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 ソリューションの展開

初版：2023 年 6 月 1 日

はじめに

本書では、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 ソリューションの展開方法について説明します。

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。

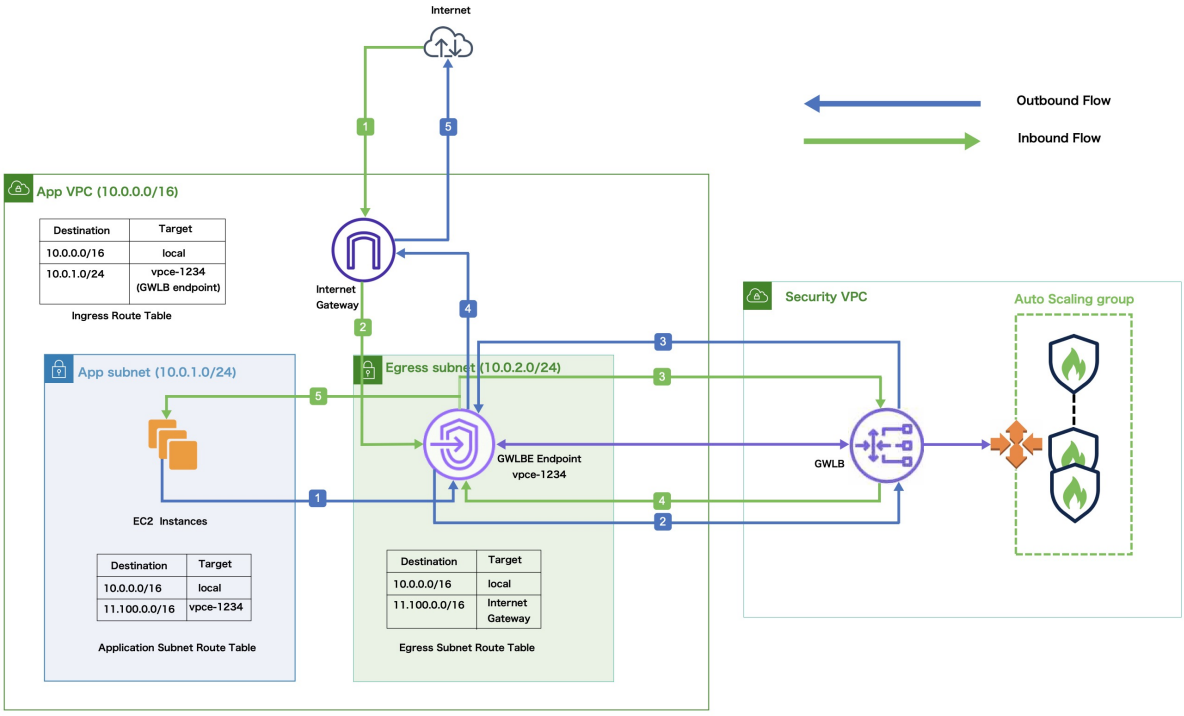
GWLB は、内部および外部で生成されたトラフィックを管理する単一のエントリおよびエグジットポイントとして機能し、トラフィック負荷に基づいて Threat Defense Virtual インスタンスの数をリアルタイムでスケールアップまたはスケールダウンします。

（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

このトポロジの例は、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 *1 : GWLB* を使用した *Threat Defense Virtual Auto Scale* ソリューション



