AWS への Threat Defense Virtual Auto Scale ソリューションの導入

このドキュメントでは、Threat Defense Virtual Auto Scale ソリューションを AWS に展開する方法について説明します。

AWS での Threat Defense Virtual Auto Scale ソリューションについて

AWS などのパブリッククラウド環境に展開された Threat Defense Virtual インスタンスでは、ネットワークトラフィックでスパイクとディップが発生することがあるアプリケーションがサポートされます。トラフィックのスパイクにより、展開された Threat Defense Virtual インスタンスの数がネットワークトラフィックの検査には足りないシナリオが発生する可能性があります。トラフィックがディップすると、Threat Defense Virtual インスタンスがアイドル状態になり、不要な運用コストが発生する可能性があります。

Auto Scale ソリューションは、トラフィックがスパイクした場合に Threat Defense Virtual インスタンスの数を自動的にスケールアップし、トラフィックが小休止しているときにインスタンスの数をスケールダウンするのに役立つため、ネットワークリソースを効率的に処理して、運用コストを削減できます。

AWS の Threat Defense Virtual Auto Scale は、AWS 環境の Threat Defense Virtual インスタンスに Auto Scaling 機能を追加する完全なサーバーレス実装です（この機能の自動化に関与するヘルパー VM はありません）。

バージョン 6.4 以降、ネットワークロードバランサ（NLB）ベースの Auto Scale ソリューションは、Management Center によって管理される Threat Defense Virtual でサポートされます。バージョン 7.2 以降では、ゲートウェイロードバランサ（GWLB）ベースの Auto Scale ソリューションもサポートされています。

シスコでは、Lambda、Auto Scaling グループ、Elastic Load Balancing（ELB）、Amazon S3 バケット、SNS、CloudWatch などの複数の AWS サービスを使用して、Threat Defense Virtual ファイアウォールの Auto Scaling グループを展開するための CloudFormation テンプレートとスクリプトを提供しています。

Threat Defense Virtual Auto Scale ソリューションは、以下の内容を提供する CloudFormation テンプレートベースの導入です。

- Management Center による Threat Defense Virtual インスタンスの登録と登録解除の完全な自動化。
- スケールアウトされた Threat Defense Virtual インスタンスへの NAT ポリシー、アクセスコントロール ポリシー、およびルートの自動適用。
- ロードバランサとマルチ可用性ゾーンのサポート。

- Management Center でのみ動作し、Device Manager はサポート対象外。

Auto Scale の機能拡張（バージョン 6.7）

- カスタム指標パブリッシャ：新しい Lambda 機能は、Auto Scale グループ内のすべての Threat Defense Virtual インスタンスのメモリ消費量について を 2 分ごとに Management Center をポーリングし、その値を CloudWatch メトリックに公開します。
- メモリ消費に基づく新しいスケーリングポリシーを使用できます。
- Management Center への SSH およびセキュアトンネル用の Threat Defense Virtual プライベート IP 接続。
- Management Center 設定の検証.
- ELB でより追加のリスニングポートを開くためのサポート。
- シングルスタック展開に変更。すべての Lambda 機能と AWS リソースは、合理化された展開のためにシングルスタックから展開されます。

NLB を使用した Auto Scale ソリューション

AWS ロードバランサはインバウンドで開始された接続のみを許可するため、外部で生成されたトラフィックのみが Cisco Threat Defense Virtual ファイアウォール経由で内部を通過できます。

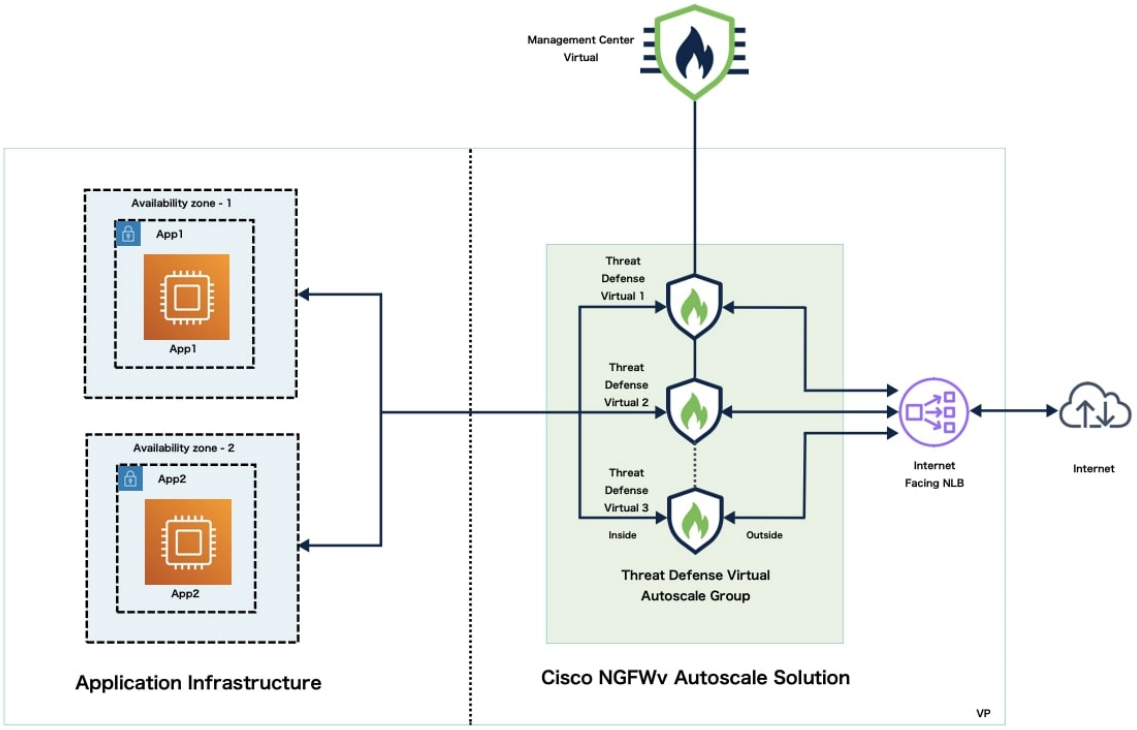
インターネットに面したロードバランサは、ネットワークロードバランサまたはアプリケーション ロード バランサです。いずれの場合も、AWS のすべての要件と条件が適用されます。以下のサンプルトポロジで示されているように、点線の右側部分は Threat Defense Virtual テンプレートを介して展開されます。左側はユーザー定義の部分です。

（注） アプリケーションが開始したアウトバウンドトラフィックは Threat Defense Virtual を通過しません。

トラフィックのポートベースの分岐が可能です。この分岐は、NAT ルールによって実現できます。Management Center での「ホストオブジェクトの作成」、「デバイスグループの追加」、「NLB を使用した Auto Scale ソリューション：ネットワークアドレス変換（NAT）ポリシーの設定と展開」、基本的なアクセス コントロール ポリシーの作成 （37 ページ）、「基本的なアクセス コントロール ポリシーの作成」を参照してください。たとえば、インターネットに面した LB DNS、ポート：80 のトラフィックは、アプリケーション 1 にルーティングでき、ポート：88 のトラフィックはアプリケーション 2 にルーティングできます。

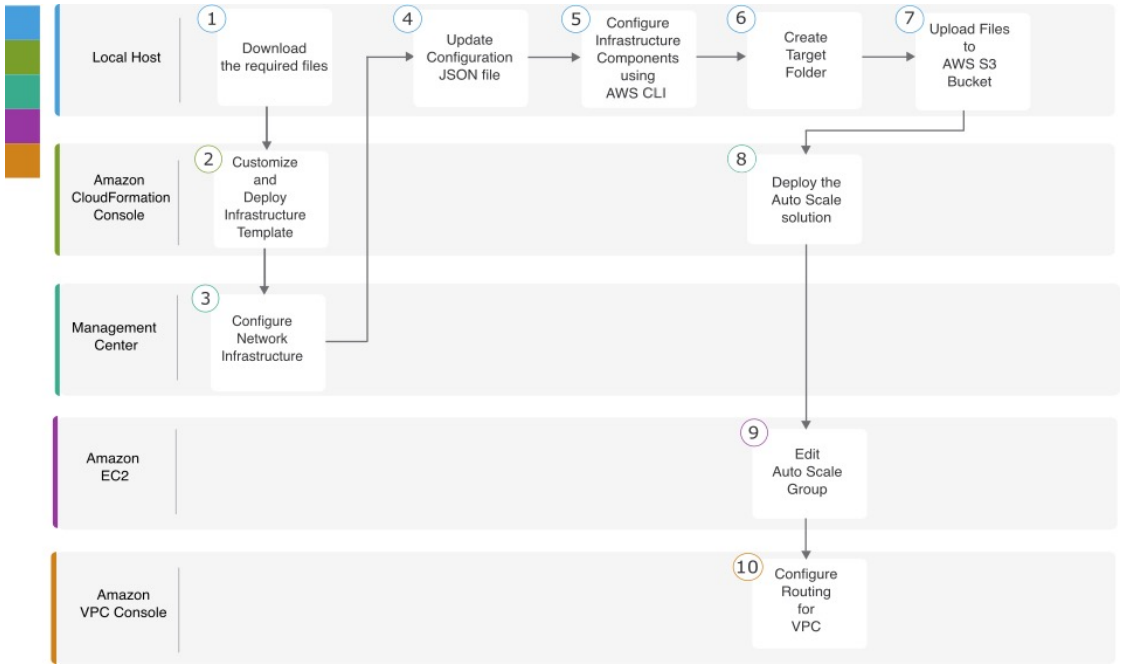
トポロジの例

図 *1 : NLB* を使用した *Threat Defense Virtual Auto Scale* ソリューション



NLB を使用して Auto Scale ソリューションを展開するためのエンドツーエンドのプロセス

次のフローチャートは、Amazon Web Services（AWS）に NLB を使用して Threat Defense Virtual Auto Scale ソリューションを展開する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 からローカルホストへの必要なファイルと CFT のダウンロード
2	Amazon CloudFormation コンソール	NLB を使用した Auto Scale ソリューション：Amazon CloudFormation コンソールでの NLB インフラストラクチャテンプレートのカスタマイズと展開 （30ページ）
3	Management Center	Management Center でのネットワーク インフラストラクチャの設定 （32 ページ）
4	ローカルホスト	Configuration.json ファイルの更新 （38 ページ）
5	ローカルホスト	AWS CLI を使用したインフラストラクチャ コンポーネントの設定 （39 ページ）
6	ローカルホスト	target フォルダの作成 （41 ページ）
7	ローカルホスト	Amazon S3 バケットへのファイルのアップロード （41 ページ）
8	Amazon CloudFormation コンソール	NLB を使用した Auto Scale ソリューション：NLB を使用した Auto Scale ソリューションの展開 （41 ページ）
9	Amazon EC2 コンソール	Auto Scale グループの編集 （44 ページ）

	ワークスペース	手順
10	Amazon VPC コンソール	VPC のルーティングの設定 （43 ページ）

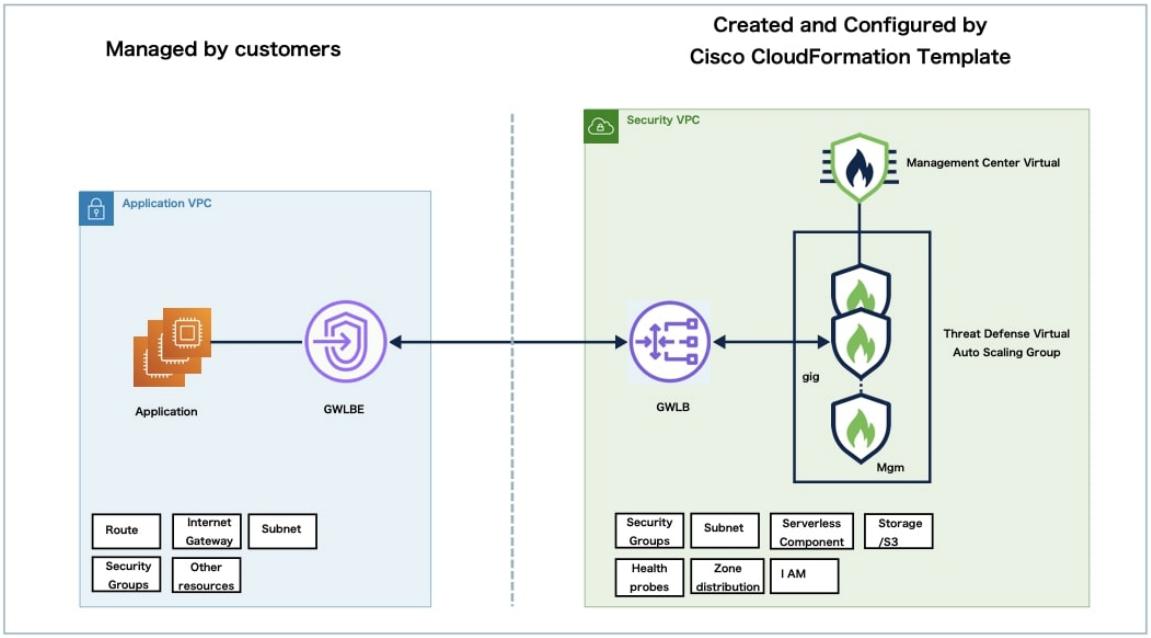
GWLB を使用した Auto Scale ソリューション

AWS ゲートウェイロードバランサ（GWLB）を使用すると、インバウンド接続とアウトバウンド接続の両方を許可できるため、内部と外部で生成されたトラフィックが Cisco Threat Defense Virtual ファイアウォール経由で内部を通過できます。

トラフィックは GWLBe から GWLB に送信され、その後、検査のために Threat Defense Virtual に送信されます。いずれの場合も、AWS のすべての要件と条件が適用されます。導入例の図に示されているように、点線の右側部分は Threat Defense Virtual テンプレートを介して展開された Threat Defense Virtual GWLB Auto Scale ソリューションです。左側は完全にユーザー定義の部分です。

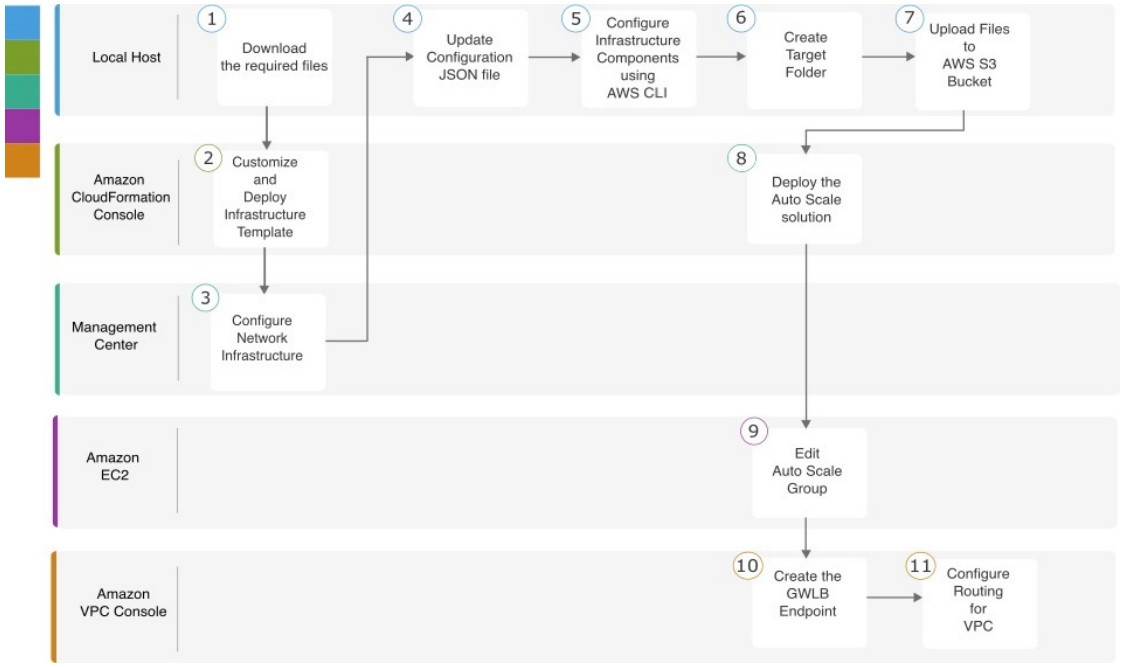
トポロジの例

図 *2 : GWLB* を使用した *Threat Defense Virtual Auto Scale* ソリューション



GWLB を使用して Auto Scale ソリューションを展開するためのエンドツーエンドのプロセス

次のフローチャートは、Amazon Web Services（AWS）に GWLB を使用して Threat Defense Virtual Auto Scale ソリューションを展開する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 からローカルホストへの必要なファイルと CFT のダウンロード
2	Amazon CloudFormation コンソール	GWLB を使用した Auto Scale ソリューション：Amazon CloudFormation コンソールでの GWLB インフラストラクチャ テンプレートのカスタマイズと展開 （31 ページ）
3	Management Center	Management Center でのネットワーク インフラストラクチャの設定 （32 ページ）
4	ローカルホスト	Configuration.json ファイルの更新 （38 ページ）
5	ローカルホスト	AWS CLI を使用したインフラストラクチャ コンポーネントの設定 （39 ページ）
6	ローカルホスト	target フォルダの作成 （41 ページ）

