 1
route inside 35.191.0.0 255.255.0.0 <Inside_Gateway> 1
route inside 130.211.0.0 255.255.252.0 <Inside_Gateway> 1
route inside 209.85.152.0 255.255.252.0 <Inside_Gateway> 1
route inside 209.85.204.0 255.255.252.0 <Inside_Gateway> 1
```

デフォルト ルートが使用できない場合、ヘルス チェック用のトラフィックのルーティングにポリシーベース ルーティングを使用できます。

Management Center へのクラスタの追加（手動展開）

クラスタを手動で展開した場合は、この手順を使用してクラスタを Firewall Management Center に追加します。テンプレートを使用した場合、クラスタは自動的に Firewall Management Center に登録されます。

クラスタ ユニットのいずれかを新しいデバイスとして Firewall Management Center に追加します。Firewall Management Center は、他のすべてのクラスタ メンバーを自動検出します。

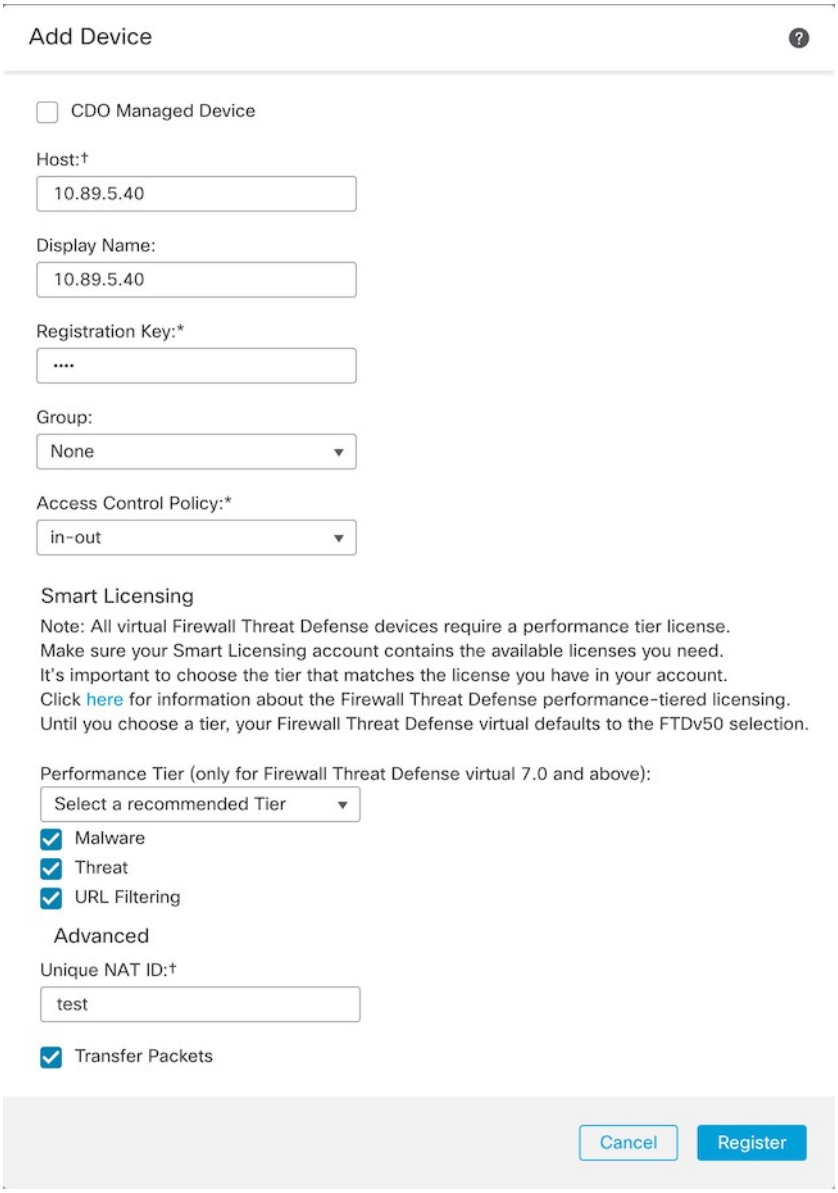
始める前に

- すべてのクラスタユニットは、Firewall Management Center に追加する前に、正常な形式のクラスタ内に存在している必要があります。また、どのユニットが制御ユニットかを確認することも必要です。Firewall Threat Defense **show cluster info** コマンドを使用します。

手順

ステップ 1 Firewall Management Center で、**[デバイス（Devices）]** > **[デバイス管理（Device Management）]** を選択してから、**[追加（Add）]** > **[デバイスの追加（Add Device）]** を選択し、制御ユニットの管理 IP アドレスを使用して制御ユニットを追加します。

図 *15* : デバイスの追加



a) [ホスト（Host）] フィールドに、制御ユニットの IP アドレスまたはホスト名を入力します。

最適なパフォーマンスを得るため、制御ユニットの追加を推奨しますが、クラスタの任意のユニットを追加できます。

デバイスのセットアップ時に NAT ID を使用した場合は、このフィールドを入力する必要がない可能性があります。詳細については、「NAT 環境」を参照してください。

b) [表示名（Display Name）] フィールドに、Firewall Management Center での制御ユニットの表示名を入力します。

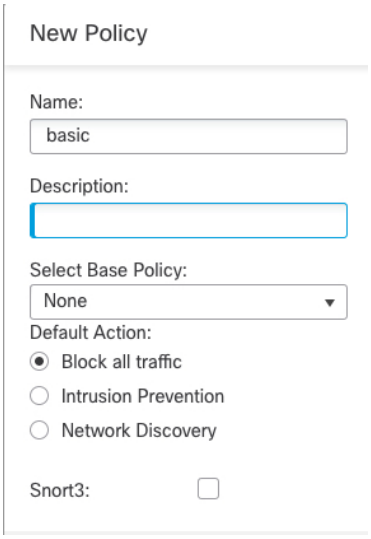
この表示名はクラスタ用ではありません。追加する制御ユニット専用です。後で、他のクラスタメンバーの名前やクラスタ表示名を変更できます。

c) [登録キー（Registration Key）] フィールドに、デバイスの設定時に使用したものと同じ登録キーを入力します。登録キーは、1 回限り使用可能な共有シークレットです。

d) （任意） デバイスをデバイス**グループ**に追加します。

e) 登録後すぐに、デバイスに展開する最初の [アクセス コントロール ポリシー（Access Control Policy）] を選択するか、新しいポリシーを作成します。

新しいポリシーを作成する場合は、基本ポリシーのみを作成します。必要に応じて、後でポリシーをカスタマイズできます。



f) デバイスに適用するライセンスを選択します。

g) デバイスの設定時に、NAT ID を使用した場合、[詳細（Advanced）] セクションを展開し、[一意の NAT ID（Unique NAT ID）] フィールドに同じ NAT ID を入力します。

h) [パケットの転送（Transfer Packets）] チェックボックスをオンにし、デバイスで Firewall Management Center にパケットを転送することを許可します。

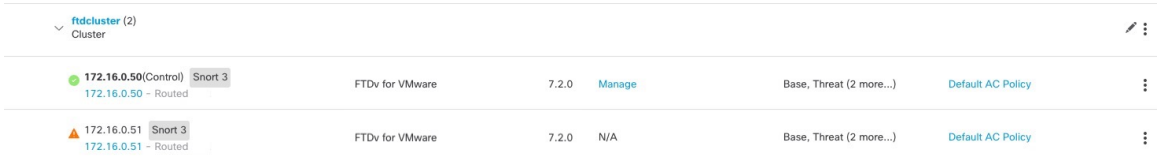
このオプションは、デフォルトで有効です。このオプションを有効にして IPS や Snort などのイベントがトリガーされた場合は、デバイスが検査用としてイベント メタデータ情報とパケット データを Firewall Management Center に送信します。このオプションを無効にした場合は、イベント情報だけが Firewall Management Center に送信され、パケット データは送信されません。

i) [登録（Register）] をクリックします。

Firewall Management Center は、制御ユニットを識別して登録した後に、すべてのデータユニットを登録します。制御ユニットが正常に登録されていない場合、クラスタは追加されません。クラスタが稼働状態になかった場合や、接続問題などが原因で、登録エラーが発生する場合があります。こうした状況では、クラスタ ユニットを再度追加することをお勧めします。

[デバイス（Devices）] > [デバイス管理（Device Management）] ページにクラスタ名が表示されます。クラスタを展開して、クラスタユニットを表示します。

図 *16*：クラスタの管理

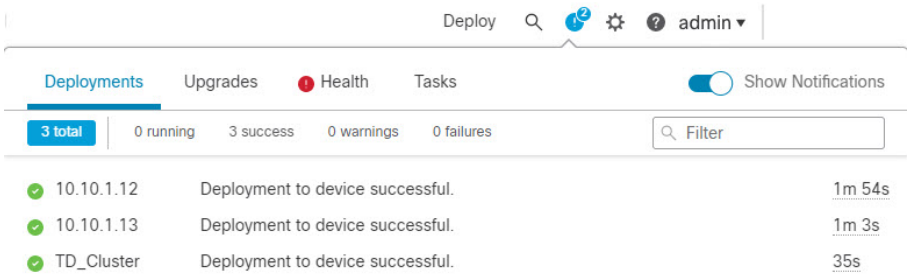


現在登録されているユニットには、ロード アイコンが表示されます。

図 *17*：ノードの登録

（注）
GCP は、クラスタノードの検出中にパブリック IP アドレスを持つノードを優先します。Firewall Threat Defense Virtual クラスタがプライベート IP アドレスを使用して Management Center Virtual に登録されるようにするには、最初に Firewall Threat Defense Virtual クラスタノードでパブリック IP アドレスを無効にする必要があります。これにより、GCP ノードの検出が Management Center Virtual への登録ノードのプライベート IP アドレスを使用して続行されます。

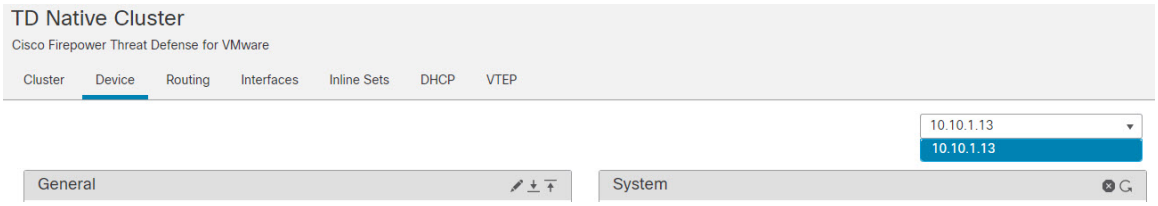
クラスタユニットの登録をモニターするには、[通知（Notifications）] アイコンをクリックし、[タスク（Tasks）] を選択します。Firewall Management Center は、ユニットの登録ごとにクラスタ登録タスクを更新します。いずれかのユニットの登録に失敗した場合には、クラスタノードの照合 （74 ページ） を参照してください。



ステップ 2 クラスタの [編集（Edit）]（✎） をクリックして、デバイス固有の設定を指定します。

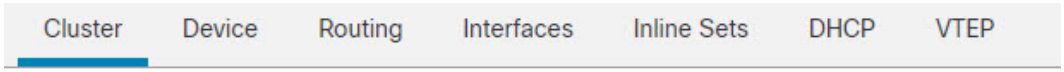
ほとんどの設定は、クラスタ内のノードではなく、クラスタ全体に適用できます。たとえば、ノードごとに表示名を変更できますが、インターフェイスはクラスタ全体についてのみ設定できます。

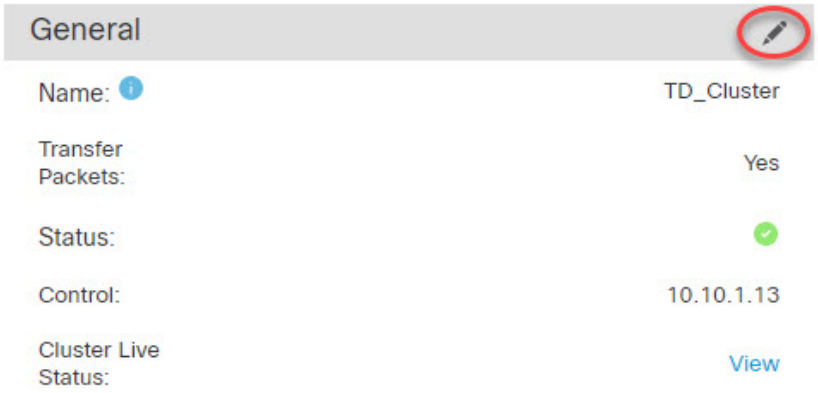
ステップ 3 **[デバイス（Devices）]** > **[デバイス管理（Device Management）]** > **[クラスタ（Cluster）]** 画面に、[全般（General）]、[ライセンス（License）]、[システム（System）]、および [ヘルス（Health）] の設定が表示されます。



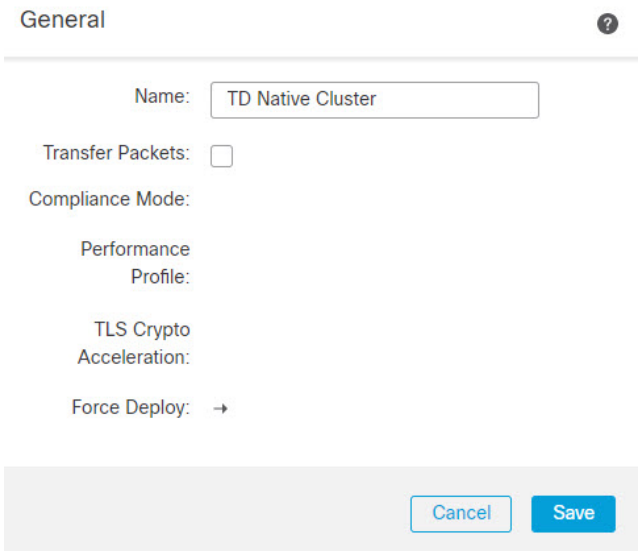
次のクラスタ固有の項目を参照してください。

- [全般（General）] > [名前（Name）] : [編集（Edit）]（✎） をクリックして、クラスタの表示名を変更します。

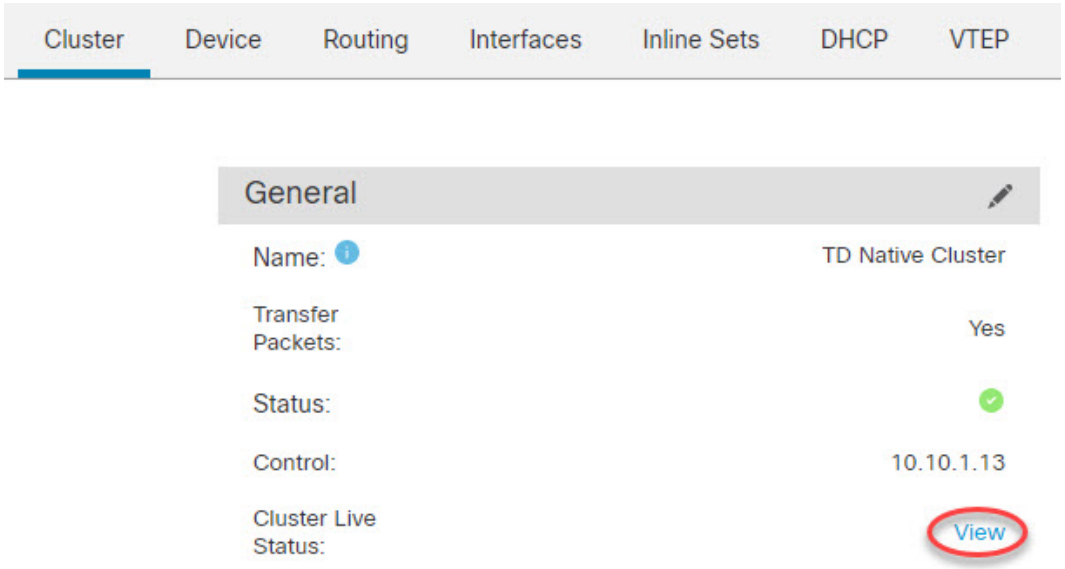




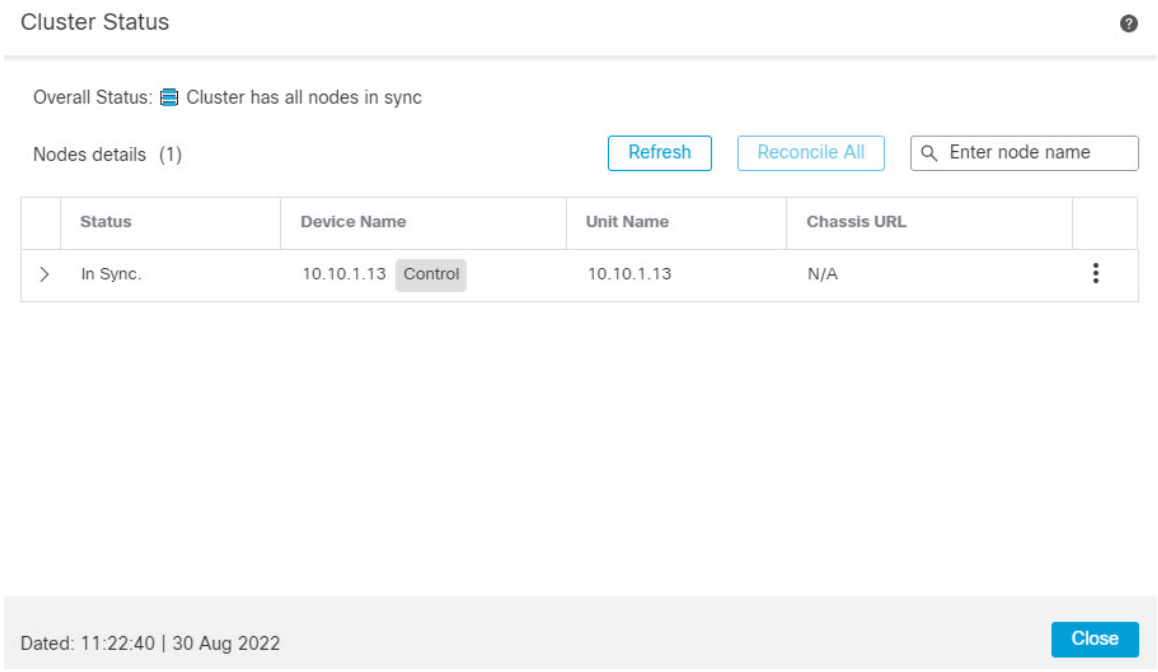
その後に、[名前（Name）] フィールドを設定します。