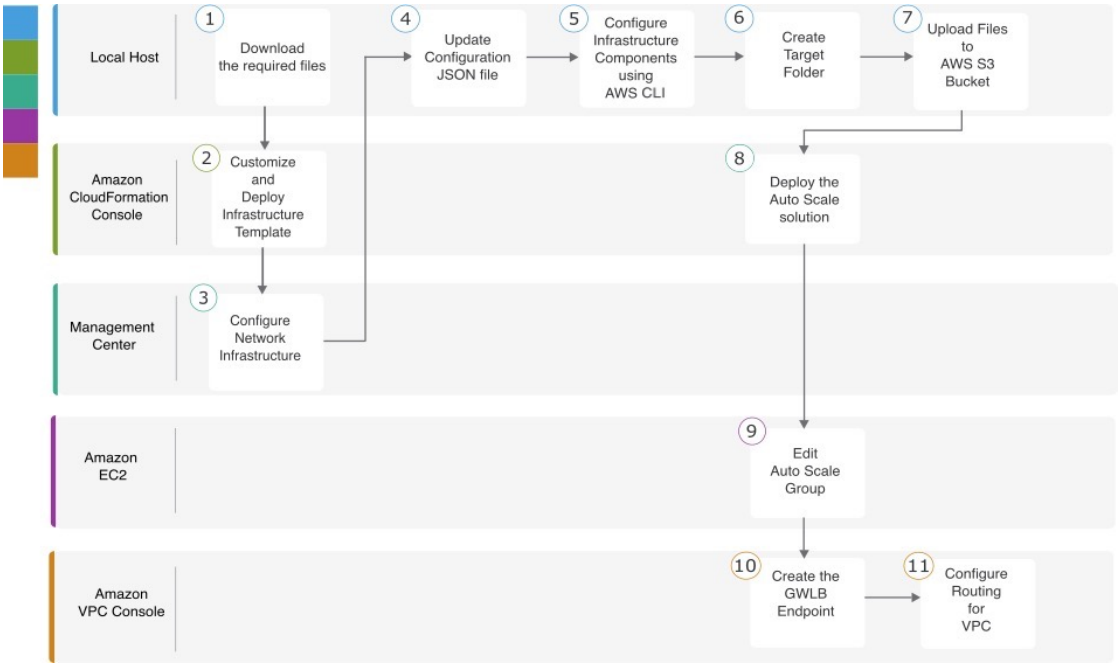
インバウンドトラフィック検査	
1	インターネットゲートウェイ（IGW）が、インターネットからトラフィックを受信します。
2	トラフィックが、入力ルートテーブルのルートに従ってゲートウェイロードバランサのエンドポイント（GWLBe）にルーティングされます。
3	GWLBeが、セキュリティ仮想プライベートクラウド（VPC）のエンドポイントサービスに接続されます。GWLBが受信したトラフィックをカプセル化し、検査のために Threat Defense Virtual Auto Scaling グループに転送します。
4	Auto Scaling グループによって検査されたトラフィックが GWLB 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 を使用した Threat Defense Virtual の Auto Scale ソリューションの展開
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。

手順

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

パラメータ	値
Lambda サブネット CIDR	20.1.50.0/24,20.1.51.0/24

ステップ 6 [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

手順

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

手順

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* と入力します。

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

手順

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`** と入力します。

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

ステップ 6 **[デバイス（Device）]** > **[インターフェイス（Interfaces）]** > ドロップダウンリストから、追加するインターフェイスを含むデバイスを選択します。

ステップ 7 セキュリティゾーンを作成または編集すると、**[デバイス（Device）]** > **[インターフェイス（Interfaces）]** > ドロップダウンリストに、高可用性デバイスのクラスタ名が表示されます。追加するインターフェイスを含むクラスタを選択します。

ステップ 8 1 つ以上のインターフェイスを選択します。

ステップ 9 [追加（Add）] をクリックして、選択したインターフェイス（デバイス別にグループ化済み）を追加します。

ステップ 10 [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更新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。

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）] : *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。

手順

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。

手順

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 [適用（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。

手順

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 関数に不可欠な 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 を使用した Threat Defense Virtual の Auto Scale ソリューションの展開

手順

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 [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 関数のサブネット	subnet-0f6bbd4de47d50c6b,subnet-0672f4c24156ac443
Lambda 関数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

パラメータ	値
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	
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層の値	*FTDv20*

パラメータ	値
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*
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。

手順

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

手順

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 Service）] 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

手順

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。

手順

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。

手順

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。