	ワークスペース	手順
7	ローカルホスト	Amazon S3 バケットへのファイルのアップロード （41 ページ）
8	Amazon CloudFormation コンソール	GWLB を使用した Auto Scale ソリューション：GWLB を使用した Auto Scale ソリューションの展開 （42 ページ）
9	Amazon EC2 コンソール	Auto Scale グループの編集 （44 ページ）
10	Amazon VPC コンソール	GWLB ソリューションを使用した Auto Scale：GWLB エンドポイントの作成 （42 ページ）
11	Amazon VPC コンソール	VPC のルーティングの設定 （43 ページ）

Threat Defense Virtual および AWS のガイドラインと制限事項

ライセンシング

- シスコ スマート ライセンス アカウントを使用する BYOL（Bring Your Own License）がサポートされています。
- PAYG（Pay As You Go）ライセンス。顧客がシスコ スマート ライセンシングを購入せずに Threat Defense Virtual を実行できる従量制課金モデル。登録された PAYG Threat Defense Virtual デバイスでは、ライセンス供与されたすべての機能（マルウェア、脅威、URL フィルタリング、VPN など）が有効になっています。ライセンス供与された機能は、Management Center から編集または変更することはできません（バージョン 6.5 以上）。

Note PAYG ライセンスは、Device Manager モードで展開されている Threat Defense Virtual デバイスではサポートされていません。

Threat Defense Virtual デバイスのライセンス取得のガイドラインについては、『Firepower Management Center Administration Guide』の「Licenses」の章を参照してください。

Threat Defense Virtual スマートライセンスのパフォーマンス階層

Threat Defense Virtual のバージョン 7.0.0 リリース以降では、Threat Defense Virtual は導入要件に基づいて異なるスループットレベルと VPN 接続制限を提供するパフォーマンス階層型ライセンスをサポートしています。

Table 1: Threat Defense Virtual 権限付与に基づくライセンス機能の制限

パフォーマンス階層	デバイス仕様（コア/RAM）	レート制限	RA VPN セッション制限
FTDv5	4 コア/8 GB	100 Mbps	50
FTDv10	4 コア/8 GB	1 Gbps	250
FTDv20	4 コア/8 GB	3Gbps	250
FTDv30	8 コア/16 GB	5 Gbps	250
FTDv50	12 コア/24 GB	10 Gbps	750
FTDv100	16 コア/34 GB	16 Gbps	10,000

ベスト プラクティス

- Management Center Virtual で必要なコンポーネントを設定していることを確認します。詳細については、「Management Center でのネットワーク インフラストラクチャの設定」を参照してください。
- CloudFormation テンプレートのパラメータに必要な値を入力していることを確認します。詳細について は、「GitHub の CloudFormation テンプレート」を参照してください。

前提条件

- AWS アカウントhttp://aws.amazon.com/ で 1 つ作成できます。
- Threat Defense Virtual のコンソールにアクセスするには、SSH クライアント（例：Windows の場合は PuTTY、macOS の場合はターミナル）が必要です。
- Cisco スマートアカウント。Cisco Software Central で作成できますhttps://software.cisco.com/ 。
- 設定ファイルとテンプレートをダウンロードするための GitHub アカウント。
- Threat Defense Virtual インターフェイスの要件：
  - 管理インターフェイス（2）：1 つは Threat Defense Virtual を Management Center に接続するために使用されます。もう 1 つは診断目的に使用され、通過トラフィックには使用できません。
  - 必要に応じて、管理インターフェイスの代わりに、データインターフェイスを Management Center の管理用に設定できます。管理インターフェイスはデータインターフェイス管理の前提条件であるため、初期設定でこれを設定する必要があります。データインターフェイスから Management Center へのアクセスは、高可用性の展開ではサポートされません。Management Center アクセス用のデータインターフェイスの

設定の詳細については、FTD コマンドリファレンスの configure network management-data-interface コマンドを参照してください。

- トラフィック インターフェイス（2）：Threat Defense Virtual を内部ホストおよびパブリックネットワークに接続するために使用されます。

- 通信パス：Threat Defense Virtual にアクセスするためのパブリック IP/Elastic IP。

GWLB または NLB を使用した Auto Scale ソリューションの設定に必要なコンポーネント

Auto Scale ソリューションは、次のコンポーネントで構成されています。

CloudFormation テンプレート

CloudFormation テンプレートは、AWS の Auto Scale ソリューションの設定に必要なリソースを展開するために使用されます。テンプレートの構成は次のとおりです。

- Auto Scale グループ、ロードバランサ、セキュリティグループ、およびその他のコンポーネント。
- 展開をカスタマイズするためのユーザー入力を取り込むテンプレート。

（注） テンプレートのユーザー入力の検証には限界があるため、展開時に入力を検証するのはユーザーの責任です。

Lambda 関数

Auto Scale ソリューションは、Python で開発された一連の Lambda 関数で、ライフサイクルフック、SNS、CloudWatch イベントやアラームイベントからトリガーされます。基本的な機能は次のとおりです。

- インスタンスに対して Diag、Gig0/0、および Gig 0/1 インターフェイスを追加/削除します。
- ロードバランサのターゲットグループに Gig0/1 インターフェイスを登録します。
- Management Center で Threat Defense Virtual を新規登録します。
- Management Center を介して新規の Threat Defense Virtual を展開します。
- スケールインした Threat Defense Virtual を Management Center から登録解除（削除）します。
- Management Center からメモリメトリックをパブリッシュします。

Lambda 関数は、Python パッケージの形式でお客様に提供されます。

ライフサイクルフック

- ライフサイクルフックは、インスタンスに関するライフサイクルの変更通知を取得するために使用されます。
- インスタンス起動の場合、ライフサイクルフックを使用して、Threat Defense Virtual インスタンスにインターフェイスを追加し、ターゲットグループに外部インターフェイス IP を登録できる Lambda 機能をトリガーします。
- インスタンス終了の場合、ライフサイクルフックを使用して Lambda 機能をトリガーし、ターゲットグループから Threat Defense Virtual インスタンスを登録解除します。

Simple Notification Service（SNS）

- AWS の Simple Notification Service（SNS）を使用してイベントが生成されます。
- AWS にはサーバーレス Lambda 関数に適した Orchestrator がないという制限があるため、ソリューションは、イベントに基づいて Lambda 関数をオーケストレーションするための一種の関数チェーンとして SNS を使用します。

VPC

アプリケーション要件に応じて VPC を作成する必要があります。VPC には、インターネットへのルートがある少なくとも 1 つのサブネットを持つインターネットゲートウェイがあることが想定されます。セキュリティグループ、サブネットなどの要件については、該当するセクションを参照してください。

セキュリティ グループ

提供された Auto Scale グループテンプレートでは、すべての接続が許可されます。Auto Scale ソリューションを機能させるために必要なのは、次の接続だけです。

ポート	使用方法	サブネット
8305	Management Center から Threat Defense Virtual へのセキュアなトンネル接続	管理サブネット
正常性プローブポート（デフォルト：8080）	インターネットに面したロードバランサの正常性プローブ	外部サブネット、内部サブネット
アプリケーションポート	アプリケーション データ トラフィック	外部サブネット、内部サブネット

Management Center インスタンスのセキュリティグループまたは ACL

これらは、Lambda 機能と Management Center 間の HTTPS 接続を許可するために必要です。Lambda 機能は、NAT ゲートウェイをデフォルトルートとして持つ Lambda サブネットに保持

されるため、Management Center には NAT ゲートウェイ IP アドレスからのインバウンド HTTPS 接続を設定できます。

サブネット

サブネットは、アプリケーションの要件に応じて作成できます。Threat Defense Virtual が動作するには 3 つのサブネットが必要です。

（注） 複数の可用性ゾーンのサポートが必要な場合、サブネットは AWS クラウド内のゾーンプロパティであるため、各ゾーンにサブネットが必要です。

外部サブネット

外部サブネットには、インターネットゲートウェイへの「0.0.0.0/0」のデフォルトルートが必要です。このサブネットには、Threat Defense Virtual の外部インターフェイスが含まれ、インターネットに面した NLB も含まれます。

内部サブネット

これは、NAT/インターネットゲートウェイの有無にかかわらず、アプリケーションサブネットに似ています。Threat Defense Virtual の正常性プローブでは、ポート 80 経由で AWS メタデータサーバー（169.254.169.254）に到達できる必要があることに注意してください。

（注） この Auto Scale ソリューションでは、ロードバランサの正常性プローブが inside/Gig0/0 インターフェイスを介して AWS メタデータサーバーにリダイレクトされますが、ロードバランサから Threat Defense Virtual に送信される正常性プローブ接続を提供する独自のアプリケーションで変更できます。その場合、AWS メタデータサーバー オブジェクトをアプリケーションの IP アドレスに置き換えて、正常性プローブ応答を提供する必要があります。

管理サブネット

このサブネットには、Threat Defense Virtual 管理インターフェイスが含まれます。 このサブネットで Management Center を使用している場合、Threat Defense Virtual への Elastic IP アドレス（EIP）の割り当ては任意です。診断インターフェイスもこのサブネット上にあります。

Lambda サブネット

AWS Lambda 関数では、デフォルトゲートウェイとして NAT ゲートウェイを持つ 2 つのサブネットが必要です。これにより、Lambda 関数が VPC に対してプライベートになります。Lambda サブネットは、他のサブネットと同じ幅である必要はありません。

アプリケーションサブネット

Auto Scale ソリューションからこのサブネットに課せられる制限はありませんが、アプリケーションに VPC 外部のアウトバウンド接続が必要な場合は、サブネット上にそれぞれのルートが設定されている必要があります。これは、アウトバウンドで開始されたトラフィックがロー

ドバランサを通過しないためです。詳細については、AWS Elastic Load Balancing User Guide [英語] を参照してください。

サーバーレスコンポーネント

S3 バケット

Amazon Simple Storage Service（Amazon S3）は、業界をリードする拡張性、データ可用性、セキュリティ、およびパフォーマンスを提供するオブジェクト ストレージ サービスです。必要なすべてのファイルを S3 バケットに配置できます。

テンプレートが展開されると、S3 バケット内の zip ファイルを参照して Lambda 機能が作成されるため、S3 バケットはユーザーアカウントにアクセスできる必要があります。

GitHub の CloudFormation テンプレート

サポートされている Auto Scale ソリューション用に 2 つのテンプレートセットが用意されています。1 つのセットは NLB を使用した Auto Scale ソリューションの設定用で、もう 1 つのセットは GWLB を使用した Auto Scale ソリューションの設定用です。

NLB を使用した Auto Scale ソリューション

GitHub では、次のテンプレートを使用できます。

- infrastructure.yaml
- deploy_ngfw_autoscale.yaml

表 *2:* テンプレートパラメータのリスト

パラメータ	使用できる値/タイプ	説明
PodNumber	許可される文字列パターン：'^\d{1,3}$'	これはポッド番号です。これは、Auto Scale グループ名（threat defense virtual-Group-Name）の末尾に追加されます。たとえば、値が「1」の場合、グループ名は threat defense virtual-Group-Name-1 になります。 1 桁以上 3 桁以下の数字である必要があります。デフォルト：1。
AutoscaleGrpNamePrefix	文字列	これは Auto Scale グループ名プレフィックスです。ポッド番号がサフィックスとして追加されます。 最大：18 文字 例：Cisco-threat Defense virtual-1。

パラメータ	使用できる値/タイプ	説明
NotifyEmailID	文字列	Auto Scale イベントはこの電子メールアドレスに送信されます。サブスクリプション電子メール要求を受け入れる必要があります。 例：admin@company.com。
VpcId	文字列	デバイスを展開する必要がある VPC ID。これは、AWS の要件に従って設定する必要があります。 タイプ：AWS::EC2::VPC::Id 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
LambdaSubnets	リスト	Lambda 関数が展開されるサブネット。 タイプ：List<AWS::EC2::Subnet::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
LambdaSG	リスト	Lambda 機能のセキュリティグループ。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
S3BktName	文字列	ファイルの S3 バケット名。これは、AWS の要件に従ってアカウントに設定する必要があります。 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
LoadBalancerType	文字列	インターネットに面したロードバランサのタイプ（「アプリケーション」または「ネットワーク」）。 例：アプリケーション

パラメータ	使用できる値/タイプ	説明
LoadBalancerSG	文字列	ロードバランサのセキュリティグループ。ネットワークロードバランサの場合は使用されません。ただし、セキュリティグループ ID を指定する必要があります。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
LoadBalancerPort	整数	ロードバランサポート。このポートは、選択したロードバランサタイプに基づいて、プロトコルとして HTTP/HTTPS または TCP/TLS を使用して LB で開きます。 ポートが有効な TCP ポートであることを確認します。これはロードバランサリスナーの作成に使用されます。 デフォルト：80
SSL認証	文字列	セキュアポート接続の SSL 証明書の ARN。指定しない場合、ロードバランサで開かれるポートは TCP/HTTP になります。指定した場合、ロードバランサで開かれるポートは TLS/HTTPS になります。
TgHealthPort	整数	このポートは、正常性プローブのターゲットグループによって使用されます。Threat Defense Virtual のこのポートに到達する正常性プローブは、AWS メタデータサーバーにルーティングされるため、トラフィックには使用しないでください。このポートは有効な TCP ポートである必要があります。 アプリケーション自体が正常性プローブに応答するようにする場合は、そのように Threat Defense Virtual の NAT ルールを変更できます。そのような場合、アプリケーションが応答しないと、Unhealthy インスタンスのしきい値アラームにより、Threat Defense Virtual は非正常とマークされて削除されます。 例：8080

パラメータ	使用できる値/タイプ	説明
AssignPublicIP	ブール値	「true」を選択すると、パブリック IP が割り当てられます。BYOL タイプの Threat Defense Virtual の場合、パブリック IP は https://tools.cisco.com に接続するために必要です。 例：TRUE
InstanceType	文字列	Amazon マシンイメージ（AMI）は、さまざまなインスタンスタイプをサポートしています。インスタンスタイプによって、インスタンスのサイズと必要なメモリ容量が決まります。 Threat Defense Virtual をサポートする AMI インスタンスタイプのみを使用する必要があります。 例：c4.2xlarge
LicenseType	文字列	Threat Defense Virtual ライセンスタイプ（BYOL または PAYG）。関連する AMI ID が同じライセンスタイプであることを確認します。 例：BYOL
AmiId	文字列	Threat Defense Virtual AMI ID（有効な Cisco Threat Defense Virtual AMI ID）。 タイプ：AWS::EC2::Image::Id リージョンとイメージの目的のバージョンに応じて、正しい AMI ID を選択してください。Auto Scale 機能は、バージョン 6.4+、BYOL/PAYG イメージをサポートします。いずれの場合も、AWS マーケットプレイスでライセンスに同意する必要があります。 BYOL の場合、設定 JSON ファイルの「licenseCaps」キーを「BASE」、「MALWARE」、「THREAT」、「URLFilter」などの機能で更新してください。