- [全般（General）] > [クラスタステータスの表示（View cluster status）] : [クラスタステータスの表示（View cluster status）] リンクをクリックして [クラスタステータス（Cluster Status）] ダイアログボックスを開きます。



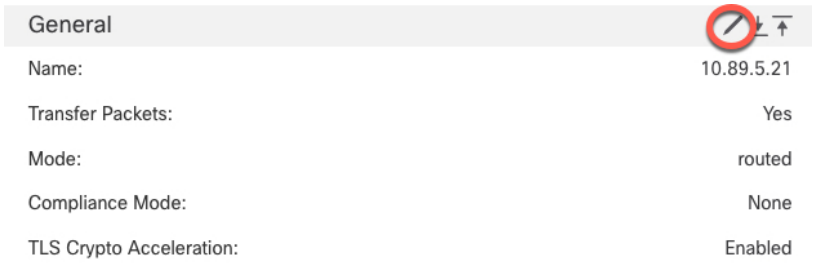
[クラスタステータス（Cluster Status）] ダイアログボックスで、[照合（Reconcile）] をクリックしてデータユニットの登録を再試行することもできます。



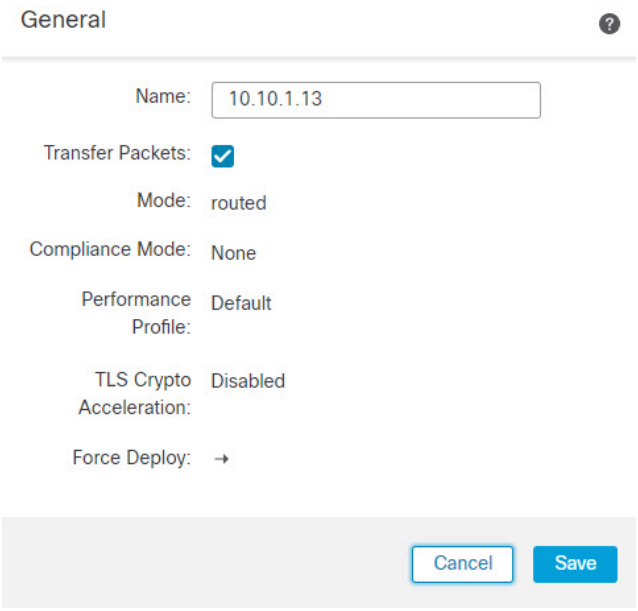
- [ライセンス（License）]：[編集（Edit）]（✎）をクリックして、ライセンス付与資格を設定します。

ステップ 4 **[デバイス（Devices）]** > **[デバイス管理（Device Management）]** > **[デバイス（Devices）]** の右上のドロップダウンメニューで、クラスタ内の各メンバーを選択し、次の設定を指定することができます。

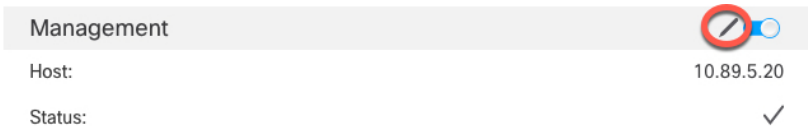
- [全般（General）] > [名前（Name）]：[編集（Edit）]（✎）をクリックして、クラスタメンバーの表示名を変更します。



その後に、[名前（Name）] フィールドを設定します。



- [管理（Management）]>[ホスト（Host）]：デバイス設定で管理 IP アドレスを変更する場合、Firewall Management Center で新しいアドレスを一致させてネットワーク上のデバイスに到達できるようにし、[管理（Management）]領域で[ホスト（Host）]アドレスを編集します。



クラスタのヘルスモニターの設定

[クラスタ（Cluster）]ページの[クラスタヘルスモニターの設定（Cluster Health Monitor Settings）]セクションには、次の表で説明されている設定が表示されます。

図 *18 :* クラスタのヘルスモニターの設定

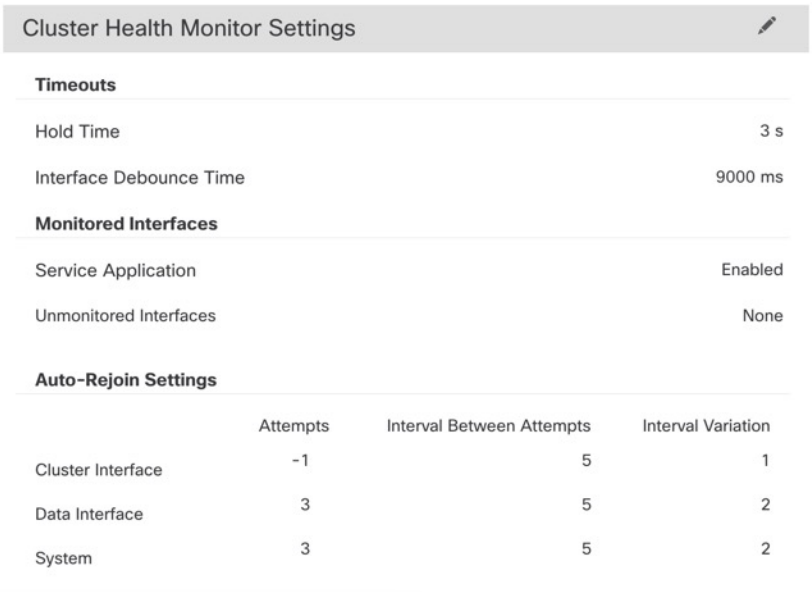


表 *5 : [*クラスタヘルスモニターの設定（***Cluster Health Monitor Settings***）*]* セクションテーブルのフィールド

フィールド	説明
タイムアウト（Timeouts）	
保留時間（Hold Time）	指定できる範囲は 0.3 ～ 45 秒です。デフォルトは 3 秒です。ノードの状態を確認するため、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。
インターフェイスのデバウンス時間（Interface Debounce Time）	指定できる範囲は 300 ～ 9000 ミリ秒です。デフォルトは 500 ms です。インターフェイスのデバウンス時間は、インターフェイスで障害が発生していると見なされ、クラスタからノードが削除されるまでの時間です。
Monitored Interfaces（モニタリング対象インターフェイス）	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。特定の論理インターフェイスのすべての物理ポートが、特定のノード上では障害が発生したが、別のノード上の同じ論理インターフェイスでアクティブポートがある場合、そのノードはクラスタから削除されます。ノードがメンバーをクラスタから削除するまでの時間は、インターフェイスのタイプと、そのノードが確立済みノードであるか、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。

フィールド	説明
サービスアプリケーション（Service Application）	Snort プロセスおよび disk-full プロセスが監視されているかどうかを示します。
モニタリング対象外のインターフェイス（Unmonitored Interfaces）	モニタリング対象外のインターフェイスを表示します。
自動再結合の設定（Auto-Rejoin Settings）	
クラスタインターフェイス（Cluster Interface）	クラスタ制御リンクに障害が発生した後に自動再結合の設定を表示します。
試行（*Attempts*）	指定できる範囲は -1 ～ 65535 です。デフォルトは -1（無制限）です。再結合の試行回数を設定します。
試行の間隔（*Interval Between Attempts*）	指定できる範囲は 2 ～ 60 です。デフォルトは 5 分です。再結合試行の間隔を分単位で定義します。
間隔のバリエーション（*Interval Variation*）	指定できる範囲は 1 ～ 3 です。デフォルトは間隔の 1 倍です。試行ごとに間隔を長くするかどうかを定義します。
データインターフェイス（Data Interfaces）	データインターフェイスに障害が発生した後に自動再結合の設定を表示します。
試行（*Attempts*）	指定できる範囲は -1 ～ 65535 です。デフォルトは 3 です。再結合の試行回数を設定します。
試行の間隔（*Interval Between Attempts*）	指定できる範囲は 2 ～ 60 です。デフォルトは 5 分です。再結合試行の間隔を分単位で定義します。
間隔のバリエーション（*Interval Variation*）	指定できる範囲は 1 ～ 3 です。デフォルトは間隔の 2 倍です。試行ごとに間隔を長くするかどうかを定義します。
システム（System）	内部エラーが発生した後に自動再結合の設定を表示します。内部エラーには、アプリケーション同期のタイムアウト、一貫性のないアプリケーションステータスなどがあります。
試行（*Attempts*）	指定できる範囲は -1 ～ 65535 です。デフォルトは 3 です。再結合の試行回数を設定します。
試行の間隔（*Interval Between Attempts*）	指定できる範囲は 2 ～ 60 です。デフォルトは 5 分です。再結合試行の間隔を分単位で定義します。
間隔のバリエーション（*Interval Variation*）	指定できる範囲は 1 ～ 3 です。デフォルトは間隔の 2 倍です。試行ごとに間隔を長くするかどうかを定義します。

（注） システムのヘルスチェックを無効にすると、システムのヘルスチェックが無効化されている場合に適用されないフィールドは表示されません。

これらの設定は、このセクションから変更できます。

任意のポートチャネル ID、単一の物理インターフェイス ID、Snort プロセス、および disk-full プロセスを監視できます。ヘルス モニタリングは VLAN サブインターフェイス、または VNI やBVI などの仮想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されています。

手順

ステップ 1 **[デバイス（Devices）]** > **[デバイス管理（Device Management）]** を選択します。

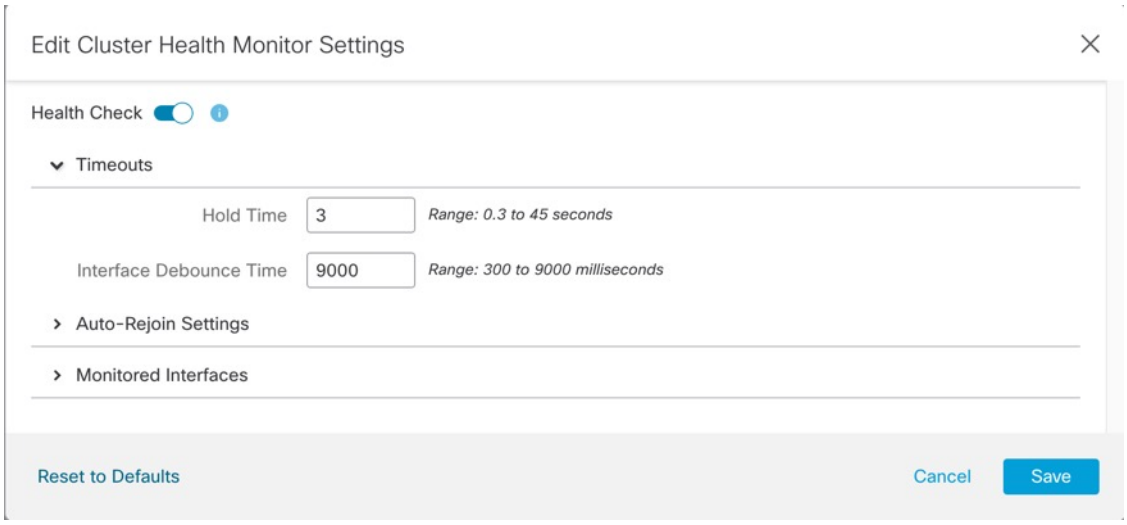
ステップ 2 変更するクラスタの横にある [編集（Edit）]（✎） をクリックします。

ステップ 3 [クラスタ（Cluster）] をクリックします。

ステップ 4 [クラスタのヘルスモニターの設定（Cluster Health Monitor Settings）] セクションで、[編集（Edit）]（✎） をクリックします。

ステップ 5 [ヘルスチェック（Health Check）] スライダをクリックして、システムのヘルスチェックを無効にします。

図 *19 :* システムヘルスチェックの無効化



何らかのトポロジ変更（たとえばデータインターフェイスの追加/削除、ノードやスイッチのインターフェイスの有効化/無効化、VSS や vPC（または VNet）を形成するスイッチの追加）を行うときには、システムの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モニタリングも無効にしてください。トポロジの変更が完了して、設定の変更がすべてのノー

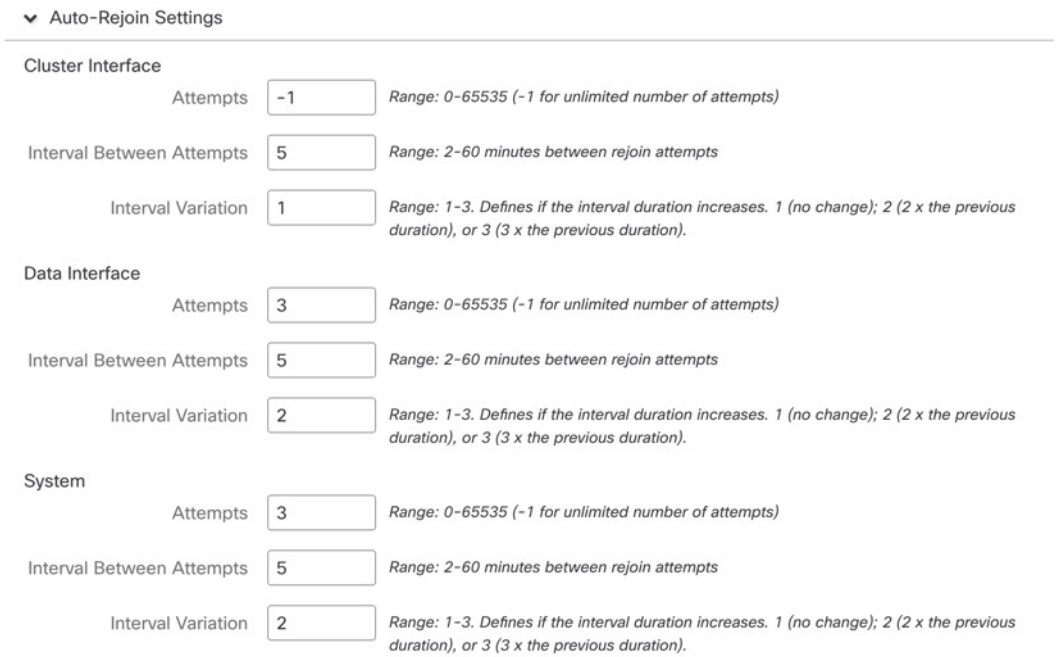
ドに同期されたら、システムのヘルスチェック機能を再度有効にてインターフェイスをモニタリングできます。

ステップ 6 ホールド時間とインターフェイスのデバウンス時間を設定します。

- [ホールド時間（Hold Time）]：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。
- [インターフェイスのデバウンス時間（Interface Debounce Time）]：デバウンス時間は 300 ～ 9000 ms の範囲で値を設定します。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、インターフェイス障害としてマーク付けされるまで、ノードは指定されたミリ秒数待機します。その後、ノードはクラスタから削除されます。EtherChannel がダウン状態からアップ状態に移行する場合（スイッチがリロードされた、スイッチで EtherChannel が有効になったなど）、デバウンス時間がより長くなり、ポートのバンドルにおいて別のクラスタノードの方が高速なため、クラスタノードでインターフェイスの障害が表示されることを妨げることがあります。

ステップ 7 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。

図 *20*：自動再結合の設定

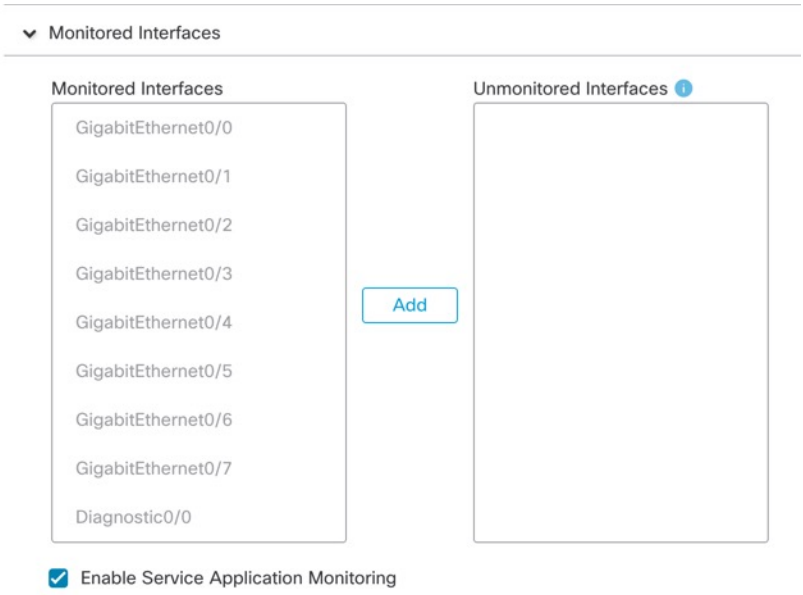


[クラスタインターフェイス（Cluster Interface）]、[データインターフェイス（Data Interface）]、および[システム（System）]に次の値を設定します（内部エラーには、アプリケーションの同期タイムアウト、一貫性のないアプリケーションステータスなどがあります）。

- [試行数（Attempts）]：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。[クラスタインターフェイス（Cluster Interface）] のデフォルト値は -1（無制限）です。[データインターフェイス（Data Interface）] と [システム（System）] のデフォルト値は 3 です。
- [試行の間隔（Interval Between Attempts）]：再結合試行の間隔を 2 ～ 60 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。
- [間隔のバリエーション（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（Cluster Interface）] の場合は **1**、[データインターフェイス（Data Interface）] および [システム（System）] の場合は **2** です。

ステップ 8 [モニタリング対象のインターフェイス（Monitored Interfaces）] または [モニタリング対象外のインターフェイス（Unmonitored Interfaces）] ウィンドウでインターフェイスを移動して、モニタリング対象のインターフェイスを設定します。[サービスアプリケーションのモニタリングを有効にする（Enable Service Application Monitoring）] をオンまたはオフにして、Snort プロセスと disk-full プロセスのモニタリングを有効または無効にすることもできます。

図 *21* : モニタリング対象インターフェイスの設定



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。特定の論理インターフェイスのすべての物理ポートが、特定のノード上では障害が発生したが、別のノード上の同じ論理インターフェイスでアクティブポートがある場合、そのノードはクラスタから削除されます。ノードがメンバーをクラスタから削除するまでの時間は、インターフェイスのタイプと、そのノードが確立済みノードであるか、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、および Snort プロセスと disk-full プロセスで有効になっています。

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無効にできます。

何らかのトポロジ変更（たとえばデータインターフェイスの追加/削除、ノードやスイッチのインターフェイスの有効化/無効化、VSS や vPC（または VNet）を形成するスイッチの追加）を行うときには、システムの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モニタリングも無効にしてください。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、システムのヘルスチェック機能を再度有効にてインターフェイスをモニタリングできます。

ステップ 9 [保存（Save）] をクリックします。

ステップ 10 設定変更を展開します設定変更の展開を参照してください。

クラスタノードの管理

クラスタリングを無効にする

ノードの削除に備えて、またはメンテナンスのために一時的にノードを非アクティブ化する場合があります。この手順は、ノードを一時的に非アクティブ化するためのものです。ノードは引き続き Firewall Management Center のデバイスリストに表示されます。ノード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。

（注） クラスタリングを無効にせずにノードの電源を切らないでください。

手順

ステップ 1 無効にするユニットに対して、[デバイス（Devices）]>[デバイス管理（Device Management）] の順に選択して [その他（More）] (⋮) をクリックし、[ノードのクラスタリングを無効にする（Disable Node Clustering）] を選択します。

ステップ 2 ノードのクラスタリングを無効にすることを確認します。

ノードは、[デバイス（Devices）]>[デバイス管理（Device Management）] リストの名前の横に [（無効（Disabled））] と表示されます。

ステップ 3 クラスタリングを再び有効にするには、クラスタへの再参加 （74 ページ）を参照してください。

クラスタへの再参加

（たとえば、インターフェイスで障害が発生したために）ノードがクラスタから削除された場合、または手動でクラスタリングを無効にした場合は、クラスタに手動で再参加する必要があります。クラスタへの再参加を試行する前に、障害が解決されていることを確認します。ノードをクラスタから削除できる理由の詳細については、「クラスタへの再参加 （92 ページ）」を参照してください。

手順

ステップ 1 再度有効にするユニットに対して、[デバイス（Devices）] > [デバイス管理（Device Management）] の順に選択して [その他（More）] をクリックし、[ノードのクラスタリングを有効にする（Enable Node Clustering）] を選択します。 >

ステップ 2 ノードのクラスタリングを有効にすることを確認します。

クラスタノードの照合

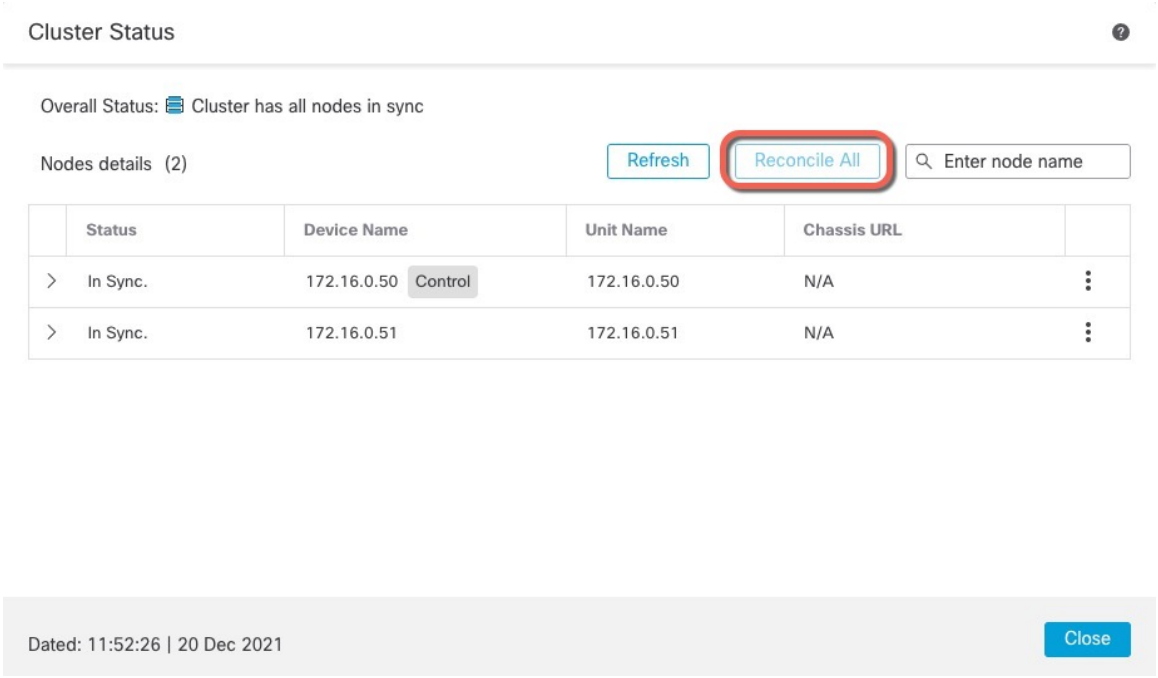
クラスタノードの登録に失敗した場合は、デバイスから Firewall Management Center に対してクラスタメンバーシップを照合できます。たとえば、Firewall Management Center が特定のプロセスで占領されているか、ネットワークに問題がある場合、データノードの登録に失敗することがあります。

手順

ステップ 1 クラスタの **[Devices]** > **[Device Management]** > [その他（More）] を選択し、次に [Cluster Live Status] を選択して [Cluster Status] ダイアログボックスを開きます。

ステップ 2 [すべてを照合（Reconcile All）] をクリックします。

図 *22 :* すべてを照合



クラスタ ステータスの詳細については、クラスタのモニタリング （76 ページ）を参照してください。

クラスタまたはノードの削除（登録解除）と新しいFirewaManagement Center への登録

Firewall Management Center からクラスタを登録解除できます。これにより、クラスタはそのまま維持されます。クラスタを新しい Firewall Management Center に追加する場合は、クラスタを登録解除することができます。