パラメータ	使用できる値/タイプ	説明
NoOfAZs	整数	Threat Defense Virtual を展開する必要がある可用性ゾーンの数（1 ー 3）。ALB 導入の場合、AWS で必要な最小値は 2 です。 例：2。
ListOfAzs	カンマ区切り文字列	ゾーンの順序のカンマ区切りリスト。 （注） ゾーンのリスト順は重要です。サブネットリストは同じ順序で指定する必要があります。 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。 例：us-east-1a、us-east-1b、us-east-1c
MgmtInterfaceSG	文字列	Threat Defense Virtual 管理インターフェイスのセキュリティグループ。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
InsideInterfaceSG	文字列	Threat Defense Virtual 内部インターフェイスのセキュリティグループ。 タイプ：AWS::EC2::SecurityGroup::Id 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
OutsideInterfaceSG	文字列	Threat Defense Virtual 外部インターフェイスのセキュリティグループ。 タイプ：AWS::EC2::SecurityGroup::Id 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。 例：sg-0c190a824b22d52bb

パラメータ	使用できる値/タイプ	説明
MgmtSubnetId	カンマ区切りリスト	管理サブネットIDのカンマ区切りリスト。リストは、対応する可用性ゾーンと同じ順序にする必要があります。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
InsideSubnetId	カンマ区切りリスト	内部/Gig0/0 サブネット ID のカンマ区切りリスト。リストは、対応する可用性ゾーンと同じ順序にする必要があります。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
OutsideSubnetId	カンマ区切りリスト	外部/Gig0/1 サブネット ID のカンマ区切りリスト。リストは、対応する可用性ゾーンと同じ順序にする必要があります。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
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）。指定した場合、Management Center および Threat Defense Virtual のパスワードを暗号化する必要があります。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。 暗号化パスワードの生成例： " aws kms encrypt --key-id <KMS ARN> --plaintext <password>" 次のような生成されたパスワードを使用してください。 例：arn:aws:kms:us-east-1:[AWS Account]:key/7d586a25-5875-43b1-bb68-a452e2f6468e

パラメータ	使用できる値/タイプ	説明
ngfwPassword	文字列	すべての Threat Defense Virtual インスタンスには、起動テンプレート（Auto Scale グループ）の [ユーザーデータ（Userdata）] フィールドに入力されたデフォルトのパスワードが設定されています。 この入力により、Threat Defense Virtual にアクセスできるようになると、パスワードが新しく提供されたパスワードに変更されます。 KMS ARN が使用されていない場合は、プレーンテキストのパスワードを使用してください。KMS ARN が使用されている場合は、暗号化されたパスワードを使用する必要があります。 例：Cisco123789! または AQIAgcQFAGtz/hvaxMtJvY/x/rfHnI3lPpSXU
fmcServer	数値文字列	Lambda 機能と Threat Defense Virtual 管理インターフェイスの両方に到達可能な Management Center 管理用の IP アドレス。 例：10.10.17.21
fmcOperationsUsername	文字列	Management Center を管理する際に作成された Network-Admin 以上の特権ユーザー。ユーザーおよびロールの作成の詳細については、Cisco Secure Firewall Management Center デバイス コンフィギュレーション ガイド [英語] を参照してください。 例：apiuser-1
fmcOperationsPassword	文字列	KMS ARN が記載されていない場合は、プレーンテキストのパスワードを使用してください。記載されている場合は、暗号化されたパスワードを使用する必要があります。 例：Cisco123@ または AQICAHgcQAtz/hvaxMtJvY/x/mKI3dFPpSXUHQRnCAajB
fmcDeviceGrpName	文字列	Management Center のデバイスグループ名。 例：AWS-Cisco-NGFW-VMs-1

パラメータ	使用できる値/タイプ	説明
fmcPerformanceLicenseTier	文字列	Threat Defense Virtual デバイスを Management Center Virtual に登録する際に使用されたパフォーマンス階レイヤライセンス。 使用できる値：FTDv/FTDv5/FTDv10/FTDv20/FTDv30/FTDv50/FTDv100
fmcPublishMetrics	ブール値	「TRUE」に設定すると、指定されたデバイスグループ内の登録済み Threat Defense Virtual センサーのメモリ消費量を取得するために、2 分に 1 回実行される Lambda 機能が作成されます。 使用可能な値：TRUE、FALSE 例：TRUE
fmcMetricsUsername	文字列	AWS CloudWatch にメトリックを公開するための一意の Management Center ユーザー名。ユーザーおよびロールの作成の詳細については、Cisco Secure Firewall Management Center デバイス コンフィギュレーション ガイド [英語] を参照してください。 「fmcPublishMetrics」が「FALSE」に設定されている場合は、この入力を行う必要はありません。 例：publisher-1
fmcMetricsPassword	文字列	AWS CloudWatch にメトリックを公開するための Management Center パスワード。KMS ARN が記載されていない場合は、プレーンテキストのパスワードを使用してください。記載されている場合は、暗号化されたパスワードを使用する必要があります。 「fmcPublishMetrics」が「FALSE」に設定されている場合は、この入力を行う必要はありません。 例：Cisco123789!

パラメータ	使用できる値/タイプ	説明
CpuThresholds	カンマ区切り整数	CPU しきい値の下限と CPU しきい値の上限。最小値は 0 で、最大値は 99 です。 デフォルト：10, 70 しきい値の下限はしきい値の上限よりも小さくする必要があります。 例：30,70
MemoryThresholds	カンマ区切り整数	MEM しきい値の下限と MEM しきい値の上限。最小値は 0 で、最大値は 99 です。 デフォルト：40, 70 しきい値の下限はしきい値の上限よりも小さくする必要があります。「fmcPublishMetrics」パラメータが「FALSE」の場合、影響はありません。 例：40,50

GWLB を使用した Auto Scale ソリューション

GitHub で利用可能なテンプレート

- infrastructure_gwlb.yaml
- deploy_ngfw_autoscale_with_gwlb.yaml

表 *3:* テンプレートパラメータのリスト

パラメータ	使用できる値/タイプ	説明
PodNumber	文字列 許可パターン：'^\d{1,3}$'	これはポッド番号です。Auto Scale グループ名（Threat Defense Virtual-Group-Name）の末尾に追加されます。たとえば、この値が「1」の場合、グループ名は *Threat Defense Virtual-Group-Name-1.* になります。 1 桁以上 3 桁以下の数字である必要があります。 デフォルト：1

パラメータ	使用できる値/タイプ	説明
AutoscaleGrpNamePrefix	文字列	これはAuto Scaleグループ名プレフィックスです。ポッド番号がサフィックスとして追加されます。 最大：18 文字 例：Cisco-Threat Defense Virtual-1
NotifyEmailID	文字列	Auto Scale イベントはこの電子メールアドレスに送信されます。サブスクリプション電子メール要求を受け入れる必要があります。 例：admin@company.com
VpcId	文字列	デバイスを展開する必要がある VPC ID。これは、AWS の要件に従って設定する必要があります。 タイプ：AWS::EC2::VPC::Id 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
LambdaSubnets	リスト	Lambda 関数が展開されるサブネット。 タイプ：List<AWS::EC2::Subnet::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
LambdaSG	リスト	Lambda 機能のセキュリティグループ。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
S3BktName	文字列	ファイルの S3 バケット名。これは、AWS の要件に従ってアカウントに設定する必要があります。 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。

パラメータ	使用できる値/タイプ	説明
LoadBalancerType	文字列	インターネットに面したロードバランサのタイプ（「アプリケーション」または「ネットワーク」）。 例：アプリケーション
LoadBalancerSG	文字列	ロードバランサのセキュリティグループ。ネットワークロードバランサの場合は使用されません。ただし、セキュリティグループ ID を指定する必要があります。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
LoadBalancerPort	整数	ロードバランサポート。このポートは、選択したロードバランサタイプに基づいて、プロトコルとして HTTP/HTTPS または TCP/TLS を使用して LB で開きます。 ポートが有効な TCP ポートであることを確認します。これはロードバランサリスナーの作成に使用されます。 デフォルト：80
SSL認証	文字列	セキュアポート接続の SSL 証明書の ARN。指定しない場合、ロードバランサで開かれるポートは TCP/HTTP になります。指定した場合、ロードバランサで開かれるポートは TLS/HTTPS になります。

パラメータ	使用できる値/タイプ	説明
TgHealthPort	整数	このポートは、正常性プローブのターゲットグループによって使用されます。Threat Defense Virtual のこのポートに到達する正常性プローブは、AWS メタデータサーバーにルーティングされるため、トラフィックには使用しないでください。このポートは有効な TCP ポートである必要があります。 アプリケーション自体が正常性プローブに応答するようにする場合は、それに応じて Threat Defense Virtual の NAT ルールを変更できます。このような場合、アプリケーションが応答しないと、Threat Defense Virtual は Unhealthy インスタンスのしきい値アラームにより、非正常としてマークされ、削除されます。 例：8080
AssignPublicIP	ブール値	「true」を選択すると、パブリック IP が割り当てられます。BYOL タイプの Threat Defense Virtual の場合、これは https://tools.cisco.com に接続するために必要です。 例：TRUE
InstanceType	文字列	Amazon マシンイメージ（AMI）は、さまざまなインスタンスタイプをサポートしています。インスタンスタイプによって、インスタンスのサイズと必要なメモリ容量が決まります。 Threat Defense Virtual をサポートする AMI インスタンスタイプのみを使用する必要があります。 例：c4.2xlarge
LicenseType	文字列	Threat Defense Virtual ライセンスタイプ（BYOL または PAYG）。関連する AMI ID が同じライセンスタイプであることを確認します。 例：BYOL

パラメータ	使用できる値/タイプ	説明
AmiId	文字列	Threat Defense Virtual AMI ID（有効な Cisco Threat Defense Virtual AMI ID）。 タイプ：AWS::EC2::Image::Id リージョンとイメージの目的のバージョンに応じて、正しい AMI ID を選択してください。Auto Scale 機能は、バージョン 6.4+、BYOL/PAYG イメージをサポートします。いずれの場合も、AWS マーケットプレイスでライセンスに同意する必要があります。 BYOL の場合、設定 JSON ファイルの「licenseCaps」キーを「BASE」、「MALWARE」、「THREAT」、「URLFilter」などの機能で更新してください。
NoOfAZs	整数	Threat Defense Virtual を展開する必要がある可用性ゾーンの数（1 ー 3）。ALB 導入の場合、AWS で必要な最小値は 2 です。 例：2。
ListOfAzs	カンマ区切り文字列	ゾーンの順序のカンマ区切りリスト。 （注） ゾーンのリスト順は重要です。サブネットリストは同じ順序で指定する必要があります。 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。 例：us-east-1a、us-east-1b、us-east-1c
MgmtInterfaceSG	文字列	Threat Defense Virtual 管理インターフェイスのセキュリティグループ。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。

パラメータ	使用できる値/タイプ	説明
InsideInterfaceSG	文字列	Threat Defense Virtual 内部インターフェイスのセキュリティグループ。 タイプ：AWS::EC2::SecurityGroup::Id 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
OutsideInterfaceSG	文字列	Threat Defense Virtual 外部インターフェイスのセキュリティグループ。 タイプ：AWS::EC2::SecurityGroup::Id 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。 例：sg-0c190a824b22d52bb
MgmtSubnetId	カンマ区切りリスト	管理サブネットIDのカンマ区切りリスト。リストは、対応する可用性ゾーンと同じ順序にする必要があります。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
InsideSubnetId	カンマ区切りリスト	内部/Gig0/0サブネットIDのカンマ区切りリスト。リストは、対応する可用性ゾーンと同じ順序にする必要があります。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。

パラメータ	使用できる値/タイプ	説明
OutsideSubnetId	カンマ区切りリスト	外部/Gig0/1 サブネット ID のカンマ区切りリスト。リストは、対応する可用性ゾーンと同じ順序にする必要があります。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
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）。 指定した場合、Management Center と Threat Defense Virtual のパスワードを暗号化する必要があります。 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。 暗号化パスワードの生成例： " aws kms encrypt --key-id <KMS ARN> --plaintext <password> " 次のような生成されたパスワードを使用してください。 例：arn:aws:kms:us-east-1:[AWS Account]:key/7d586a25-5875-43b1-bb68-a452e2f6468e
ngfwPassword	文字列	すべての Threat Defense Virtual インスタンスには、起動テンプレート（自動スケールグループ）の [ユーザーデータ（Userdata）] フィールドに入力されたデフォルトのパスワードが設定されています。 この入力により、Threat Defense Virtual にアクセスできるようになると、パスワードが新しく提供されたパスワードに変更されます。 KMS ARN が使用されていない場合は、プレーンテキストのパスワードを使用してください。KMS ARN が使用されている場合は、暗号化されたパスワードを使用する必要があります。 例：Cisco123789! または AQIAgcQFAGtz/hvaxMtJvY/x/rfHnI3lPpSXU
fmcServer	数値文字列	Lambda 関数と Threat Defense Virtual 管理インターフェイスの両方に到達可能な Management Center 管理用の IP アドレス。 例：10.10.17.21

パラメータ	使用できる値/タイプ	説明
fmcOperationsUsername	文字列	Management Center を管理する際に作成された Network-Admin 以上の特権ユーザー。ユーザとロールの作成の詳細については、『Cisco Secure Firewall Management Center デバイス構成ガイド』を参照してください。 例：apiuser-1
fmcOperationsPassword	文字列	KMS ARN が記載されていない場合は、プレーンテキストのパスワードを使用してください。記載されている場合は、暗号化されたパスワードを使用する必要があります。 例：Cisco123@ または AQICAHgcQAtz/hvaxMtJvY/x/mKI3clFPpSXUHQRnCAajB
fmcDeviceGrpName	文字列	Management Center のデバイスグループ名。 例：AWS-Cisco-NGFW-VMs-1
fmcPerformanceLicenseTier	文字列	Threat Defense Virtual デバイスを Management Center Virtual に登録する際に使用されたパフォーマンス階レイヤライセンス。 使用できる値：FTDv/FTDv20/FTDv30/FTDv50/FTDv100 （注） FTDv5 および FTDv10 パフォーマンス階レイヤライセンスは、AWS ゲートウェイロードバランサではサポートされていません。
fmcPublishMetrics	ブール値	「TRUE」に設定すると、指定されたデバイスグループ内の登録済み Threat Defense Virtual センサーのメモリ消費量を取得するために、2 分に 1 回実行される Lambda 関数が作成されます。 使用可能な値：TRUE、FALSE 例：TRUE

パラメータ	使用できる値/タイプ	説明
fmcMetricsUsername	文字列	AWS CloudWatch にメトリックを公開するための一意の Management Center ユーザー名。ユーザとロールの作成の詳細については、『Cisco Secure Firewall Management Center デバイス構成ガイド』を参照してください。 「fmcPublishMetrics」が「FALSE」に設定されている場合は、この入力を行う必要はありません。 例：publisher-1
fmcMetricsPassword	文字列	AWS CloudWatch にメトリックを公開するための Management Center パスワード。KMS ARN が記載されていない場合は、プレーンテキストのパスワードを使用してください。記載されている場合は、暗号化されたパスワードを使用する必要があります。 「fmcPublishMetrics」が「FALSE」に設定されている場合は、この入力を行う必要はありません。 例：Cisco123789!
CpuThresholds	カンマ区切り整数	CPU しきい値の下限と CPU しきい値の上限。最小値は 0 で、最大値は 99 です。 デフォルト：10, 70 しきい値の下限はしきい値の上限よりも小さくする必要があります。 例：30,70
MemoryThresholds	カンマ区切り整数	MEM しきい値の下限と MEM しきい値の上限。最小値は 0 で、最大値は 99 です。 デフォルト：40, 70 しきい値の下限はしきい値の上限よりも小さくする必要があります。「fmcPublishMetrics」パラメータが「FALSE」の場合、影響はありません。 例：40,50

GitHub からローカルホストへの必要なファイルと CFT のダウンロード

GitHub から **lambda-python-files** フォルダをダウンロードします。このフォルダには、次のファイルが含まれています。

- Lambda レイヤの作成に使用される Python（.py）ファイル。
- 必要に応じて、スタティックルートを追加し、ネットワークパラメータをカスタマイズするために使用される configuration.json ファイル。

GitHub から次の CloudFormation テンプレートをダウンロードします。

- NLB を使用した Auto Scale ソリューションのテンプレート：
  - **Infrastructure.yaml**：AWS 環境のコンポーネントをカスタマイズするために使用されます。
  - **deploy_ngfw_autoscale.yaml**：NLB ソリューションを使用した AWS Auto Scale の展開に使用されます。
- GWLB を使用した Auto Scale ソリューションのテンプレート：
  - **Infrastructure_gwlb.yaml**：AWS 環境のコンポーネントをカスタマイズするために使用されます。
  - **deploy_ngfw_autoscale_with_gwlb.yaml**：GWLB ソリューションを使用して AWS Auto Scale を展開するために使用されます。

（注） 可能な場合は、テンプレートパラメータの値を収集します。収集すると、AWS 管理コンソールでテンプレートを展開するときに、値をすばやく簡単に入力できます。

NLB を使用した Auto Scale ソリューション：Amazon CloudFormation コンソールでの NLB インフラストラクチャテンプレートのカスタマイズと展開

NLB を使用して Auto Scale ソリューションを展開する場合は、この項に記載されている手順を実行します。

ステップ 1 AWS 管理コンソールで、**[サービス（Services）]** > **[管理とガバナンス（Management and Governance）]** > **[CloudFormation]**の順に選択し、**[スタックの作成（Create stack）]** > **[新しいリソースを使用（標準）（With new resources (standard)）]**の順にクリックします。

ステップ 2 [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、ファイルをダウンロードしたフォルダから **infrastructure.yaml** を選択します。

ステップ 3 [次へ（Next）] をクリックします。

ステップ 4 [スタックの詳細の指定（Specify stack details）] ページで、スタックの名前を入力します。

ステップ 5 ***Infrastructure.yaml*** テンプレートの入力パラメータの値を指定します。

ステップ 6 [次へ（Next）] をクリックします。

ステップ 7 [スタックオプションの設定（Configure Stack Options）] ウィンドウで [次へ（Next）] をクリックします。

ステップ 8 [確認（Review）] ページで設定を確認して確定します。

ステップ 9 [スタックの作成（Create Stack）] をクリックして **infrastructure.yaml** テンプレートを展開し、スタックを作成します。

ステップ 10 展開が完了したら、[出力（Outputs）] に移動し、**S3 バケット名**を書き留めます。

GWLB を使用した Auto Scale ソリューション：Amazon CloudFormation コンソールでの GWLB インフラストラクチャ テンプレートのカスタマイズと展開

GWLB を使用して Auto Scale ソリューションを展開する場合は、この項に記載されている手順を実行します。

ステップ 1 AWS 管理コンソールで、**[サービス（Services）]** > **[管理とガバナンス（Management and Governance）]** > **[CloudFormation]**の順に選択し、**[スタックの作成（Create stack）]** > **[新しいリソースを使用（標準）（With new resources (standard)）]**の順にクリックします。

ステップ 2 [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、ファイルをダウンロードしたフォルダから **infrastructure_gwlb.yaml** を選択します。

ステップ 3 [次へ（Next）] をクリックします。

ステップ 4 [スタックの詳細の指定（Specify stack details）] ページで、スタックの名前を入力します。

ステップ 5 ***Infrastructure_gwlb.yaml*** テンプレートの入力パラメータの値を指定します。

ステップ 6 [次へ（Next）] をクリックします。

ステップ 7 [スタックオプションの設定（Configure Stack Options）] ウィンドウで [次へ（Next）] をクリックします。

ステップ 8 [確認（Review）] ページで設定を確認して確定します。

ステップ 9 [スタックの作成（Create Stack）] をクリックして **infrastructure_gwlb.yaml** テンプレートを展開し、スタックを作成します。

ステップ 10 展開が完了したら、[出力（Outputs）] に移動し、**S3 バケット名**を書き留めます。

Management Center でのネットワーク インフラストラクチャの設定

登録済みの Threat Defense Virtual の Management Center で、デバイスグループ、オブジェクト、ヘルスチェックポート、NAT ポリシー、およびアクセスポリシーを作成および設定します。

別のサーバー上で実行されるフル機能のマルチデバイスマネージャである Management Center を使用して Threat Defense Virtual を管理できます。Threat Defense Virtual は、Threat Defense Virtual 仮想マシンに割り当てた管理インターフェイス上の Management Center を登録して通信します。

詳細については、「Cisco Secure Firewall Management Center を備えた Cisco Secure Firewall Threat Defense Virtual について」を参照してください。

Threat Defense Virtual 設定に使用されるオブジェクトはすべて、ユーザーが作成する必要があります。

重要 デバイスグループを作成し、ルールを適用する必要があります。デバイスグループに適用されたすべての設定が Threat Defense Virtual インスタンスにプッシュされます。

デバイスグループの追加

Management Center を使用すると、デバイスをグループ化して、複数のデバイスへのポリシーの展開やアップデートのインストールを簡単に実行できます。グループに属するデバイスのリストは、展開または縮小表示できます。

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

ステップ 2 [追加（Add）] ドロップダウン メニューから、[グループの追加（Add Group）] を選択します。

ステップ 3 既存のグループを編集するには、編集するグループの [編集（Edit）]（編集アイコン）をクリックします。

ステップ 4 名前を入力します。

ステップ 5 [使用可能なデバイス（Available Devices）] から、デバイス グループに追加するデバイスを 1 つ以上選択します。複数のデバイスを選択する場合は、Ctrl または Shift を押しながらクリックします。

ステップ 6 [追加（Add）] をクリックして、選択したデバイスをデバイス グループに追加します。

ステップ 7 [OK] をクリックして、デバイス グループを追加します。

ホストオブジェクトの作成

ステップ 1 Management Center にログインします。

ステップ 2 **[オブジェクト（Objects）]** > **[オブジェクト管理（Object Management）]** を選択します。

ステップ 3 オブジェクト タイプのリストから [ネットワーク（Network）] を選択します。

ステップ 4 [ネットワークを追加（Add Network）] ドロップダウン メニューで、[オブジェクトの追加（Add Object）] を選択します。

ステップ 5 名前を入力します。

ステップ 6 説明を入力します。

ステップ 7 [ネットワーク（Network）] フィールドで [ホスト（Host）] オプションを選択し、次の値を入力します。

a) オブジェクトタイプの名前：**aws-metadata-server**。

b) ホストプロトコルのタイプに応じて、IPv4 の IP アドレス **169.254.169.254** を入力します。

ステップ 8 [保存（Save）] をクリックします。

ポートオブジェクトの作成

ステップ 1 Management Center にログインします。

ステップ 2 **[オブジェクト（Objects）]** > **[オブジェクト管理（Object Management）]** を選択します。

ステップ 3 オブジェクト タイプのリストから [ポート（Port）] を選択します。

ステップ 4 [ポートの追加（Add Port）] ドロップダウンメニューで、[オブジェクトの追加（Add Object）] を選択します。

ステップ 5 名前を入力します。

ステップ 6 [プロトコル（Protocol）] を選択します。[ホスト（Host）] オブジェクトタイプに入力したプロトコルを選択する必要があります。選択したプロトコルに応じて、[ポート（Port）] で制限するか、または ICMP の [タイプ（Type）] および [コード（Code）] を選択します。

ステップ 7 **8080** と入力します。ここで入力するポート番号は、要件に応じてカスタマイズできます。

（注） [すべて（All）] のプロトコルと一致させることを選択した場合は、[その他（Other）] ドロップダウンリストを使用して、ポートでオブジェクトを制限する必要があります。

ステップ 8 [保存（Save）] をクリックします。

セキュリティゾーンおよびインターフェイス グループ オブジェクトの作成

ステップ 1 **[オブジェクト（Objects）]** > **[オブジェクト管理（Object Management）]** を選択します。

ステップ 2 オブジェクトタイプのリストから、[インターフェイス（Interface）] を選択します。

ステップ 3 **[追加（Add）]**>**[セキュリティゾーン（Security Zone）]**の順にクリックするか、**[追加（Add）]**>**[インターフェイスグループ（Interface Group）]**の順にクリックします。

ステップ 4 [名前（Name）] : *inside-sz/outside-sz* を入力します。

ステップ 5 [インターフェイスタイプ（Interface Type）] : [ルーテッド（Routed）] を選択します。

ステップ 6 [保存（Save）] をクリックします。

ヘルスチェックプローブのポートの有効化

ヘルスチェックプローブのポート 22（SSH）またはポート 443（HTTP）を有効にできます。

ヘルスチェックプローブのポート 22（SSH）の有効化

ヘルスチェックプローブにポート 22（SSH）を使用している場合は、次の手順を実行して、ヘルスチェックプローブのポートを有効にします。

ステップ 1 **[デバイス（Devices）]** > **[プラットフォーム設定（Platform Settings）]** > **[SSHアクセス（SSH Access）]**の順に選択します。

ステップ 2 **[+ Add]** をクリックします。

ステップ 3 ドロップダウンリストから関連する [IPアドレス（IP Address）] を選択します。

ステップ 4 [使用可能なゾーン/インターフェイス（Available Zones/Interfaces）] ウィンドウで、GWLB または外部サブネットに接続されている外部インターフェイスを選択します。

ステップ 5 [追加（Add）] をクリックして、選択したインターフェイスを [選択したゾーン/インターフェイス（Selected Zones/Interfaces）] ウィンドウに追加します。

ステップ 6 [OK] をクリックします。

ステップ 7 [保存（Save）] をクリックします。

ヘルスチェックプローブのポート 443（HTTP）の有効化

ヘルスチェックプローブにポート 443（HTTP）を使用している場合は、次の手順を実行して、ヘルスチェックプローブのポートを有効にします。

ステップ 1 **[デバイス（Devices）] > [プラットフォーム設定（Platform Settings）] > [HTTPアクセス（HTTP Access）]** の順に選択します。

ステップ 2 [HTTPサーバーの有効化（Enable HTTP Server）] チェックボックスをオンにします。

ステップ 3 [ポート（Port）] フィールドに、**443** と入力します。

ステップ 4 **[+ Add]** をクリックします。

ステップ 5 ドロップダウンリストから関連する [IPアドレス（IP Address）] を選択します。

ステップ 6 [使用可能なゾーン/インターフェイス（Available Zones/Interfaces）] ウィンドウで、GWLB または外部サブネットに接続されている外部インターフェイスを選択します。

ステップ 7 [追加（Add）] をクリックして、選択したインターフェイスを [選択したゾーン/インターフェイス（Selected Zones/Interfaces）] ウィンドウに追加します。

ステップ 8 [OK] をクリックします。

ステップ 9 [保存（Save）] をクリックします。

NLB を使用した Auto Scale ソリューション：ネットワークアドレス変換（NAT）ポリシーの設定と展開

一般的な NAT ルールでは、内部アドレスを外部インターフェイスの IP アドレスのポートに変換します。このタイプの NAT ルールのことをインターフェイス ポート アドレス変換（PAT）と呼びます。NAT ポリシーの詳細については、「Cisco Secure Firewall Management Center を使用した Cisco Secure Firewall Threat Defense Virtual の管理」の「NAT の設定」を参照してください。

NAT ポリシーには 1 つの必須ルールが必要です。以下に、NAT ルールの例を示します。

- 送信元ゾーン（Source Zone）：外部ゾーン
- 宛先ゾーン（Dest Zone）：内部ゾーン
- 元の送信元（Original-sources）：any-ipv4
- 元の送信元ポート（Original source port）：元/デフォルト
- 元の宛先（Original Destinations）：インターフェイス（Interface）
- 元の宛先ポート（Original-destination-port）：8080 またはユーザーが設定する正常性ポート
- 変換済み送信元（Translated-sources）：any-ipv4
- 変換済み送信元ポート（Translated source port）：元/デフォルト
- 変換済み宛先（Translated-destination）：aws-metadata-server
- 変換済み宛先ポート（Translated-destination-port）：80/HTTP

同様に、この設定が Threat Defense Virtual デバイスにプッシュされるように、データトラフィックの NAT ルールを追加できます。

重要 作成された NAT ポリシーは、デバイスグループに適用する必要があります。これは、Lambda 機能による Management Center 検証によって検証されます。

ステップ 1 Cisco Secure Firewall Management Center にログインします。

ステップ 2 [デバイス（Devices）] メニューで、[NAT] をクリックします。

ステップ 3 新しいポリシーを作成するには、**[新しいポリシー（New Policy）]** > **[Threat Defense NAT]**をクリックします。

ステップ 4 NAT ポリシーの名前と説明を入力します。

ステップ 5 **[保存（Save）]** をクリックします。

新しいポリシーが追加されて、[NAT] ページに表示されます。

ステップ 6 [ルールの追加（Add Rule）] をクリックします。

ステップ 7 [NAT ルール（NAT Rule）] ドロップダウンリストから [手動NAT ルール（Manual NAT Rule）] を選択します。

ステップ 8 [挿入（Insert）] ドロップダウンリストから、[カテゴリ内（In Category）] および [前の NAT ルール（NAT Rule Before）] を選択します。

ステップ 9 [タイプ（Type）] ドロップダウンメニューから [静的（Static）] を選択します。

ステップ 10 説明を入力します。

ステップ 11 [インターフェイスオブジェクト（Interface Objects）] メニューで、送信元と宛先のオブジェクトを追加します。

ステップ 12 [変換（Translations）] メニューで、各パラメータに次の値を追加します。

パラメータ	値
Original Source	any-ipv4
[元の宛先（Original Destination）]	アドレス（Address）
[元の送信元ポート（Original Source Port）]	HTTP
[元の宛先ポート（Original Destination Port）]	8080
Translated Source	any-ipv4
[変換された送信元ポート（Translated Source Port）]	元/デフォルト
[変換済みの宛先（Translated Destination）]	aws-metadata-server
[変換された宛先ポート（Translated Destination Port）]	80/HTTP

ステップ 13 [保存（Save）] をクリックして、ルールを保存して追加します。

ステップ 14 Threat Defense Virtual に展開するために作成した新しいルールを選択します。

ステップ 15 **[展開（Deploy）]**>**[展開（Deployment）]**の順にクリックし、割り当てたデバイスにポリシーを展開します。変更はポリシーを展開するまで有効になりません。

基本的なアクセス コントロール ポリシーの作成

内部から外部へのトラフィックを許可するアクセス制御を設定します。必要なすべてのポリシーを含むアクセスポリシーを作成できます。このポートのトラフィックが到達できるように、正常性ポートオブジェクトを許可する必要があります。アクセスポリシーの詳細については、「Cisco Secure Firewall Management Center を使用した Cisco Secure Firewall Threat Defense Virtual の管理」の「アクセス制御の設定」を参照してください。

新しいアクセス コントロール ポリシーを作成すると、そのポリシーにデフォルトのアクションと設定が含まれます。ポリシーを作成すると、要件に合わせてポリシーを調整できるよう、すぐに編集セッションに移行します。

ステップ 1 **[ポリシー（Policies）]** > **[アクセス制御（Access Control）]** を選択します。

ステップ 2 [新しいポリシー（New Policy）] をクリックします。

ステップ 3 一意の名前と説明を入力します。

ステップ 4 最初の [デフォルトアクション（Default Action）] : [すべてのトラフィックをブロック（Block all traffic）] を指定します。