クラスタからノードを除外することなく、Firewall Management Center からノードを登録解除することもできます。ノードは Firewall Management Center に表示されていませんが、まだクラスタの一部であり、引き続きトラフィックを渡して制御ノードになることも可能です。現在動作している制御ノードを登録解除することはできません。Firewall Management Center から到達不可能になったノードは登録解除してもかまいませんが、管理接続をトラブルシューティングする間、クラスタの一部として残しておくことも可能です。

クラスタの登録解除：

- Firewall Management Center とクラスタとの間のすべての通信が切断されます。
- [デバイス管理（Device Management）] ページからクラスタが削除されます。

- クラスタのプラットフォーム設定ポリシーで、NTP を使用して Firewall Management Center から時間を受信するように設定されている場合は、クラスタがローカル時間管理に戻されます。
- 設定はそのままになるため、クラスタはトラフィックの処理を続行します。

  NAT や VPN などのポリシー、ACL、およびインターフェイス構成は維持されます。

同じまたは別の Firewall Management Center にクラスタを再登録すると、設定が削除されるため、クラスタはその時点でトラフィックの処理を停止します。クラスタ設定はそのまま維持されるため、クラスタ全体を追加できます。登録時にアクセス コントロール ポリシーを選択できますが、トラフィックを再度処理する前に、登録後に他のポリシーを再適用してから設定を展開する必要があります。

始める前に

この手順では、いずれかのノードへの CLI アクセスが必要です。

手順

ステップ 1 [デバイス（Devices）]>[デバイス管理（Device Management）]の順に選択し、クラスタかノードの [その他（More）] (⋮) をクリックして [登録解除] [削除（Delete）] を選択します。 >

ステップ 2 クラスタかノードを削除するよう求められたら、[はい（Yes）] をクリックします。

ステップ 3 クラスタメンバーの 1 つを新しいデバイスとして追加することにより、クラスタを新しい（または同じ）Firewall Management Center に登録できます。

クラスタノードの 1 つをデバイスとして追加するだけで、残りのクラスタノードが検出されます。

a) 1 つのクラスタノードの CLI に接続し、**configure manager add** コマンドを使用して新しい Firewall Management Center を識別します。

b) **[デバイス（Devices）] > [デバイス管理（Device Management）]** を選択し、[デバイスの追加（Add Device）] をクリックします。

ステップ 4 削除したノードを再度追加する方法については、「クラスタノードの照合 （74 ページ）」を参照してください。

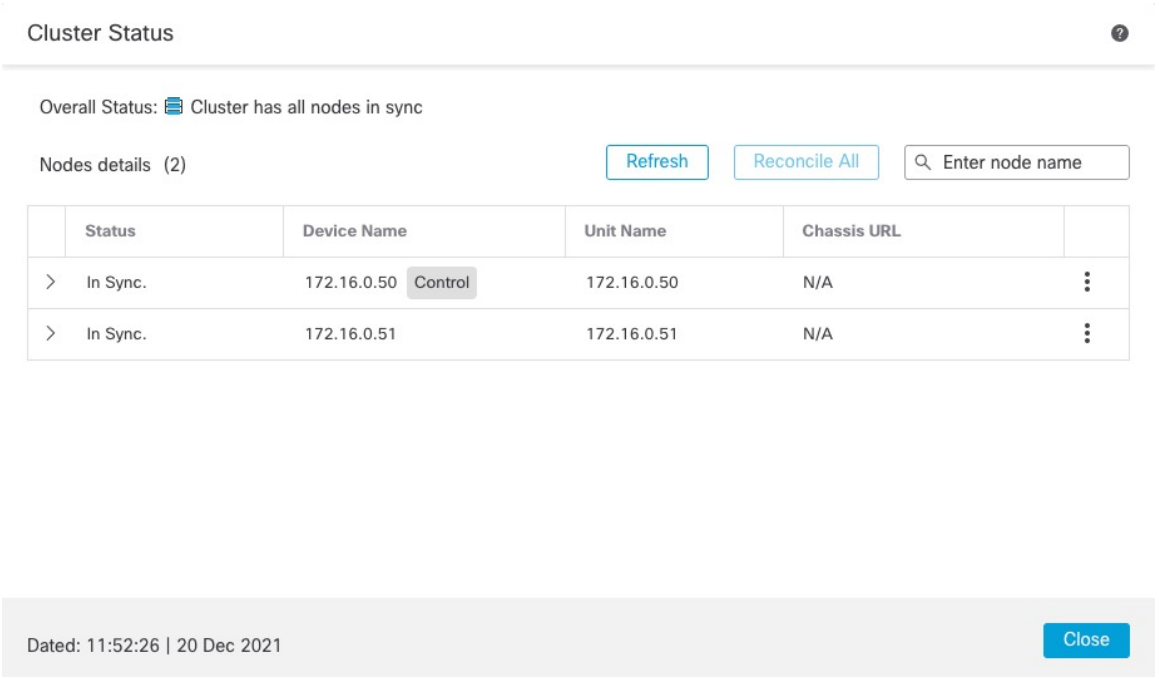
クラスタのモニタリング

クラスタは、Firewall Management Center と Firewall Threat Defense の CLI でモニターできます。

- [クラスタステータス（Cluster Status）] ダイアログボックスには、[デバイス（Devices）] > [デバイス管理（Device Management）] > [その他（More）] (⋮) アイコンから、または [デバイス（Devices）]>[デバイス管理（Device Management）]>[クラスタ（Cluster）] ペー

ジ > [全般（General）] 領域 > [クラスタのライブステータス（Cluster Live Status）] リンクからアクセスできます。 > > >

図 *23 :* クラスタのステータス



制御ノードには、そのロールを示すグラフィックインジケータがあります。

クラスタメンバーステータスには、次の状態が含まれます。

- 同期中（In Sync）：ノードは Firewall Management Center に登録されています。
- 登録の保留中（Pending Registration）：ノードはクラスタの一部ですが、まだ Firewall Management Center に登録されていません。ノードの登録に失敗した場合は、[すべてを照合（Reconcile All）] をクリックして登録を再試行できます。
- クラスタリングが無効（Clustering is disabled）：ノードは Firewall Management Center に登録されていますが、クラスタの非アクティブなメンバーです。クラスタリング設定は、後で再有効化する予定がある場合は変更せずに維持できます。また、ノードをクラスタから削除することも可能です。
- クラスタに参加中...（Joining cluster...）：ノードがシャーシ上でクラスタに参加していますが、参加は完了していません。参加後に Firewall Management Center に登録されます。

ノードごとに [概要（Summary）] と [履歴（History）] を表示できます。

図 *24 : ノードの [*概要（***Summary***）*]*

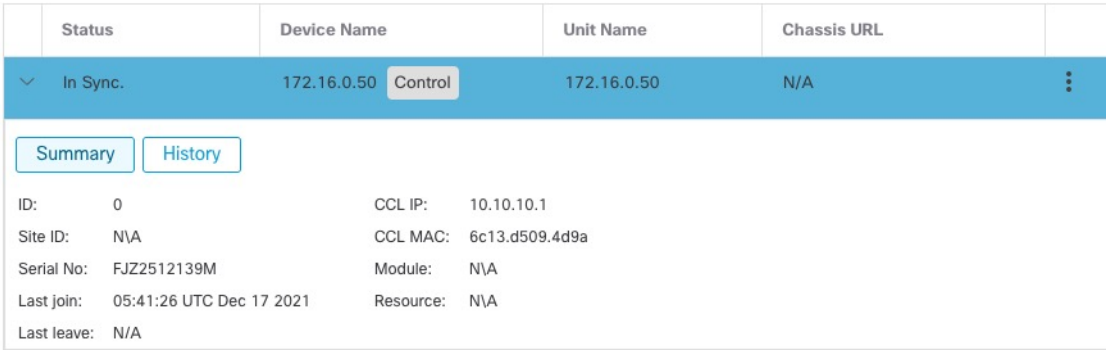
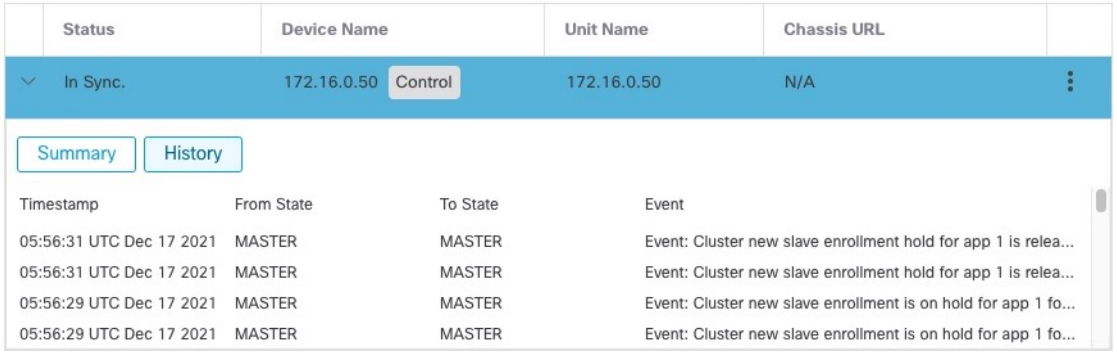


図 *25 : ノードの [*履歴（***History***）*]*



- [システム（System）]（⚙）> [Tasks] ページ。

  [タスク（Tasks）]ページには、ノードが登録されるたびにクラスタ登録タスクの最新情報が表示されます。

- [デバイス（Devices）] > [デバイス管理（Device Management）] > cluster_name。 >

  デバイスの一覧表示ページでクラスタを展開すると、IP アドレスの横にそのロールが表示されている制御ノードを含む、すべてのメンバーノードを表示できます。登録中のノードには、ロード中のアイコンが表示されます。

- **show cluster** {**access-list** [*acl_name*] | **conn** [**count**] | **cpu** [**usage**] | **history** | **interface-mode** | **memory** | **resource usage** | **service-policy** | **traffic** | **xlate count**}

  クラスタ全体の集約データまたはその他の情報を表示するには、**show cluster** コマンドを使用します。

- **show cluster info** [**auto-join** | **clients** | **conn-distribution** | **flow-mobility counters** | **goid** [*options*] | **health** | **incompatible-config** | **loadbalance** | **old-members** | **packet-distribution** | **trace** [*options*] | **transport** { **asp** | **cp**}]

  クラスタ情報を表示するには、**show cluster info** コマンドを使用します。

クラスタ ヘルス モニター ダッシュボード

クラスタのヘルスモニター

Firewall Threat Defense がクラスタの制御ノードである場合、Firewall Management Center はデバイス メトリック データ コレクタからさまざまなメトリックを定期的に収集します。クラスタの ヘルスモニターは、次のコンポーネントで構成されています。

- 概要ダッシュボード：クラスタトポロジ、クラスタ統計、およびメトリックチャートに関する情報を表示します。