ステップ 5 **[保存（Save）]** をクリックします。

ステップ 6 作成した新しいポリシーの [編集（Edit）] アイコンをクリックします。

ステップ 7 [ルールの追加（Add Rule）] をクリックします。

ステップ 8 次のパラメータを設定します。

- 名前：inside-to-outside
- 挿入：into Mandatory
- アクション：許可
- 送信元ゾーンと宛先ゾーンを追加します。

ステップ 9 [Apply] をクリックします。

Configuration.json ファイルの更新

configuration.json ファイルは、GitHub からダウンロードした **lambda_python_files** フォルダにあります。Management Center で設定したパラメータを使用して、**configuration.json** ファイルのパラメータを更新します。JSON キーは変更しないでください。

configuration.json ファイル内のスクリプトは次のとおりです。

```
{
  "licenseCaps": ["BASE", "MALWARE", "THREAT"],   //Management center virtual licenses
  "fmcIpforDeviceReg": "DONTRESOLVE",   //Management center virtual IP address
  "RegistrationId": "cisco",  //Registration ID used while configuring the manager in
the Threat defense virtual
  "NatId": "cisco",  //NAT ID used while configuring the manager in the Threat defense
virtual
  "fmcAccessPolicyName": "aws-asg-policy",  //Access policy name configured in the
Management center virtual
  "fmcNatPolicyName": "AWS-Cisco-NGFW-VMs",  //NAT Policy name configured in the Management
 center virtual (Not required for GWLB-based deployment)
  "fmcInsideNicName": "inside", //Threat defense virtual inside interface name
  "fmcOutsideNicName": "outside", //Threat defense virtual outside interface name
  "fmcInsideNic": "GigabitEthernet0/0", //Threat defense virtual inside interface NIC
Name - GigabitEthernet for c4 instance types, and TenGigabitEthernet for c5 instance
types)
  "fmcOutsideNic": "GigabitEthernet0/1", //Threat defense virtual outside interface NIC
 Name - GigabitEthernet for c4 instance types, and TenGigabitEthernet for c5 instance
types
  "fmcOutsideZone": "Outside-sz", //Outside Interface security zone name that is set in
 the Management center virtual
  "fmcInsideZone": "Inside-sz", //Inside Interface security zone name that is set in the
 Management center virtual
  "MetadataServerObjectName": "aws-metadata-server", //Host object name created for the
 IP 169.254.169.254 in the Management center virtual  (Not required for GWLB-based
deployment)
  "interfaceConfig": [
    {
      "managementOnly": "false",
      "MTU": "1500",
      "securityZone": {
        "name": "Inside-sz"
      },
      "mode": "NONE",
      "ifname": "inside",
      "name": "GigabitEthernet0/0"
    },
    {
      "managementOnly": "false",
      "MTU": "1500",
      "securityZone": {
        "name": "Outside-sz"
      },
      "mode": "NONE",
      "ifname": "outside",
      "name": "GigabitEthernet0/1"
    }
  ],  //Interface-related configuration
  "trafficRoutes": [
    {
      "interface": "inside",
      "network": "any-ipv4",
```

```
      "gateway": "",
      "metric": "1"
    }
  ] //This traffic route is used for the Threat defense virtual instance's health check
}
```

このファイルの **trafficRoutes** パラメータを変更することで、Threat Defense Virtual のスタティックルートを設定できます。スタティックルートの設定例を次に示します。

```
{
      "interface": "inside",
      "network": "any-ipv4",
      "gateway": "",
      "metric": "1"
 }
```

AWS CLI を使用したインフラストラクチャ コンポーネントの設定

テンプレートでは、Threat Defense Virtual および Management Center の Lambda レイヤと暗号化されたパスワードは作成されません。次の手順を使用して、各コンポーネントを設定します。AWS CLI の詳細については、「AWS コマンドラインインターフェイス」を参照してください。

コンピューティングリソースを管理するための Lambda レイヤ zip ファイルの作成

Linux ホストに Python フォルダを作成し、Lambda レイヤを作成します。

ステップ 1 Linux ホストに Python フォルダ（Ubuntu 22.04 など）を作成します。

ステップ 2 Linux ホストに Python 3.9 をインストールします。以下に、Python 3.9 をインストールするためのサンプルスクリプトを示します。

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt install python3.9
$ sudo apt install python3-virtualenv
$ sudo apt install zip
$ sudo apt-get install python3.9-distutils
$ sudo apt-get install python3.9-dev
$ sudo apt-get install libffi-dev
```

ステップ 3 Linux 環境で Lambda レイヤ zip ファイル（autoscale_layer.zip）を作成します。このファイルは、Lambda 機能に不可欠な Python ライブラリを提供します。

次のスクリプトを実行して、autoscale_layer.zip ファイルを作成します。

```
#!/bin/bash
mkdir -p layer
mkdir -p python
```

```
virtualenv -p /usr/bin/python3.9 ./layer/
source ./layer/bin/activate
pip3 install attrs==23.1.0
pip3 install bcrypt==3.2.2
pip3 install certifi==2022.12.7
pip3 install cffi==1.15.1
pip3 install chardet==3.0.4
pip3 install cryptography==2.9.1
pip3 install idna==2.10
pip3 install jsonschema==3.2.0
pip3 install paramiko==2.7.1
pip3 install pycparser==2.21
pip3 install pycryptodome==3.15.0
pip3 install PyNaCl==1.5.0
pip3 install pyrsistent==0.19.3
pip3 install requests==2.23.0
pip3 install scp==0.13.2
pip3 install six==1.16.0
pip3 install urllib3==1.25.11
echo "Copy from ./layer directory to ./python\n"
cp -r./layer/lib/python3.9/site-packages/* ./python/
zip -r autoscale_layer.zip ./python
```

ステップ 4 **autoscale_layer.zip** ファイルを作成したら、GitHub からダウンロードした **lambda-python-files** フォルダに **autoscale_layer.zip** ファイルをコピーします。

（任意）Threat Defense Virtual および Management Center の暗号化パスワードの作成

Infrastructure_gwlb.yaml テンプレートファイルに KMS ARN 値が入力されている場合は、Threat Defense Virtual および Management Center で設定するパスワードを暗号化する必要があります。AWS KMS コンソールを使用してキー ARN を特定するには、Finding the key ID and key ARN [英語] を参照してください。ローカルホストで、次の AWS CLI コマンドを実行してパスワードを暗号化します。