  - トポロジセクションには、クラスタのライブステータス、個々の脅威防御の状態、脅威防御ノードのタイプ（制御ノードまたはデータノード）、およびデバイスの状態が表示されます。デバイスの状態は、[無効（Disabled）]（デバイスがクラスタを離れたとき）、[初期状態で追加（Added out of box）]（パブリッククラウドクラスタで Firewall Management Center に属していない追加ノード）、または [標準（Normal）]（ノードの理想的な状態）のいずれかです。
  - クラスタの統計セクションには、CPU使用率、メモリ使用率、入力レート、出力レート、アクティブな接続数、および NAT 変換数に関するクラスタの現在のメトリックが表示されます。

（注） CPU とメモリのメトリックは、データプレーンと Snort の使用量の個々の平均を示します。

  - メトリックチャート、つまり、CPU使用率、メモリ使用率、スループット、および接続数は、指定された期間におけるクラスタの統計を図表で示します。
- 負荷分散ダッシュボード：2 つのウィジェットでクラスタノード全体の負荷分散を表示します。
  - 分布ウィジェットには、クラスタノード全体の時間範囲における平均パケットおよび接続分布が表示されます。このデータは、ノードによって負荷がどのように分散されているかを示します。このウィジェットを使用すると、負荷分散の異常を簡単に特定して修正できます。
  - ノード統計ウィジェットには、ノードレベルのメトリックが表形式で表示されます。クラスタノード全体の CPU 使用率、メモリ使用率、入力レート、出力レート、アクティブな接続数、および NAT 変換数に関するメトリックデータが表示されます。このテーブルビューでは、データを関連付けて、不一致を簡単に特定できます。
- メンバー パフォーマンス ダッシュボード：クラスタノードの現在のメトリックを表示します。セレクタを使用してノードをフィルタリングし、特定ノードの詳細を表示できます。メトリックデータには、CPU使用率、メモリ使用率、入力レート、出力レート、アクティブな接続数、および NAT 変換数が含まれます。

- CCL ダッシュボード：クラスタの制御リンクデータ、つまり入力レートと出力レートをグラフ形式で表示します。
- トラブルシューティングとリンク：頻繁に使用されるトラブルシューティングのトピックと手順への便利なリンクを提供します。
- 時間範囲：さまざまなクラスタ メトリック ダッシュボードやウィジェットに表示される情報を制限するための調整可能な時間枠。
- カスタムダッシュボード：クラスタ全体のメトリックとノードレベルのメトリックの両方に関するデータを表示します。ただし、ノードの選択は脅威防御メトリックにのみ適用され、ノードが属するクラスタ全体には適用されません。

クラスタ ヘルスの表示

この手順を実行するには、管理者ユーザー、メンテナンスユーザー、またはセキュリティアナリスト ユーザーである必要があります。

クラスタヘルスモニターは、クラスタとそのノードのヘルスステータスの詳細なビューを提供します。このクラスタヘルスモニターは、一連のダッシュボードでクラスタのヘルスステータスと傾向を提供します。

始める前に

- Firewall Management Center の 1 つ以上のデバイスからクラスタを作成しているかを確認します。

手順

ステップ 1 [システム（System）]（⚙）> **[正常性（Health）]** > **[モニタ（Monitor）]** を選択します。

[モニタリング（Monitoring）] ナビゲーションウィンドウを使用して、ノード固有のヘルスモニターにアクセスします。

ステップ 2 デバイスリストで [展開（Expand）]（>）と [折りたたみ（Collapse）]（⌄）をクリックして、管理対象のクラスタデバイスのリストを展開または折りたたみます。

ステップ 3 クラスタのヘルス統計を表示するには、クラスタ名をクリックします。デフォルトでは、クラスタモニターは、いくつかの事前定義されたダッシュボードで正常性およびパフォーマンスのメトリックを報告します。メトリックダッシュボードには次のものが含まれます。

- [概要（Overview）]：他の事前定義されたダッシュボードからの主要なメトリックを表示します。ノード、CPU、メモリ、入力レート、出力レート、接続統計情報、NAT 変換情報などが含まれます。
- [負荷分散（Load Distribution）]：クラスタノード間のトラフィックとパケットの分散。

- [メンバーパフォーマンス（Member Performance）]：CPU 使用率、メモリ使用率、入力スループット、出力スループット、アクティブな接続、および NAT 変換に関するノードレベルの統計情報。
- [CCL]：インターフェイスのステータスおよび集約トラフィックの統計情報。

ラベルをクリックすると、さまざまなメトリックダッシュボードに移動できます。サポートされているクラスタメトリックの包括的なリストについては、「Cisco Secure Firewall Threat Defense Health Metrics」を参照してください。

ステップ 4 右上隅のドロップダウンで、時間範囲を設定できます。最短で 1 時間前（デフォルト）から、最長では 2 週間前からの期間を反映できます。ドロップダウンから [Custom] を選択して、カスタムの開始日と終了日を設定します。

更新アイコンをクリックして、自動更新を 5 分に設定するか、自動更新をオフに切り替えます。

ステップ 5 選択した時間範囲について、トレンドグラフの展開オーバーレイの展開アイコンをクリックします。

展開アイコンは、選択した時間範囲内の展開数を示します。垂直の帯は、展開の開始時刻と終了時刻を示します。複数の展開の場合、複数の帯または線が表示されます。展開の詳細を表示するには、点線の上部にあるアイコンをクリックします。

ステップ 6 （ノード固有のヘルスモニターの場合）ページ上部のデバイス名の右側にあるアラート通知で、ノードの正常性アラートを確認します。

正常性アラートにポインタを合わせると、ノードの正常性の概要が表示されます。ポップアップウィンドウに、上位 5 つの正常性アラートの概要の一部が表示されます。ポップアップをクリックすると、正常性アラート概要の詳細ビューが開きます。

ステップ 7 （ノード固有のヘルスモニターの場合）デフォルトでは、デバイスモニターは、いくつかの事前定義されたダッシュボードで正常性およびパフォーマンスのメトリックを報告します。メトリックダッシュボードには次のものが含まれます。

- Overview：CPU、メモリ、インターフェイス、接続統計情報など、他の定義済みダッシュボードからの主要なメトリックを表示します。ディスク使用量と重要なプロセス情報も含まれます。
- CPU：CPU 使用率。プロセス別および物理コア別の CPU 使用率を含みます。
- Memory：デバイスのメモリ使用率。データプレーンと Snort のメモリ使用率を含みます。
- Interfaces：インターフェイスのステータスおよび集約トラフィック統計情報。
- Connections：接続統計（エレファントフロー、アクティブな接続数、ピーク接続数など）および NAT 変換カウント。
- [Snort]：Snort プロセスに関連する統計情報。
- [ASP ドロップ（ASP drops）]：さまざまな理由でドロップされたパケットに関連する統計情報。

ラベルをクリックすると、さまざまなメトリックダッシュボードに移動できます。サポートされているデバイスメトリックの包括的なリストについては、「Cisco Secure Firewall Threat Defense Health Metrics」を参照してください。

ステップ 8 正常性モニターの右上隅にあるプラス記号[新しいダッシュボードの追加（Add New Dashboard）]（＋）をクリックして、使用可能なメトリックグループから独自の変数セットを構成し、カスタムダッシュボードを作成します。

クラスタ全体のダッシュボードの場合は、クラスタのメトリックグループを選択してから、メトリックを選択します。

クラスタメトリック

クラスタのヘルスモニターは、クラスタとそのノードに関連する統計情報と、負荷分散、パフォーマンス、および CCL トラフィックの統計データの集約結果を追跡します。

表 *6:* クラスタメトリック

メトリック	説明	フォーマット（Format）
CPU	クラスタノード上の CPU メトリックの平均（データプレーンと snort についてそれぞれ表示）。	パーセンテージ