```
$ aws kms encrypt --key-id <KMS-ARN> --plaintext
'MyC0mplIc@tedProtect1oN'
{
    "KeyId": "KMS-ARN",
    "CiphertextBlob":
"AQICAHgcQFAGtz/hvaxMtJvY/x/rfHnKI3clFPpSXUU7HQRnCAFwfXhXH
JAHL8tcVmDqurALAAAAajBoBgkqhki
G9w0BBwagWzBZAgEAMFQGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQM45
AIkTqjSekX2mniAgEQgCcOav6Hhol
+wxpWKtXY4y1Z1d0z1P4fx0jTdosfCbPnUExmNJ4zdx8="
}
$
```

「CiphertextBlob」の値は暗号化されたパスワードです。このパスワードは、**infrastructure_gwlb.yaml** ファイルの **NGFWv パスワード**（Threat Defense Virtual パスワード）または **Auto Scale 自動化の FMC パスワード**（Management Center のパスワード）パラメータの値として使用します。このパスワードは、**CloudWatch にメトリックを公開するための FMC パスワード**の値としても使用できます。

target フォルダの作成

ローカルホストで、次のコマンドを使用して、Amazon S3 バケットにアップロードする必要があるファイルを含む target フォルダを作成します。

python3 make.py build

ローカルホストに「target」という名前のフォルダが作成されます。target フォルダには、Auto Scale ソリューションの展開に必要な *zip* ファイルと *yaml* ファイルが含まれています。

Amazon S3 バケットへのファイルのアップロード

ローカルホストで、次のコマンドを使用して、target ディレクトリにあるすべてのファイルを Amazon S3 バケットにアップロードします。

$ cd ./target

$ aws s3 cp . s3://<*bucket-name*> **--recursive**

NLB を使用した Auto Scale ソリューション：NLB を使用した Auto Scale ソリューションの展開

NLB を使用して Auto Scale ソリューションを展開する場合は、この項に記載されている手順を実行します。

ステップ 1 AWS 管理コンソールで、**[サービス（Services）]** > **[管理とガバナンス（Management and Governance）]** > **[CloudFormation]** > **[スタック（Stacks）]**の順に選択し、テンプレートによって作成されたスタックをクリックします。

ステップ 2 **[スタックの作成（Create stack）]** > **[新しいリソースを使用（標準）（With new resources (standard)）]**の順にクリックします。

ステップ 3 [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、target フォルダから ***deploy_ngfw_autoscale.yaml*** を選択します。

ステップ 4 [次へ（Next）] をクリックします。

ステップ 5 [スタックの詳細の指定（Specify stack details）] ページで、スタックの名前を入力します。

ステップ 6 ***deploy_ngfw_autoscale.yaml*** テンプレートの入力パラメータの値を指定します。

ステップ 7 [スタックオプションの設定（Configure Stack Options）] ウィンドウで [次へ（Next）] をクリックします。

ステップ 8 [確認（Review）] ページで設定を確認して確定します。

ステップ 9 [スタックの作成（Create Stack）] をクリックして ***deploy_ngfw_autoscale.yaml*** テンプレートを展開し、スタックを作成します。

これで、NLB を使用した Threat Defense Virtual の Auto Scale ソリューションを設定するために必要な両方のテンプレートの展開が完了しました。

GWLB を使用した Auto Scale ソリューション：GWLB を使用した Auto Scale ソリューションの展開

GWLB を使用して Auto Scale ソリューションを展開する場合は、この項に記載されている手順を実行します。

ステップ 1 AWS 管理コンソールで、**[サービス（Services）]** > **[管理とガバナンス（Management and Governance）]** > **[CloudFormation]** > **[スタック（Stacks）]**の順に選択し、テンプレートによって作成されたスタックをクリックします。

ステップ 2 **[スタックの作成（Create stack）]** > **[新しいリソースを使用（標準）（With new resources (standard)）]**の順にクリックします。

ステップ 3 [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、target フォルダから ***deploy_ngfw_autoscale_with_gwlb.yaml*** を選択します。

ステップ 4 [次へ（Next）] をクリックします。

ステップ 5 [スタックの詳細の指定（Specify stack details）] ページで、スタックの名前を入力します。

ステップ 6 ***deploy_ngfw_autoscale_with_gwlb.yaml*** テンプレートの入力パラメータの値を指定します。

ステップ 7 [スタックオプションの設定（Configure Stack Options）] ウィンドウで [次へ（Next）] をクリックします。

ステップ 8 [確認（Review）] ページで設定を確認して確定します。

ステップ 9 [スタックの作成（Create Stack）] をクリックして ***deploy_ngfw_autoscale_with_gwlb.yaml*** テンプレートを展開し、スタックを作成します。

これで、GWLB を使用して Threat Defense Virtual 用の Auto Scale ソリューションを設定するために必要な両方のテンプレートの展開が完了しました。

GWLB ソリューションを使用した Auto Scale：GWLB エンドポイントの作成

GWLB を使用して Auto Scale ソリューションを展開する場合は、この項に記載されている手順を実行します。

ステップ 1 AWS 管理コンソールで、**[サービス（Services）]** > **[ネットワーキングおよびコンテンツ配信（Networking & Content Delivery）]** > **[VPC]** > **[エンドポイントサービス（Endpoint Services）]**の順に選択します。

ステップ 2 [エンドポイントサービスの作成（Create Endpoint Services）] をクリックします。

ステップ 3 [ロードバランサタイプ（Load balancer type）] で [ゲートウェイ（Gateway）] を選択します。

ステップ 4 [使用可能なロードバランサ（Available load Balancers）] で、Auto Scale の展開の一部として作成されたゲートウェイロードバランサを選択します。

ステップ 5 [エンドポイントの承認が必要（Require accept for endpoint）] で、[承認が必要（Acceptance required）] を選択します。選択すると、エンドポイントサービスの接続要求を手動で受け入れる必要があります。

ステップ 6 [サポートされているIPアドレスタイプ（Supported IP address types）] で、[IPv4]を選択します。

ステップ 7 [作成（Create）] をクリックします。

ステップ 8 新たに作成したエンドポイントサービスのサービス名をコピーします。

ステップ 9 **[サービス（Services）] > [ネットワーキングおよびコンテンツ配信（Networking & Content Delivery）] > [VPC] > [エンドポイント（Endpoints）]**の順に選択します。

ステップ 10 [エンドポイントの作成（Create endpoint）] をクリックします。

ステップ 11 [サービスカテゴリ（Service category）] で [その他のエンドポイントサービス（Other endpoint services）] を選択します。

ステップ 12 [サービス名（Service name）] にサービスの名前を入力し、[サービスの確認（ Verify service）] を選択します。

ステップ 13 [VPC] フィールドで、エンドポイントを作成する VPC を選択します。

ステップ 14 [サブネット（Subnets）] で、エンドポイントを作成するサブネットを選択します。

ステップ 15 [IPアドレスタイプ（IP address type）] で、[IPv4] オプションを選択して、エンドポイント ネットワークインターフェイスに IPv4 アドレスを割り当てます。

ステップ 16 [エンドポイントの作成（Create endpoint）] をクリックします。

VPC のルーティングの設定

ステップ 1 AWS 管理コンソールで、**[サービス（Services）] > [ネットワーキングおよびコンテンツ（Networking & Content）] > [仮想プライベートクラウド（Virtual Private Cloud）] > [ルートテーブル（Route tables）]**の順に選択します。

ステップ 2 インターネットゲートウェイのルートテーブルを選択し、次の手順を実行します。

1. **[アクション（Actions）] > [ルートの編集（Edit routes）]**の順にクリックします。
2. IPv4 の場合は、[ルートの追加（Add route）] をクリックします。[宛先（Destination）] に、アプリケーションサーバーのサブネットの IPv4 CIDR ブロックを入力します。[ターゲット（Target）] で、VPC エンドポイントを選択します。
3. [変更の保存（Save Changes）] をクリックします。

ステップ 3 アプリケーションサーバーがあるサブネットのルートテーブルを選択し、次の手順を実行します。

1. **[アクション（Actions）] > [ルートの編集（Edit routes）]**の順にクリックします。
2. IPv4 の場合は、[ルートの追加（Add route）] をクリックします。[宛先（Destination）] に、**0.0.0.0/0** と入力します。[ターゲット（Target）] で、VPC エンドポイントを選択します。

3. [変更の保存（Save Changes）] をクリックします。

ステップ 4 ゲートウェイロードバランサのエンドポイントがあるサブネットのルートテーブルを選択し、次の手順を実行します。

1. **[アクション（Actions）] > [ルートの編集（Edit routes）]**の順にクリックします。
2. IPv4 の場合は、[ルートの追加（Add route）] をクリックします。[宛先（Destination）] に、**0.0.0.0/0** と入力します。[ターゲット（Target）] で、インターネットゲートウェイを選択します。
3. [変更の保存（Save Changes）] をクリックします。

Auto Scale グループの編集

デフォルトでは、Auto Scale グループの Threat Defense Virtual インスタンスの最小数と最大数はそれぞれ 0 と 2 に設定されています。要件に応じて各値を変更します。

ステップ 1 AWS 管理コンソールで、**[サービス（Services）] > [コンピューティング（Compute）] > [EC2]**の順に選択し、[Auto Scalingグループ（Auto Scaling Groups）] をクリックします。

ステップ 2 作成したAuto Scalingグループを選択し、[編集（Edit）]をクリックして、要件に応じて[必要な容量（Desired capacity）]、[最小容量（Minimum capacity）]、[最大容量（Maximum capacity）] フィールドの値を変更します。各値は、Auto Scaling 機能のために起動する Threat Defense Virtual インスタンスの数に対応します。[必要な容量（Desired capacity）] を、最小容量値と最大容量値の範囲内の値に設定します。

ステップ 3 **[更新（Update）]**をクリックします。

（注） Threat Defense Virtual インスタンスを 1 つだけ起動し、そのインスタンスが想定どおりに動作しているか確認することを推奨します。その後、要件に応じて追加のインスタンスを起動できます。

展開の検証

テンプレートの展開が成功したら、Amazon CloudWatch コンソールに移動して、ログが収集され、必要なアラームが作成されていることを確認します。

ログ

ログファイルを確認して、Management Center の接続に関する問題をトラブルシューティングします。

ステップ 1 **AWS 管理**コンソールで、**[サービス（Services）] > [管理とガバナンス（Management and Governance）] > [CloudWatch]**の順に選択します。

ステップ 2 [ロググループ（Log groups）] をクリックし、表示されているいずれかのロググループをクリックしてログを表示します。

アラーム

必要なアラームが Amazon CloudWatch コンソールで作成されていることを確認します。

ステップ 1 **AWS 管理**コンソールで、**[サービス（Services）] > [管理とガバナンス（Management and Governance）] > [CloudWatch]**の順に選択します。

ステップ 2 **[アラーム（Alarms）] > [すべてのアラーム（All Alarms）]**の順にクリックして、スケールアウトおよびスケールイン機能をトリガーする条件とともにアラームのリストを表示します。

メンテナンス タスク

スケーリングプロセス

このトピックでは、Auto Scale グループの 1 つ以上のスケーリングプロセスを一時停止してから再開する方法について説明します。

スケールアクションの開始と停止

スケールアクションを開始および停止するには、次の手順を実行します。

- AWS 動的スケーリングの場合：スケールアウトアクションを有効化または無効化する方法については、次のリンクを参照してください。

  スケーリングプロセスの一時停止と再開

ヘルスモニター

60 分ごとに、CloudWatch Cron ジョブは、Health Doctor モジュールの Auto Scale Manager Lambda をトリガーします。

- 有効な Threat Defense Virtual VM に属する異常な IP がある場合、Threat Defense Virtual の展開時間が 1 時間を超えると、そのインスタンスは削除されます。
- それらの IP が有効な Threat Defense Virtual マシン の IP ではない場合、IP だけがターゲットグループから削除されます。

ヘルスモニターは、デバイスグループ、アクセスポリシー、および NAT ルールの Management Center 構成も検証します。IP やインスタンスが正常でない場合、または Management Center の検証が失敗した場合、ヘルスモニターはユーザーに電子メールを送信します。

ヘルスモニターの無効化

ヘルスモニターを無効にするには、***constant.py*** で constant を「True」に設定します。

ヘルスモニターの有効化

ヘルスモニターを有効にするには、***constant.py*** で固定値を「False」に設定します。

ライフサイクルフックの無効化

まれに、ライフサイクルフックを無効にする必要があります。無効にすると、インスタンスに追加のインターフェイスが追加されません。また、Threat Defense Virtual インスタンスの展開に連続して失敗することがあります。

Auto Scale Manager の無効化

Auto Scale Manager を無効化するには、それぞれの CloudWatch イベント「notify-instance-launch」と「notify-instance-terminate」を無効化する必要があります。これらのイベントを無効にしても、新しいイベントの Lambda はトリガーされません。ただし、すでに実行されている Lambda アクションは続行されます。Auto Scale Manager が突然停止することはありません。スタックの削除またはリソースの削除による突然の停止を試みると、不定状態になる可能性があります。

ロードバランサのターゲット

AWS ロードバランサでは、複数のネットワーク インターフェイスを持つインスタンスに対してインスタンスタイプのターゲットが許可されないため、Gigabit0/1 インターフェイス IP はターゲットグループのターゲットとして設定されます。ただし、現在のところ、AWS Auto Scale のヘルスチェックは、IP ではなく、インスタンスタイプのターゲットに対してのみ機能します。また、これらの IP はターゲットグループから自動的に追加されたり、削除されたりしません。したがって、Auto Scale ソリューションは、これら両方のタスクをプログラムで処理します。ただし、メンテナンスやトラブルシューティングの場合は、手動で実行する必要があることがあります。

ターゲットグループへのターゲットの登録

Threat Defense Virtual インスタンスをロードバランサに登録するには、Gigabit0/1 インスタンス IP（外部サブネット）をターゲットとしてターゲットグループに追加する必要があります。「IP アドレスによるターゲットの登録または登録解除」を参照してください。

ターゲットグループからのターゲットの登録解除

ロードバランサに対する Threat Defense Virtual インスタンスの登録を解除するには、Gigabit0/1 インスタンス IP（外部サブネット）をターゲットグループのターゲットとして削除する必要があります。「IP アドレスによるターゲットの登録または登録解除」を参照してください。

インスタンスのスタンバイ

AWS では、Auto Scale グループでのインスタンスの再起動は許可されませんが、ユーザーはインスタンスをスタンバイ状態にして再起動アクションを実行できます。これは、ロードバランサのターゲットがインスタンスタイプの場合に最も機能しますが、Threat Defense Virtual マシンは、複数のネットワークインターフェイスがあるため、インスタンスタイプのターゲットとして設定できません。

インスタンスをスタンバイ状態にする

インスタンスがスタンバイ状態になると、正常性プローブが失敗するまで、ターゲットグループ内のそのインスタンスの IP は同じ状態のままになります。このため、インスタンスをスタンバイ状態にする前に、ターゲットグループからそれぞれの IP を登録解除することをお勧めします。詳細については、ロードバランサのターゲット （46 ページ）を参照してください。

IP が削除されたら、「Auto Scaling グループからのインスタンスの一時的な削除」を参照してください。

スタンバイ状態からのインスタンスの削除

同様に、インスタンスをスタンバイ状態から実行状態に移行できます。スタンバイ状態から削除すると、インスタンスの IP がターゲットグループのターゲットに登録されます。「ロードバランサのターゲット （46 ページ）」を参照してください。

トラブルシューティングやメンテナンスのためにインスタンスをスタンバイ状態にする方法の詳細については、AWS News Blog を参照してください。

Auto Scale グループからのインスタンスの削除または分離

Auto Scale グループからインスタンスを削除するには、まずインスタンスをスタンバイ状態に移行する必要があります。「インスタンスをスタンバイ状態にする」を参照してください。スタンバイ状態になったインスタンスは、削除または分離できます。「Auto Scaling グループから EC2 インスタンスをデタッチする」を参照してください。

Management Center 側に変更はありません。必要な変更は手動で実行する必要があります。

インスタンスで終了

インスタンスを終了するには、スタンバイ状態にする必要があります。インスタンスのスタンバイ （47 ページ）を参照してください。インスタンスがスタンバイ状態になったら、終了できます。

インスタンスのスケールイン保護

Auto Scale グループから特定のインスタンスが誤って削除されないようにするために、そのインスタンスをスケールイン保護として作成できます。インスタンスがスケールイン保護されている場合、スケールインイベントが原因で終了することはありません。

インスタンスをスケールイン保護状態にするには、次のリンクを参照してください。

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-termination.html

重要　正常（EC2 インスタンスだけでなく、ターゲット IP が正常）なインスタンスの最小数をスケールイン保護として設定することをお勧めします。

設定の変更

設定の変更は、すでに実行中のインスタンスには自動的に反映されません。変更は新しいデバイスにのみ反映されます。このような変更は、既存のデバイスに手動でプッシュする必要があります。

既存のインスタンスの設定を手動で更新しているときに問題が発生した場合は、それらのインスタンスをスケーリンググループから削除し、新しいインスタンスに置き換えることを推奨します。

Management Center のユーザー名とパスワードの変更

Management Center の IP、ユーザー名、またはパスワードを変更する場合は、Auto Scale Manager Lambda 関数とカスタム指標パブリッシャ Lambda 関数の環境変数でそれぞれの変更を実行する必要があります。「AWS Lambda 環境変数の使用」を参照してください。

Lambda の次回実行時に、変更された環境変数が参照されます。

（注）　環境変数は Lambda 関数に直接渡されます。パスワードの複雑さはチェックされません。

Threat Defense Virtual の管理者パスワードを変更します。

Threat Defense Virtual パスワードを変更すると、インスタンスを実行するために各デバイスでパスワードを手動で変更する必要があります。新しい Threat Defense Virtual デバイスをオンボー

ドする場合、Threat Defense Virtual パスワードは Lambda 環境変数から取得されます。「AWS Lambda 環境変数の使用」を参照してください。

登録 ID と NAT ID の変更

新しい Threat Defense Virtual デバイスを異なる登録 ID と NAT ID でオンボードする場合、Management Center 登録のために、Configuration.json ファイルでこの情報を変更する必要があります。Configuration.json ファイルは、[Lambda] リソースページにあります。

アクセスポリシーと NAT ポリシーの変更

アクセスポリシーまたは NAT ポリシーへの変更は、デバイスグループの割り当てにより、今後のインスタンスに自動的に適用されます。ただし、既存の Threat Defense Virtual インスタンスを更新するには、設定変更を手動でプッシュして、Management Center から展開する必要があります。

AWS リソースに対する変更

AWS の導入後、Auto Scale グループ、起動設定、CloudWatch イベント、スケーリングポリシーなど、多くの項目を変更できます。CloudFormation スタックにリソースをインポートするか、既存のリソースから新しいスタックを作成できます。

AWS リソースで実行される変更を管理する方法の詳細については、「既存リソースの CloudFormation 管理への取り込み」を参照してください。

CloudWatch ログの収集および分析

CloudWatch ログをエクスポートするには、「AWS CLI を使用した Amazon S3 へのログデータのエクスポート」を参照してください。

トラブルシューティング

AWS CloudFormation コンソール

AWS CloudFormation コンソールで CloudFormation スタックへの入力パラメータを確認できます。これにより、Web ブラウザからスタックを直接作成、監視、更新、削除できます。

目的のスタックに移動し、[パラメータ（parameter）] タブを確認します。[Lambda関数環境変数（Lambda Functions environment variables）] タブで Lambda 関数への入力を確認することもできます。*configuration.json* ファイルは、Auto Scale Manager Lambda 関数自体でも表示できます。

AWS CloudFormation コンソールの詳細については、『AWS CloudFormation ユーザーガイド（AWS CloudFormation User Guide）』を参照してください。

Amazon CloudWatch ログ

個々の Lambda 関数のログを表示できます。AWS Lambda はお客様の代わりに Lambda 関数を自動的に監視し、Amazon CloudWatch を通じてメトリックを報告します。関数の障害のトラブルシューティングに役立つように、Lambda は関数によって処理されたすべての要求をログに記録し、Amazon CloudWatch ログを通じてコードによって生成されたログも自動的に保存します。

Lambda コンソール、CloudWatch コンソール、AWS CLI、または CloudWatch API を使用して、Lambda のログを表示できます。ロググループと CloudWatch コンソールを介したロググループへのアクセスの詳細については、『*Amazon CloudWatch* ユーザーガイド（*Amazon CloudWatch User Guide*）』でモニターリングシステム、アプリケーション、およびカスタムログファイルについて参照してください。

ロードバランサのヘルスチェックの失敗

ロードバランサのヘルスチェックには、プロトコル、ping ポート、ping パス、応答タイムアウト、ヘルスチェック間隔などの情報が含まれます。ヘルスチェック間隔内に 200 応答コードを返す場合、インスタンスは正常と見なされます。

一部またはすべてのインスタンスの現在の状態が `OutOfService` であり、説明フィールドに「インスタンスがヘルスチェックの異常しきい値の数以上連続して失敗しました（Instance has failed at least the Unhealthy Threshold number of health checks consecutively）」というメッセージが表示された場合、インスタンスはロードバランサのヘルスチェックに失敗しています。

Management Center 構成の正常性プローブ NAT ルールを確認する必要があります。詳細については、『Troubleshoot a Classic Load Balancer: Health checks』を参照してください。

トラフィックの問題

Threat Defense Virtual インスタンスのトラフィックの問題をトラブルシューティングするには、ロードバランサルール、NAT ルール、および Threat Defense Virtual インスタンスで設定されているスタティックルートを確認する必要があります。

セキュリティグループのルールなど、展開テンプレートで提供される AWS 仮想ネットワーク/サブネット/ゲートウェイの詳細も確認する必要があります。たとえば、「EC2 インスタンスのトラブルシューティング（Troubleshooting EC2 instances）」https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-troubleshoot.htmlなど、AWS のドキュメントを参照することもできます。

Management Center への接続に失敗

管理接続が中断された場合は、設定とログイン情報を確認する必要があります。『*Firepower Management Center Configuration Guide*』の「Requirements and Prerequisites for Device Management」を参照してください。

デバイスが FMC への登録に失敗 Management Center

デバイスが Management Center に登録できない場合は、Management Center 構成に障害があるか到達不能であるか、または Management Center に新しいデバイスを収容するキャパシティがあ

るかどうかを判断する必要があります。『*Firepower Management Center Configuration Guide*』の「Add a Device to the FMC」を参照してください。

Threat Defense Virtual に SSH 接続できない

Threat Defense Virtual に SSH 接続できない場合は、テンプレートを介して複雑なパスワードが Threat Defense Virtual に渡されたかどうかを確認します。

導入例：AWS で GWLB を使用して North-South トラフィックを検査する Threat Defense Virtual の Auto Scale ソリューション

これは、AWS 環境でゲートウェイロードバランサ（GWLB）を使用して Threat Defense Virtual インスタンスの Auto Scaling を設定し、North-South トラフィックを検査する方法を説明するユースケースドキュメントです。

AWS で GWLB を使用して North-South トラフィックを検査する Threat Defense Virtual Auto Scale ソリューションの設定方法

Auto Scale ソリューションを使用すると、トラフィック検査用にホストされている Threat Defense Virtual インスタンスのグループの展開、スケーリング、および管理ができます。トラフィックは、パフォーマンスまたは使用容量に応じて、単一または複数の Threat Defense Virtual インスタンスに分散されます。

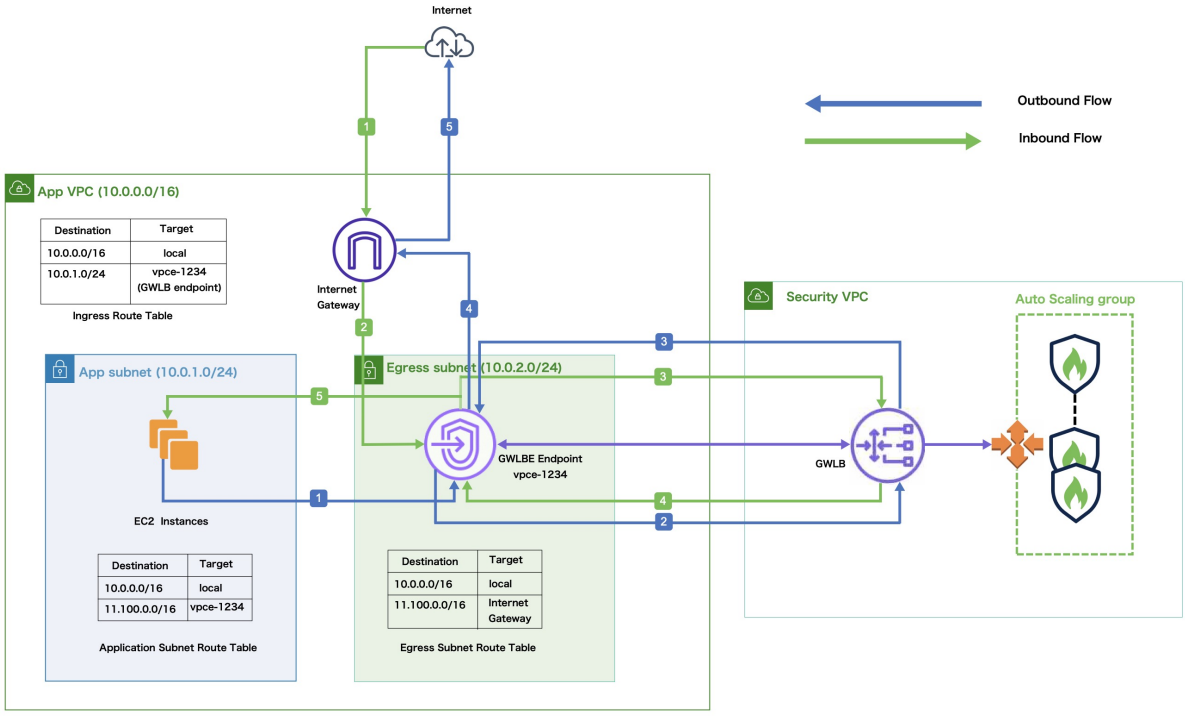
GWLB は、内部および外部で生成されたトラフィックを管理する単一のエントリおよびエグジットポイントとして機能し、トラフィック負荷に基づいて Threat Defense Virtual インスタンスの数をリアルタイムでスケールアップまたはスケールダウンします。

（注） この導入例で使用されているパラメータ値はサンプル値です。要件に応じて各値を変更します。

トポロジの例

このサンプルトポロジは、インバウンドおよびアウトバウンドのネットワーク トラフィック フローが GWLB を介して Threat Defense Virtual インスタンスに分散され、アプリケーション VPC にルーティングされてから、逆方向にルーティングされる方法を示しています。

図 *3 : GWLB* を使用した *Threat Defense Virtual Auto Scale* ソリューション



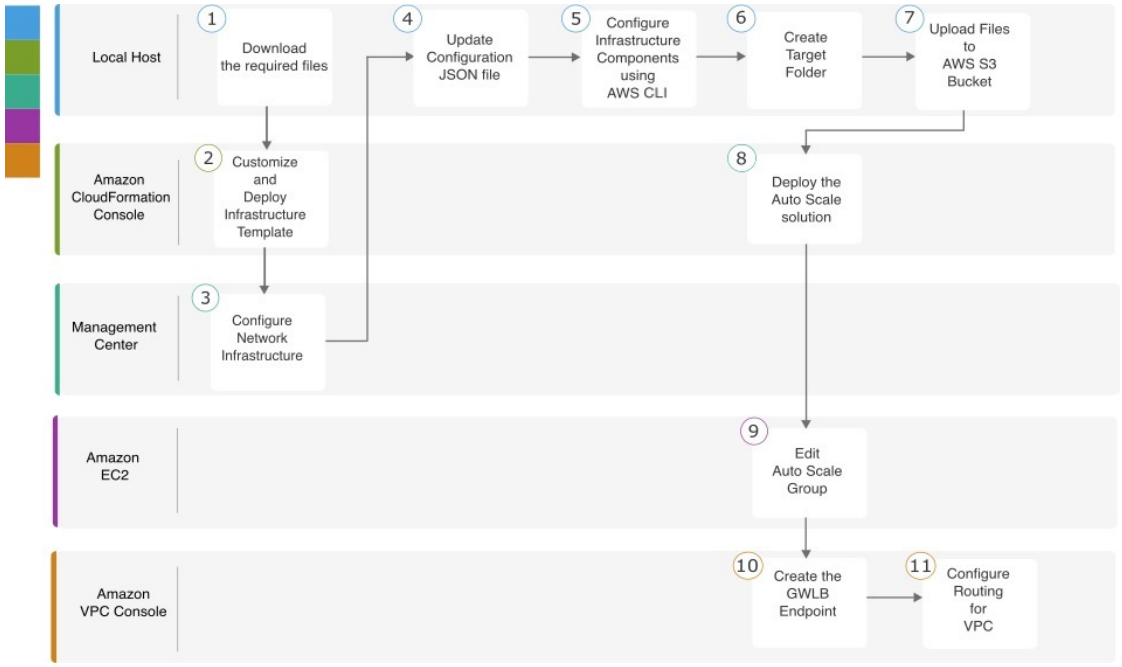
インバウンドトラフィック検査	
1	インターネットゲートウェイ（IGW）が、インターネットからトラフィックを受信します。
2	トラフィックが、入力ルートテーブルのルートに従ってゲートウェイロードバランサのエンドポイント（GWLBe）にルーティングされます。
3	GWLBe が、Security Virtual Private Cloud（VPC）のエンドポイントサービスに接続されます。GWLB が受信したトラフィックをカプセル化し、検査のために Threat Defense Virtual Auto Scaling グループに転送します。
4	Auto Scaling グループによって検査されたトラフィックが GWLB に返されてから GWLB エンドポイントに返されます。
5	GWLBエンドポイントが、アプリケーションサブネット内のリソースにルーティングされるアプリケーション VPC にトラフィックを転送します。

アウトバウンドトラフィック検査	
1	アプリケーションサブネットリソースからのトラフィックが、同じVPC内のGWLBeにルーティングされます。

アウトバウンドトラフィック検査	
2	GWLBe が、セキュリティ VPC のエンドポイントサービスに接続されます。GWLB が受信したトラフィックをカプセル化し、検査のために Auto Scaling グループに転送します。
3	Auto Scaling グループによって検査されたトラフィックが GWLB に返されてから GWLBe に返されます。
4	送信元 VPC に到着したトラフィックが、出力サブネットルートテーブルで定義されたルートに従って IGW に転送されます。
5	IGW がトラフィックをインターネットに送信します。

エンドツーエンドの手順

次のフローチャートは、Amazon Web Services（AWS）に GWLB を使用して Threat Defense Virtual Auto Scale ソリューションを展開する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	前提条件
2	Amazon CloudFormation コンソール	Amazon CloudFormation コンソール：インフラストラクチャ テンプレートのカスタマイズと展開

	ワークスペース	手順
3	Management Center	Management Center：Threat Defense Virtual の Management Center でのネットワーク インフラストラクチャの設定
4	ローカルホスト	ローカルホスト：設定 JSON ファイルの更新
5	ローカルホスト	ローカルホスト：ローカルホストでの AWS CLI を使用したインフラストラクチャ コンポーネントの設定
6	ローカルホスト	ローカルホスト：target フォルダの作成
7	ローカルホスト	ローカルホスト：Amazon S3 バケットへの AWS GWLB Auto Scale ソリューション展開ファイルのアップロード
8	Amazon CloudFormation コンソール	Amazon CloudFormation コンソール：GWLB を使用して Threat Defense Virtual の Auto Scale ソリューションを展開する
9	Amazon EC2 コンソール	Amazon EC2 コンソール：Auto Scale グループのインスタンス数の編集
10	Amazon VPC コンソール	GWLB エンドポイントの作成
11	Amazon VPC コンソール	カスタマー VPC のルーティングの設定

前提条件

- GitHub から **lambda-python-files** フォルダをダウンロードします。このフォルダには、次のファイルが含まれています。
  - Lambda レイヤの作成に使用される Python（.py）ファイル。
  - 必要に応じて、スタティックルートを追加し、ネットワークパラメータをカスタマイズするために使用される configuration.json ファイル。
- GitHub から次の CloudFormation テンプレートをダウンロードします。
  - **Infrastructure_gwlb.yaml**：AWS 環境のコンポーネントをカスタマイズするために使用されます。
  - **deploy_ngfw_autoscale_with_gwlb.yaml**：GWLB ソリューションを使用して AWS Auto Scale を展開するために使用されます。
- （任意）可能な場合は、テンプレートパラメータの値を収集します。収集すると、AWS 管理コンソールでテンプレートを展開するときに、値をすばやく簡単に入力できます。

Amazon CloudFormation コンソール：インフラストラクチャ テンプレートのカスタマイズと展開

インフラストラクチャテンプレートをカスタマイズして展開するには、この項に記載されている手順を実行します。

ステップ 1 AWS 管理コンソールで、**[サービス（Services）]** > **[管理とガバナンス（Management and Governance）]** > **[CloudFormation]**の順に選択し、**[スタックの作成（Create stack）]** > **[新しいリソースを使用（標準）（With new resources (standard)）]**の順にクリックします。

ステップ 2 [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、ファイルをダウンロードしたフォルダから **infrastructure_gwlb.yaml** を選択します。

ステップ 3 [次へ（Next）] をクリックします。

ステップ 4 [スタックの詳細の指定（Specify stack details）] ページで、スタックの名前を入力します。

ステップ 5 ***Infrastructure_gwlb.yaml*** テンプレートの入力パラメータの値を指定します。

パラメータ	値
ポッドの設定	
ポッド名	*infrastructure*
ポッド番号	1
S3 バケット名	demo-us-bkt
VPC CIDR	20.0.0.0/16
可用性ゾーンの数	2
ListOfAzs（可用性ゾーンのリスト）	us-west-1a,us-west-1b
管理サブネットの名前	MgmtSubnet-1,MgmtSubnet-2
MgmtSubnetCidrs	20.1.250.0/24,20.1.251.0/24
内部サブネットの名前	InsideSubnet-1,InsideSubnet-2
InsideSubnetCidrs	20.1.100.0/24,20.1.101.0/24
外部サブネットの名前	OutsideSubnet-1,OutsideSubnet-2
OutsideSubnetCidrs	20.1.200.0/24,20.1.201.0/24
Lambda サブネットの名前	LambdaSubnet-1,LambdaSubnet-2
Lambda サブネット CIDR	20.1.50.0/24,20.1.51.0/24

ステップ 6 [次へ（Next）] をクリックします。

ステップ 7 [スタックオプションの設定（Configure Stack Options）] ウィンドウで [次へ（Next）] をクリックします。

ステップ 8 [確認（Review）] ページで設定を確認して確定します。

ステップ 9 [スタックの作成（Create Stack）] をクリックして **infrastructure_gwlb.yaml** テンプレートを展開し、スタックを作成します。

ステップ 10 展開が完了したら、[出力（Outputs）] に移動し、**S3 バケット名**を書き留めます。

Management Center：Threat Defense Virtual の Management Center でのネットワーク インフラストラクチャの設定

登録済み Threat Defense Virtual の Management Center で、オブジェクト、デバイスグループ、ヘルスチェックポート、およびアクセスポリシーを作成および設定します。

ホストオブジェクトの作成

ステップ 1 Management Center にログインします。

ステップ 2 **[オブジェクト（Objects）]** > **[オブジェクト管理（Object Management）]** を選択します。

ステップ 3 オブジェクト タイプのリストから [ネットワーク（Network）] を選択します。

ステップ 4 [ネットワークを追加（Add Network）] ドロップダウン メニューで、[オブジェクトの追加（Add Object）] を選択します。

ステップ 5 [名前（Name）]：*aws-metadata-server* と入力します。

ステップ 6 説明を入力します。

ステップ 7 [ネットワーク（Network）] フィールドで [ホスト（Host）] オプションを選択し、IPv4 アドレス：*169.254.169.254* を入力します。

ステップ 8 [保存（Save）] をクリックします。

ポートオブジェクトの作成

ステップ 1 Management Center にログインします。

ステップ 2 **[オブジェクト（Objects）]** > **[オブジェクト管理（Object Management）]** を選択します。

ステップ 3 オブジェクト タイプのリストから [ポート（Port）] を選択します。

ステップ 4 [ポートの追加（Add Port）] ドロップダウンメニューで、[オブジェクトの追加（Add Object）] を選択します。

ステップ 5 [名前（Name）]：*test-port-object* を入力します。

ステップ 6 [プロトコル（Protocol）] を選択します。[ホスト（Host）] オブジェクトタイプに入力したプロトコルを選択する必要があります。選択したプロトコルに応じて、[ポート（Port）] で制限します。

ステップ 7 *8080* と入力します。ここで入力するポート番号は、要件に応じてカスタマイズできます。

（注） [すべて（All）] のプロトコルと一致させることを選択した場合は、[その他（Other）] ドロップダウン リストを使用して、ポートでオブジェクトを制限する必要があります。

ステップ 8 [保存（Save）] をクリックします。

セキュリティゾーンおよびインターフェイス グループ オブジェクトの作成

ステップ 1 **[オブジェクト（Objects）] > [オブジェクト管理（Object Management）]** を選択します。

ステップ 2 オブジェクトタイプのリストから、[インターフェイス（Interface）] を選択します。

ステップ 3 **[追加（Add）] > [セキュリティゾーン（Security Zone）]** の順にクリックするか、**[追加（Add）] > [インターフェイスグループ（Interface Group）]** の順にクリックします。

ステップ 4 [名前（Name）] : *inside-sz/outside-sz* を入力します。

ステップ 5 [インターフェイスタイプ（Interface Type）] : [ルーテッド（Routed）] を選択します。

ステップ 6 [保存（Save）] をクリックします。

デバイスグループの追加

Management Center を使用すると、デバイスをグループ化して、複数のデバイスへのポリシーの展開やアップデートのインストールを簡単に実行できます。グループに属するデバイスのリストは、展開または縮小表示できます。

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

ステップ 2 [追加（Add）] ドロップダウン メニューから、[グループの追加（Add Group）] を選択します。

ステップ 3 既存のグループを編集するには、編集するグループの [編集（Edit）]（編集アイコン）をクリックします。

ステップ 4 [名前（Name）] に aws-ngfw-autoscale-dg と入力します。

ステップ 5 [使用可能なデバイス（Available Devices）] から、デバイス グループに追加するデバイスを 1 つ以上選択します。複数のデバイスを選択する場合は、Ctrl または Shift を押しながらクリックします。

ステップ 6 [追加（Add）] をクリックして、選択したデバイスをデバイス グループに追加します。

ステップ 7 [OK] をクリックして、デバイス グループを追加します。

ヘルスチェックプローブのポート 443（HTTP）の有効化

ヘルスチェックプローブにポート 443（HTTP）を使用している場合は、次の手順を実行して、ヘルスチェックプローブのポートを有効にします。

ステップ 1 **[デバイス（Devices）]** > **[プラットフォーム設定（Platform Settings）]** > **[HTTPアクセス（HTTP Access）]** の順に選択します。

ステップ 2 [HTTPサーバーの有効化（Enable HTTP Server）] チェックボックスをオンにします。

ステップ 3 [ポート（Port）] フィールドに、**443** と入力します。

ステップ 4 **[+ Add]** をクリックします。

ステップ 5 ドロップダウンリストから関連する [IPアドレス（IP Address）] を選択します。

ステップ 6 [使用可能なゾーン/インターフェイス（Available Zones/Interfaces）] ウィンドウで、GWLB または外部サブネットに接続されている外部インターフェイスを選択します。

ステップ 7 [追加（Add）]をクリックして、選択したインターフェイスを [選択したゾーン/インターフェイス（Selected Zones/Interfaces）] ウィンドウに追加します。

ステップ 8 [OK] をクリックします。

ステップ 9 [保存（Save）] をクリックします。

基本的なアクセス コントロール ポリシーの作成

新しいアクセス コントロール ポリシーを作成すると、そのポリシーにデフォルトのアクションと設定が含まれます。ポリシーを作成すると、要件に合わせてポリシーを調整できるよう、すぐに編集セッションに移行します。

ステップ 1 **[ポリシー（Policies）]** > **[アクセス制御（Access Control）]** を選択します。

ステップ 2 [新しいポリシー（New Policy）] をクリックします。

ステップ 3 一意の名前（aws-access-policy）と説明を入力します。

ステップ 4 最初の [デフォルトアクション（Default Action）] : [すべてのトラフィックをブロック（Block all traffic）] を指定します。

ステップ 5 **[保存（Save）]** をクリックします。

ステップ 6 作成した新しいポリシーの [編集（Edit）] アイコンをクリックします。

ステップ 7 [ルールの追加（Add Rule）] をクリックします。

ステップ 8 次のパラメータを設定します。

- 名前：inside-to-outside
- 挿入：into Mandatory
- アクション：Allow
- 送信元ゾーンと宛先ゾーンを追加します。

ステップ 9 [Apply] をクリックします。

ローカルホスト：設定 JSON ファイルの更新

configuration.json ファイルは、GitHub からダウンロードした **lambda_python_files** フォルダにあります。Management Center で設定したパラメータを使用して、**configuration.json** ファイルのパラメータを更新します。

configuration.json ファイル内のスクリプトは次のとおりです。