メモリ	クラスタノード上のメモリメトリックの平均（データプレーンと snort についてそれぞれ表示）。	パーセンテージ
データスループット	クラスタの着信および発信データトラフィックの統計。	バイト
CCL スループット	クラスタの着信および発信 CCL トラフィックの統計。	バイト
接続	クラスタ内のアクティブな接続数。	番号
NAT 変換数	クラスタの NAT 変換数。	番号
分布	1 秒ごとのクラスタ内の接続分布数。	番号
パケット	クラスタ内の1秒ごとのパケット配信の件数。	番号

クラスタのアップグレード

Firewall Threat Defense Virtual クラスタをアップグレードするには、次の手順を実行します。

始める前に

- パブリッククラウドでクラスタをアップグレードする前に、ターゲットバージョンのイメージをクラウドイメージリポジトリにコピーし、クラスタ展開テンプレートのイメージ ID を更新します（実際には、既存のテンプレートを変更したコピーで置き換えることを推奨します）。これにより、アップグレード後に新しいインスタンス（クラスタスケーリング中に起動されたインスタンスなど）が正しいバージョンを使用するようになります。クラスタにパッチが適用されている場合など、必要なイメージがマーケットプレイスにない場合は、インスタンス固有（Day 0）の設定がなく、正しいバージョンを実行しているスタンドアロンの Firewall Threat Defense Virtual インスタンスのスナップショットからカスタムイメージを作成します。
- AWS 向け Firewall Threat Defense Virtual の場合、自動スケーリングされたクラスタのアップグレードの前に HealthCheck プロセスと ReplaceUnhealthy プロセスを一時停止します。これにより、アップグレード後の再起動中に Auto Scaling グループによってインスタンスが終了されなくなります。一時停止したプロセスは、後で再開できます。手順については、Amazon EC2 Auto Scaling のユーザーガイドの「Suspend and resume Amazon EC2 Auto Scaling processes」を参照してください。

手順

ステップ 1 ターゲット イメージ バージョンをクラウドイメージストレージにアップロードします。

ステップ 2 更新されたターゲット イメージ バージョンでクラスタのクラウド インスタンス テンプレートを更新します。

a) ターゲット イメージ バージョンを使用してインスタンステンプレートのコピーを作成します。

b) 新しく作成したテンプレートをクラスタ インスタンス グループにアタッチします。

ステップ 3 ターゲット イメージ バージョンのアップグレードパッケージを Firewall Management Center にアップロードします。

ステップ 4 アップグレードするクラスタで準備状況チェックを実行します。

ステップ 5 準備状況チェックが成功したら、アップグレードパッケージのインストールを開始します。

ステップ 6 Firewall Management Center は、クラスタノードを一度に 1 つずつアップグレードします。

ステップ 7 クラスタのアップグレードが成功すると、Firewall Management Center に通知が表示されます。

アップグレード後のインスタンスのシリアル番号と UUID に変更はありません。

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。

Threat Defense の機能とクラスタリング

Firewall Threat Defense の一部の機能はクラスタリングではサポートされず、一部は制御ユニット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。

サポートされていない機能とクラスタリング

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。

（注） クラスタリングでもサポートされていない FlexConfig 機能（WCCP インスペクションなど）を表示するには、ASA の一般的な操作のコンフィギュレーション ガイドを参照してください。FlexConfig では、Firewall Management Center GUI にはない多くの ASA 機能を設定できます。FlexConfig ポリシーを参照してください。

- リモート アクセス VPN（SSL VPN および IPsec VPN）
- 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。
- 仮想トンネルインターフェイス（VTI）
- 高可用性
- 統合ルーティングおよびブリッジング
- Firewall Management Center UCAPL/CC 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。

（注） クラスタリングでも一元化されている FlexConfig 機能（RADIUS インスペクションなど）を表示するには、ASA の一般的な操作のコンフィギュレーションガイドを参照してください。FlexConfig では、Firewall Management Center GUI にはない多くの ASA 機能を設定できます。FlexConfig ポリシーを参照してください。

- 次のアプリケーション インスペクション：
  - DCERPC
  - ESMTP
  - NetBIOS
  - PPTP
  - RSH
  - SQLNET
  - SUNRPC
  - TFTP
  - XDMCP
- スタティック ルート モニタリング

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。

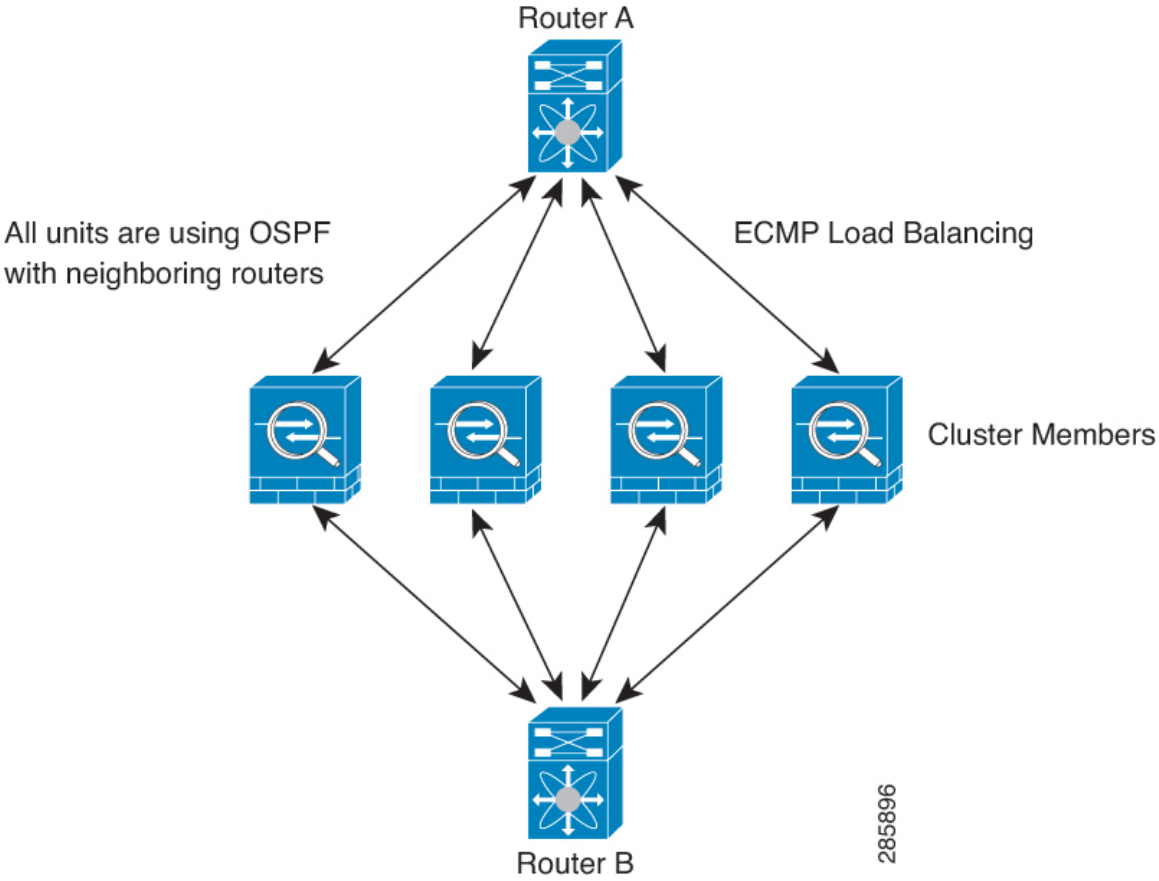
接続設定とクラスタリング

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

ダイナミック ルーティングおよびクラスタリング

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

図 *26 :* 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。

FTP とクラスタリング

- FTP データ 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 メンバよって所有されている場合は、データ 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。

NAT とクラスタリング

NAT の使用については、次の制限事項を参照してください。

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 Firewall Threat Defense 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 が使用されるときは、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 Firewall Threat Defense 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タティック 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。

- ポート ブロック割り当てによる PAT：この機能については、次のガイドラインを参照してください。

  - ホストあたりの最大制限は、クラスタ全体の制限ではなく、ノードごとに個別に適用されます。したがって、ホストあたりの最大制限が 1 に設定されている 3 ノードクラスタでは、ホストからのトラフィックが 3 つのノードすべてにロードバランシングされている場合、3 つのブロックを各ノードに 1 つずつ割り当てることができます。

  - バックアッププールからバックアップノードで作成されたポートブロックは、ホストあたりの最大制限の適用時には考慮されません。