```
"licenseCaps": ["BASE", "MALWARE", "THREAT"],   // Management center virtual licenses
  "fmcIpforDeviceReg": "DONTRESOLVE",   // Management center virtual IP address
  "RegistrationId": "cisco",  // Registration ID used while configuring the manager in
the Threat defense virtual
  "NatId": "cisco",  // NAT ID used while configuring the manager in the Threat defense
 virtual
  "fmcAccessPolicyName": "aws-access-policy",  // Access policy name configured in the
Management center virtual
  "fmcInsideNicName": "inside", //Threat defense virtual inside interface name
  "fmcOutsideNicName": "outside", //Threat defense virtual outside interface name
  "fmcInsideNic": "GigabitEthernet0/0", // Threat defense virtual inside interface NIC
Name - GigabitEthernet for c4 instance types, and TenGigabitEthernet for c5 instance
types)
  "fmcOutsideNic": "GigabitEthernet0/1", // Threat defense virtual outside interface NIC
 Name - GigabitEthernet for c4 instance types, and TenGigabitEthernet for c5 instance
types
  "fmcOutsideZone": "Outside-sz", //Outside Interface security zone name that is set in
 the Management center virtual
  "fmcInsideZone": "Inside-sz", //Inside Interface security zone name that is set in the
 Management center virtual
  "interfaceConfig": [
    {
      "managementOnly": "false",
      "MTU": "1500",
      "securityZone": {
        "name": "Inside-sz"
      },
      "mode": "NONE",
      "ifname": "inside",
      "name": "GigabitEthernet0/0"
    },
    {
      "managementOnly": "false",
      "MTU": "1500",
      "securityZone": {
        "name": "Outside-sz"
      },
      "mode": "NONE",
      "ifname": "outside",
      "name": "GigabitEthernet0/1"
    }
  ],  // Interface-related configuration
  "trafficRoutes": [
    {
      "interface": "inside",
      "network": "any-ipv4",
      "gateway": "",
      "metric": "1"
    }
  ] // This traffic route is used for the Threat defense virtual instance's health check
}
```

ローカルホスト：ローカルホストでの AWS CLI を使用したインフラストラクチャ コンポーネントの設定

テンプレートでは、Threat Defense Virtual および Management Center の Lambda レイヤと暗号化されたパスワードは作成されません。次の手順を使用して、各コンポーネントを設定します。AWS CLI の詳細については、「AWS コマンドラインインターフェイス」を参照してください。

ステップ 1 Lambda レイヤ zip ファイルを作成します。

Linux ホストに Python フォルダを作成し、Lambda レイヤを作成します。

a) Linux ホストに Python フォルダ（Ubuntu 22.04 など）を作成します。

b) Linux ホストに Python 3.9 をインストールします。以下に、Python 3.9 をインストールするためのサンプルスクリプトを示します。

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt install python3.9
$ sudo apt install python3-virtualenv
$ sudo apt install zip
$ sudo apt-get install python3.9-distutils
$ sudo apt-get install python3.9-dev
$ sudo apt-get install libffi-dev
```

c) Linux 環境で Lambda レイヤ zip ファイル（autoscale_layer.zip）を作成します。このファイルは、Lambda 機能に不可欠な Python ライブラリを提供します。

次のスクリプトを実行して、autoscale_layer.zip ファイルを作成します。

```
#!/bin/bash
mkdir -p layer
mkdir -p python
virtualenv -p /usr/bin/python3.9 ./layer/
source ./layer/bin/activate
pip3 install attrs==23.1.0
pip3 install bcrypt==3.2.2
pip3 install certifi==2022.12.7
pip3 install cffi==1.15.1
pip3 install chardet==3.0.4
pip3 install cryptography==2.9.1
pip3 install idna==2.10
pip3 install jsonschema==3.2.0
pip3 install paramiko==2.7.1
pip3 install pycparser==2.21
pip3 install pycryptodome==3.15.0
pip3 install PyNaCl==1.5.0
pip3 install pyrsistent==0.19.3
pip3 install requests==2.23.0
pip3 install scp==0.13.2
pip3 install six==1.16.0
pip3 install urllib3==1.25.11
echo "Copy from ./layer directory to ./python\n"
cp -r ./layer/lib/python3.9/site-packages/* ./python/
zip -r autoscale_layer.zip ./python
```

d) **autoscale_layer.zip** ファイルを作成したら、GitHub からダウンロードした **lambda-python-files** フォルダに **autoscale_layer.zip** ファイルをコピーします。

ステップ 2 （任意）Threat Defense Virtual および Management Center の暗号化パスワードを作成します。

Infrastructure_gwlb.yaml テンプレートファイルに KMS ARN 値が入力されている場合は、Threat Defense Virtual および Management Center で設定するパスワードを暗号化する必要があります。AWS KMS コンソールを使用してキー ARN を特定するには、Finding the key ID and key ARN [英語] を参照してください。ローカルホストで、次の AWS CLI コマンドを実行してパスワードを暗号化します。

```
$ aws kms encrypt --key-id <KMS-ARN> --plaintext 'MyC0mplIc@tedProtect1oN'
{
    "KeyId": "KMS-ARN",
    "CiphertextBlob":
"AQICAHgcQFAGtz/hvaxMtJvY/x/rfHnKI3clFPpSXUU7HQRnCAFwfXhXHJAHL8tcVmDqurALAAAAajBoBgkqhki
G9w0BBwagWzBZAgEAMFQGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQM45AIkTqjSekX2mniAgEQgCcOav6Hhol
+wxpWKtXY4y1Z1d0z1P4fx0jTdosfCbPnUExmNJ4zdx8="
}
$
```

`CiphertextBlob` の値は暗号化されたパスワードです。このパスワードは、`infrastructure_gwlb.yaml` ファイルの **NGFWv パスワード**（Threat Defense Virtualパスワード）または Auto Scale 自動化の FMC パスワード（Management Centerパスワード）パラメータの値として使用します。このパスワードは、**CloudWatch にメトリックを公開するための FMC パスワード**の値としても使用できます。

ローカルホスト：target フォルダの作成

次のコマンドを使用して、Amazon S3 バケットにアップロードする必要があるファイルを含む target フォルダを作成します。

python3 make.py build

ローカルホストに「target」という名前のフォルダが作成されます。target フォルダには、Auto Scale ソリューションの展開に必要な zip ファイルと yaml ファイルが含まれています。

ローカルホスト：Amazon S3 バケットへの AWS GWLB Auto Scale ソリューション展開ファイルのアップロード

次のコマンドを使用して、target ディレクトリにあるすべてのファイルを Amazon S3 バケットにアップロードします。

$ cd ./target

$ aws s3 cp . s3://demo-us-bkt --recursive

Amazon CloudFormation コンソール：GWLB を使用して Threat Defense Virtual の Auto Scale ソリューションを展開する

ステップ 1 AWS 管理コンソールで、**[サービス（Services）]** > **[管理とガバナンス（Management and Governance）]** > **[CloudFormation]** > **[スタック（Stacks）]**の順に選択し、テンプレートによって作成されたスタックをクリックします。

ステップ 2 **[スタックの作成（Create stack）]** > **[新しいリソースを使用（標準）（With new resources (standard)）]**の順にクリックします。

ステップ 3 [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、target フォルダから ***deploy_ngfw_autoscale_with_gwlb.yaml*** を選択します。

ステップ 4 [次へ（Next）] をクリックします。

ステップ 5 [スタックの詳細の指定（Specify stack details）] ページで、スタックの名前を入力します。

ステップ 6 ***deploy_ngfw_autoscale_with_gwlb.yaml*** テンプレートの入力パラメータの値を指定します。

スタック名：Threat-Defense-Virtual

パラメータ	**値**
ポッドの設定	
Auto Scale グループ名プレフィックス	NGFWv-AutoScale
ポッド番号	*1*
Auto Scale 電子メール通知	username@cisco.com
インフラストラクチャの詳細	
VPC ID	vpc-05277f76370396df4
S3 バケット名	demo-us-bkt
Lambda 機能のサブネット	subnet-0f6bbd4de47d50c6b,subnet-0672f4c24156ac443
Lambda 機能のセキュリティグループ	sg-023dfadb1e7d4b87e
可用性ゾーンの数	*2*
可用性ゾーン	us-west-1a, us-west-1b
NGFWv 管理インターフェイスのサブネットリスト	subnet-0e0bc4961de87b170
NGFWv 内部インターフェイスのサブネットリスト	subnet-0f6acf3b548d9e95b
NGFWv 外部インターフェイスのサブネットリスト	subnet-0cc7ac70df7144b7e
GWLB の設定	

パラメータ	値
NGFWv インスタンスのヘルスチェック用のポートを入力	*22*
Cisco NGFWv インスタンスの設定	
NGFWv インスタンスタイプ	*C4.xlarge*
NGFWv インスタンス ライセンス タイプ	*BYOL*
AWS IP プールからの NGFWv のパブリック IP の割り当て	*true*
NGFWv インスタンスのセキュリティグループ	sg-088ae4bc1093f5833
内部の NGFWv インスタンスのセキュリティグループ	sg-0e0ce5dedcd9cd4f3
外部の NGFWv インスタンスのセキュリティグループ	sg-07dc50ff47d0c8126
NGFWv AMI-ID	ami-00faf58c7ee8d11e1
KMS マスターキー ARN（条件付き）	
NGFWv パスワード	W1nch3sterBr0s
FMC 自動化の設定	
FMC ホスト IP アドレス	3.38.137.49
Auto Scale 自動化の FMC ユーザー名	autoscaleuser
Auto Scale 自動化の FMC パスワード	W1nch3sterBr0s
FMC デバイスグループ名	aws-ngfw-autoscale-dg
FMCv ライセンスのパフォーマンス階レイヤの値	*FTDv20*
FMC デバイスグループメトリックの公開の設定	
FMC からのカスタムメトリックの公開	*TRUE*
CloudWatch にメトリックを公開するための FMC ユーザー名	metricuser
CloudWatch にメトリックを公開するための FMC パスワード	W1nch3sterBr0s
スケーリングの設定	
下限および上限 CPU しきい値	*10,70*

パラメータ	値
下限および上限メモリしきい値	40、70

ステップ 7 [スタックオプションの設定（Configure Stack Options）] ウィンドウで [次へ（Next）] をクリックします。

ステップ 8 [確認（Review）] ページで設定を確認して確定します。

ステップ 9 [スタックの作成（Create Stack）] をクリックして ***deploy_ngfw_autoscale_with_gwlb.yaml*** テンプレートを展開し、スタックを作成します。

これで、GWLB を使用して Threat Defense Virtual 用の Auto Scale ソリューションを設定するために必要な両方のテンプレートの展開が完了しました。

Amazon EC2 コンソール：Auto Scale グループのインスタンス数の編集

デフォルトでは、Auto Scale グループの Threat Defense Virtual インスタンスの最小数と最大数はそれぞれ 0 と 2 に設定されています。要件に応じて各値を変更します。

ステップ 1 AWS 管理コンソールで、**[サービス（Services）]** > **[コンピューティング（Compute）]** > **[EC2]**の順に選択し、[Auto Scalingグループ（Auto Scaling Groups）] をクリックします。

ステップ 2 作成したAuto Scalingグループを選択し、[編集（Edit）]をクリックして、要件に応じて[必要な容量（Desired capacity）]、[最小容量（Minimum capacity）]、[最大容量（Maximum capacity）] フィールドの値を変更します。各値は、Auto Scaling 機能のために起動する Threat Defense Virtual インスタンスの数に対応します。[必要な容量（Desired capacity）] を、最小容量値と最大容量値の範囲内の値に設定します。

ステップ 3 **[更新（Update）]**をクリックします。

（注） Threat Defense Virtual インスタンスを 1 つだけ起動し、そのインスタンスが想定どおりに動作しているか確認することを推奨します。その後、要件に応じて追加のインスタンスを起動できます。

Amazon VPC ダッシュボードコンソール：GWLB エンドポイントの作成およびカスタマー VPC のルーティングの設定

両方の CloudFormation テンプレートを展開後、GWLB エンドポイントを作成し、カスタマー VPC のルーティングを設定する必要があります。

GWLB エンドポイントの作成

ステップ 1 AWS 管理コンソールで、**[サービス（Services）]** > **[ネットワーキングおよびコンテンツ配信（Networking & Content Delivery）]** > **[VPC]** > **[エンドポイントサービス（Endpoint Services）]**の順に選択します。

ステップ 2 [エンドポイントサービスの作成（Create Endpoint Services）] をクリックします。

ステップ 3 [ロードバランサタイプ（Load balancer type）] で [ゲートウェイ（Gateway）] を選択します。

ステップ 4 [使用可能なロードバランサ（Available load Balancers）] で、Auto Scale の展開の一部として作成されたゲートウェイロードバランサを選択します。

ステップ 5 [作成（Create）] をクリックします。

ステップ 6 新たに作成したエンドポイントサービスのサービス名をコピーします。

ステップ 7 **[サービス（Services）]** > **[ネットワーキングおよびコンテンツ配信（Networking & Content Delivery）]** > **[VPC]** > **[エンドポイント（Endpoints）]**の順に選択します。

ステップ 8 [エンドポイントの作成（Create endpoint）] をクリックします。

ステップ 9 [サービスカテゴリ（Service category）] で [その他のエンドポイントサービス（Other endpoint services）] を選択します。

ステップ 10 [サービス名（Service name）] にサービスの名前を入力し、[サービスの確認（ Verify service）] を選択します。

ステップ 11 [VPC] フィールドで、エンドポイントを作成する VPC、[アプリケーションVPC（App VPC）] を選択します。

ステップ 12 [サブネット（Subnets）] で、エンドポイントを作成するサブネット、[出力サブネット（Egress subnet）] を選択します。

ステップ 13 [IPアドレスタイプ（IP address type）] で、[IPv4] オプションを選択して、エンドポイント ネットワークインターフェイスに IPv4 アドレスを割り当てます。

ステップ 14 [エンドポイントの作成（Create endpoint）] をクリックします。

ステップ 15 **[サービス（Services）]** > **[ネットワーキングおよびコンテンツ配信（Networking & Content Delivery）]** > **[VPC]** > **[エンドポイントサービス（Endpoint services）]**の順に選択し、[エンドポイント接続（Endpoint Connections）] タブをクリックし、事前に作成した [エンドポイントID（Endpoint ID）] を選択して、[アクション（Actions）] > [エンドポイント接続要求の受け入れ（Accept endpoint connection request）] の順にクリックします。

カスタマー VPC のルーティングの設定

ステップ 1 AWS 管理コンソールで、**[サービス（Services）]** > **[ネットワーキングおよびコンテンツ（Networking & Content）]** > **[仮想プライベートクラウド（Virtual Private Cloud）]** > **[ルートテーブル（Route tables）]**の順に選択します。

ステップ 2 入力ルートテーブルを作成し、次の手順を実行します。

1. **[アクション（Actions）]** > **[ルートの編集（Edit routes）]**の順にクリックします。

2. IPv4 の場合は、[ルートの追加（Add route）] をクリックします。[宛先（Destination）] に、アプリケーションサーバーのサブネットの IPv4 CIDR ブロック（10.0.1.0/24）を入力します。[ターゲット（Target）] で、VPC エンドポイントを選択します。
3. [変更の保存（Save Changes）] をクリックします。
4. [エッジの関連付け（Edge Associations）] タブで [エッジの関連付けの編集（Edit edge associations）] をクリックし、[インターネットゲートウェイ（Internet gateway）] を選択します。
5. [変更の保存（Save Changes）] をクリックします。

ステップ 3 アプリケーションサーバーがあるサブネットのルートテーブルを選択し、次の手順を実行します。

1. **[アクション（Actions）] > [ルートの編集（Edit routes）]**の順にクリックします。
2. IPv4 の場合は、[ルートの追加（Add route）] をクリックします。[宛先（Destination）] に、**0.0.0.0/0** と入力します。[ターゲット（Target）] で、VPC エンドポイントを選択します。
3. [変更の保存（Save Changes）] をクリックします。

ステップ 4 ゲートウェイロードバランサのエンドポイントがあるサブネットのルートテーブルを選択し、次の手順を実行します。

1. **[アクション（Actions）] > [ルートの編集（Edit routes）]**の順にクリックします。
2. IPv4 の場合は、[ルートの追加（Add route）] をクリックします。[宛先（Destination）] に、**0.0.0.0/0** と入力します。[ターゲット（Target）] で、インターネットゲートウェイを選択します。
3. [変更の保存（Save Changes）] をクリックします。

Amazon CloudWatch：展開の検証

テンプレートの展開が成功したら、Amazon CloudWatch コンソールに移動して、ログが収集され、必要なアラームが作成されていることを確認します。

ログ

ログファイルを確認して、Management Center の接続に関する問題をトラブルシューティングします。

ステップ 1 **AWS 管理**コンソールで、**[サービス（Services）] > [管理とガバナンス（Management and Governance）] > [CloudWatch]**の順に選択します。

ステップ 2 [ロググループ（Log groups）] をクリックし、表示されているいずれかのロググループをクリックしてログを表示します。

アラーム

必要なアラームが Amazon CloudWatch コンソールで作成されていることを確認します。

ステップ 1 **AWS 管理**コンソールで、**[サービス（Services）] > [管理とガバナンス（Management and Governance）] > [CloudWatch]**の順に選択します。

ステップ 2 **[アラーム（Alarms）] > [すべてのアラーム（All Alarms）]**の順にクリックして、スケールアウトおよびスケールイン機能をトリガーする条件とともにアラームのリストを表示します。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。