  - PAT プールが完全に新しい IP アドレスの範囲で変更される On-the-fly PAT ルールの変更では、新しいプールが有効になっていてもいまだ送信中の xlate バックアップ要求に対する xlate バックアップの作成が失敗します。この動作はポートのブロック割り当て機能に固有なものではなく、プールが分散されトラフィックがクラスタノード間でロードバランシングされるクラスタ展開でのみ見られる一時的な PAT プールの問題です。

  - クラスタで動作している場合、ブロック割り当てサイズを変更することはできません。新しいサイズは、クラスタ内の各デバイスをリロードした後にのみ有効になります。各デバイスのリロードの必要性を回避するために、すべてのブロック割り当てルールを削除し、それらのルールに関連するすべての xlate をクリアすることをお勧めします。その後、ブロックサイズを変更し、ブロック割り当てルールを再作成できます。

- ダイナミック PAT の NAT プールアドレス配布：PAT プールを設定すると、クラスタはプール内の各 IP アドレスをポートブロックに分割します。デフォルトでは、各ブロック

は512ポートですが、ポートブロック割り当てルールを設定すると、代わりにユーザのブロック設定が使用されます。これらのブロックはクラスタ内のノード間で均等に分散されるため、各ノードには PAT プール内の IP アドレスごとに 1 つ以上のブロックがあります。したがって、想定される PAT 接続数に対して十分である場合には、クラスタの PAT プールに含める IP アドレスを 1 つだけにすることができます。PAT プールの NAT ルールで予約済みポート 1 ～ 1023 を含めるようにオプションを設定しない限り、ポートブロックは 1024 ～ 65535 のポート範囲をカバーします。

- 複数のルールにおける PAT プールの再利用：複数のルールで同じ PAT プールを使用するには、ルールにおけるインターフェイスの選択に注意を払う必要があります。すべてのルールで特定のインターフェイスを使用するか、あるいはすべてのルールで「任意の」インターフェイスを使用するか、いずれかを選択する必要があります。ルール全般にわたって特定のインターフェイスと「任意」のインターフェイスを混在させることはできません。混在させると、システムがリターントラフィックとクラスタ内の適切なノードを一致させることができなくなる場合があります。ルールごとに固有の PAT プールを使用することは、最も信頼性の高いオプションです。
- ラウンドロビンなし：PAT プールのラウンドロビンは、クラスタリングではサポートされません。
- 拡張 PAT なし：拡張 PAT はクラスタリングでサポートされません。
- 制御ノードによって管理されるダイナミック NAT xlate：制御ノードが xlate テーブルを維持し、データノードに複製します。ダイナミック NAT を必要とする接続をデータノードが受信したときに、その xlate がテーブル内にない場合、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 xlate：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。
- 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
  - RSH
  - SQLNET
  - TFTP
  - XDMCP
  - SIP
- 1 万を超える非常に多くの NAT ルールがある場合は、デバイスの CLI で **asp rule-engine transactional-commit nat** コマンドを使用してトランザクション コミット モデルを有効にする必要があります。有効にしないと、ノードがクラスタに参加できない可能性があります。

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。

SNMP とクラスタリング

SNMP エージェントは、個々の Firewall Threat Defense を、その [診断（Diagnostic）] 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ユーザは新しいノードに複製されません。SNMPv3ユーザは、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。ユーザを削除して再追加し、設定を再展開して、ユーザを新しいノードに強制的に複製する必要があります。

syslog とクラスタリング

- 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。

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。

ノードヘルスモニタリング

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。

すべての物理インターフェイスがモニタリングされます。ただし、モニタリングできるのは、名前付きインターフェイスのみです。ヘルスチェックは、インターフェイスごとに、モニターリングをオプションで無効にすることができます。

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。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、Firewall Threat Defense は自動的にクラスタへの再参加を試みます。

（注） Firewall Threat Defense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/診断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。

クラスタへの再参加

クラスタ メンバ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。

- 最初に参加するときに障害が発生したクラスタ制御リンク：クラスタ制御リンクの問題を解決した後、クラスタリングを再び有効にして、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：FTD は、無限に 5 分ごとに自動的に再参加を試みます。
- データ インターフェイスの障害：Firewall Threat Defense 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、Firewall Threat Defense アプリケーションはクラスタリングを無効にします。データ インターフェイスの問題を解決した後、手動でクラスタリングを有効にする必要があります。
- 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る限り、ノードは再起動するとクラスタに再参加します。Firewall Threat Defense アプリケーションは 5 秒ごとにクラスタへの再参加を試みます。
- 内部エラー：内部エラーには、アプリケーション同期のタイムアウト、一貫性のないアプリケーション ステータスなどがあります。
- 障害が発生した設定の展開：FMC から新しい設定を展開し、展開が一部のクラスタメンバーでは失敗したものの、他のメンバーでは成功した場合、失敗したノードはクラスタから削除されます。クラスタリングを再度有効にして手動でクラスタに再参加する必要があります。制御ノードで展開が失敗した場合、展開はロールバックされ、メンバーは削除されません。すべてのデータノードで展開が失敗した場合、展開はロールバックされ、メンバーは削除されません。

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

Traffic	状態のサポート	注意
Up time	Yes	システムアップタイムをトラッキングします。
ARP Table	あり	—
MAC アドレス テーブル	あり	—
ユーザ アイデンティティ	Yes	—
IPv6 ネイバー データベース	○	—
ダイナミック ルーティング	○	—
SNMP エンジン ID	**[いいえ（No）]**	—

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て、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。

  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。
- 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。

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 識別子であり、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てから、そのオーナーへのフローを確立します。これは、この接続に関してフォワーダが受信するその他のパケット用です。ディレクタは、フォワーダにもなることができます。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 クッ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 は使用されないので、ディレクタへの問い合わせが必要です）。存続期間が短いフロー（たとえばDNSやICMP）の場合は、フォワーダは問い合わせの代わりにパケットを即座にディレクタに送信し、ディレクタがそのパケットをオーナーに送信します。1 つの接続に対して、複数のフォワーダが存在できます。最も効率的なスループットを実現できるのは、フォワーダが 1 つもなく、接続のすべてのパケットをオーナーが受信するという、優れたロードバランシング方法が使用されている場合です。

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

- フラグメントオーナー：フラグメント化されたパケットの場合、フラグメントを受信するクラスタノードは、フラグメントの送信元と宛先の IP アドレス、およびパケット ID のハッシュを使用してフラグメントオーナーを特定します。その後、すべてのフラグメントがクラスタ制御リンクを介してフラグメント所有者に転送されます。スイッチのロードバランスハッシュで使用される 5 タプルは、最初のフラグメントにのみ含まれているため、フラグメントが異なるクラスタノードにロードバランシングされる場合があります。他のフラグメントには、送信元ポートと宛先ポートは含まれず、他のクラスタノードにロードバランシングされる場合があります。フラグメント所有者は一時的にパケットを再アセンブルするため、送信元/宛先 IP アドレスとポートのハッシュに基づいてディレクタを指定

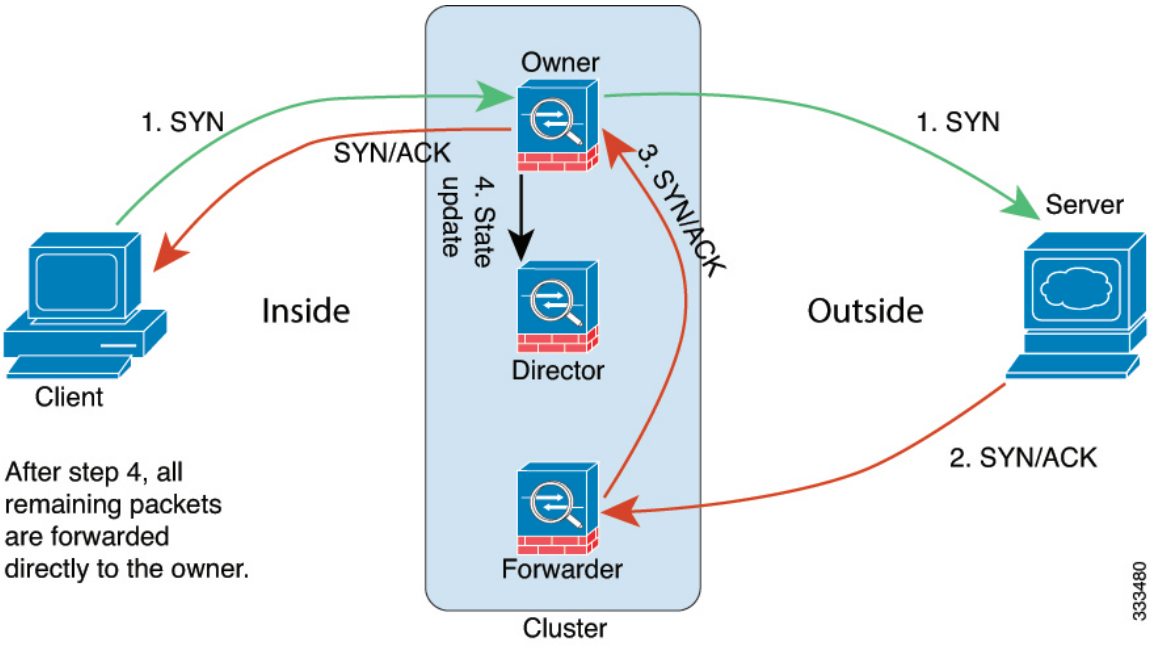
できます。新しい接続の場合は、フラグメントの所有者が接続所有者として登録されます。既存の接続の場合、フラグメント所有者は、クラスタ制御リンクを介して、指定された接続所有者にすべてのフラグメントを転送します。その後、接続所有者はすべてのフラグメントを再構築します。

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。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。



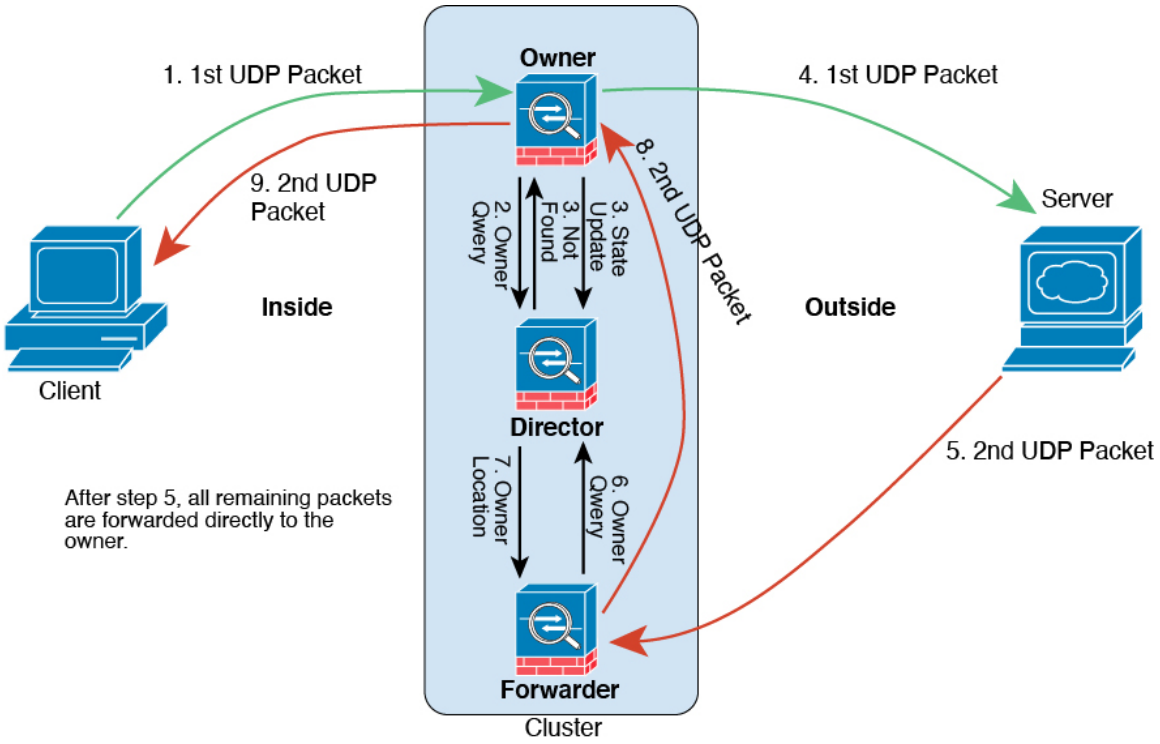
1. SYN パケットがクライアントから発信され、Firewall Threat Defense の 1 つ（ロード バランシング方法に基づく）に配信されます。これがオーナーとなります。オーナーはフローを作成し、オーナー情報をエンコードして SYN Cookie を生成し、パケットをサーバに転送します。
2. SYN-ACK パケットがサーバから発信され、別の Firewall Threat Defense（ロード バランシング方法に基づく）に配信されます。この Firewall Threat Defense はフォワーダです。
3. フォワーダはこの接続を所有してはいないので、オーナー情報を SYN Cookie からデコードし、オーナーへの転送フローを作成し、SYN-ACK をオーナーに転送します。
4. オーナーはディレクタに状態アップデートを送信し、SYN-ACK をクライアントに転送します。

5. ディレクタは状態アップデートをオーナーから受信し、オーナーへのフローを作成し、オーナーと同様に TCP ステート情報を記録します。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てオーナーを特定し、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 *27 : 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 Firewall Threat Defense（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の Threat Defense Virtual クラスタリングの履歴

表 *8 :*

機能	最小 Firewall Management Center	最小 Firewall Threat Defense	詳細
クラスタのヘルスモニターの設定。	7.3.0	いずれか	クラスタのヘルスモニター設定を編集できるようになりました。 新規/変更された画面：**[デバイス（Devices）]** > **[デバイス管理（Device Management）]** > **クラスタ（Cluster）** > **[クラスタのヘルスモニターの設定（Cluster Health Monitor Settings）]** （注） 以前に FlexConfig を使用してこれらの設定を行った場合は、展開前に必ず FlexConfig の設定を削除してください。削除しなかった場合は、FlexConfig の設定によって Management Center の設定が上書きされます。
クラスタ ヘルス モニターダッシュボード。	7.3.0	いずれか	クラスタのヘルス モニター ダッシュボードでクラスタの状態を表示できるようになりました。 新規/変更された画面：**[システム（System）]** > **[正常性（Health）]** > **[モニター（Monitor）]**

機能	最小 Firewall Management Center	最小 Firewall Threat Defense	詳細
Azure の Firewall Threat Defense Virtual のクラスタリング	7.3.0	7.3.0	Azure ゲートウェイロードバランサまたは外部のロードバランサについて、Azure の Firewall Threat Defense Virtual で最大 16 ノードのクラスタリングを構成できるようになりました。 新規/変更された画面： • [デバイス（Devices）] > [デバイス管理（Device Management）] > [クラスタの追加（Add Cluster）] • **[デバイス（Devices）] > [デバイス管理（Device Management）] > [詳細（More）]**メニュー • **[Devices] > [Device Management] > [Cluster]** サポートされているプラットフォーム：Azure の Firewall Threat Defense Virtual
パブリッククラウドでの Firewall Threat Defense Virtual のクラスタリング（Amazon Web Services および Google Cloud Platform）。	7.2.0	7.2.0	Firewall Threat Defense Virtual はパブリッククラウド（AWS および GCP）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。 新しい/変更された画面： • **[デバイス（Devices）] > [デバイス管理（Device Management）] > [デバイスの追加（Add Device）]** • **[デバイス（Devices）] > [デバイス管理（Device Management）] > [詳細（More）]**メニュー • **[Devices] > [Device Management] > [Cluster]** サポートされているプラットフォーム：AWS および GCP 上の Firewall Threat Defense Virtual

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。