AWS への ASA 仮想 Auto Scale ソリューションの導入

AWS での Threat Defense Virtual ASA 仮想 の Auto Scale ソリューション

次のセクションでは、Auto Scale ソリューションのコンポーネントが AWS の ASA Virtual でどのように機能するかについて説明します。

Auto Scale ソリューションについて

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、ASA Virtual ファイアウォールの Auto Scaling グループを導入するための CloudFormation テンプレートとスクリプトを提供しています。

AWS の ASA Virtual Auto Scale は、AWS 環境の ASA Virtual インスタンスに水平 Auto Scaling 機能を追加する、完全なサーバーレス実装です（つまり、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、 Auto Scale ソリューションは、Management Center によって管理される でサポートされます。

ASA 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。

- スケールアウトされた ASA 仮想 インスタンスに完全に自動化された構成を自動適用。
- ロードバランサとマルチ可用性ゾーンのサポート。

- Auto Scale 機能の有効化と無効化をサポート。

サンドイッチトポロジを使用した Auto Scale の導入例

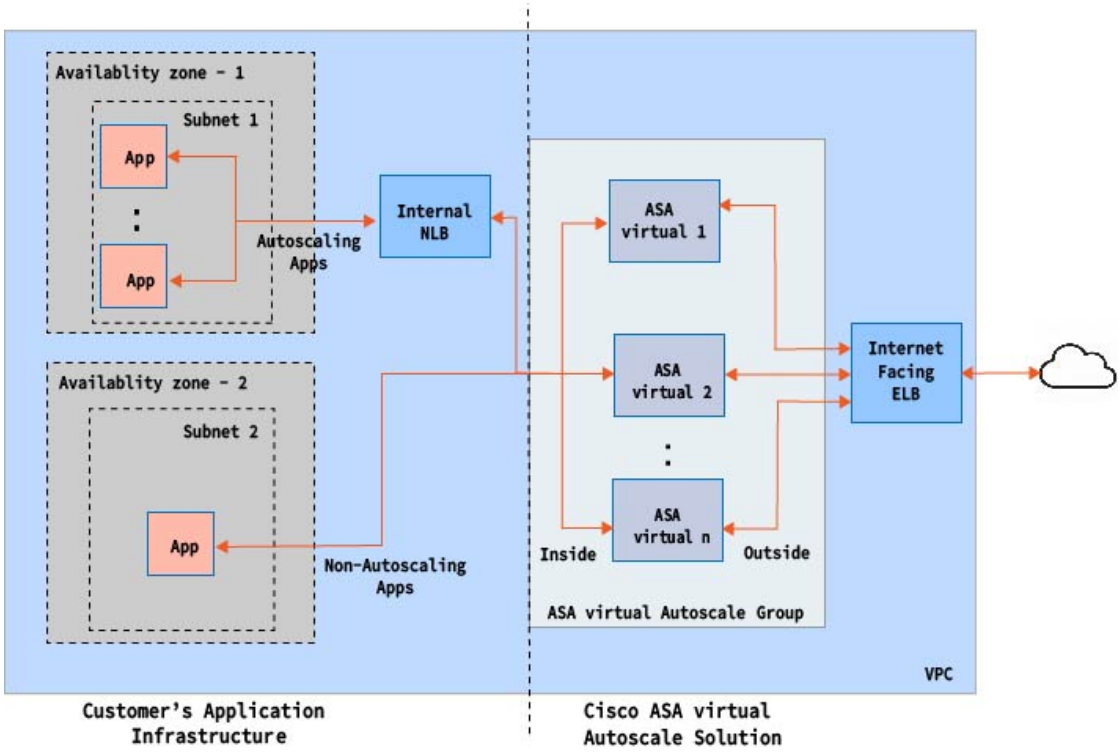
この ASA 仮想 AWS Auto Scale ソリューションの導入例は、導入例の図に示されています。AWS ロードバランサはインバウンドで開始された接続のみを許可するため、外部で生成されたトラフィックのみが ASA 仮想 ファイアウォール経由で内部を通過できます。

（注） 前提条件のSSL サーバー証明書 （7 ページ）で説明されているように、セキュアなポートには SSL/TLS 証明書が必要です。

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。導入例の図に示されているように、点線の右側部分は ASA 仮想 テンプレートを介して展開されます。左側は完全に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 ASA 仮想 を通過しません。

図 *1:* サンドイッチトポロジを使用した *ASA* 仮想 *Auto Scale* の導入例の図



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。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。

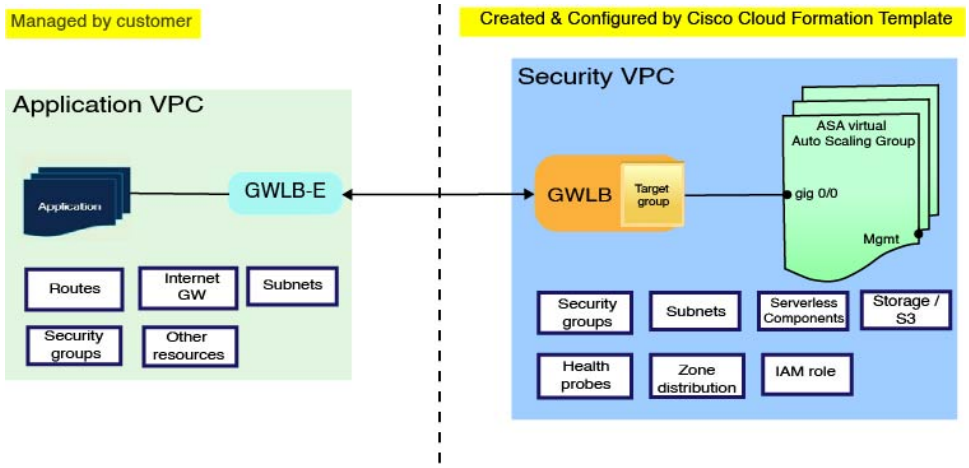
AWS ゲートウェイロードバランサの自動スケールの導入例

この ASA 仮想 AWS Gateway Load Balancer（GWLB）Auto Scale ソリューションの導入例は、導入例の図に示されています。AWS GWLB はインバウンド接続とアウトバウンド接続の両方を許可するため、内部と外部で生成されたトラフィックは Cisco ASA 仮想 ファイアウォール経由で内部を通過できます。

インターネットに接続するロードバランサは、AWS ゲートウェイロードバランサのエンドポイント（GWLBe）にすることができます。GWLBe はトラフィックを GWLB に送信し、検査のために ASA 仮想 に送信します。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 ASA 仮想 テンプレートを介して展開された ASA 仮想 GWLB 自動スケール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。

（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 ASA 仮想 を通過しません。

図 *2 : ASA* 仮想 *AWS GWLB Auto Scale* の導入例の図



Auto Scale ソリューションの仕組み

ASA Virtual インスタンスをスケールインおよびスケールアウトするには、Auto Scale Manager と呼ばれる外部エンティティがメトリックをモニターし、Auto Scale グループに ASA Virtual インスタンスの追加または削除を指示し、 ASA Virtualインスタンスを設定します。

Auto Scale Manager は、AWS サーバーレスアーキテクチャを使用して実装され、AWS リソース および ASA 仮想 と通信します。シスコでは、Auto Scale Manager コンポーネントの導入を

自動化する CloudFormation テンプレートを提供しています。このテンプレートにより、包括的なソリューションが機能するために必要なその他のリソースも展開されます。

（注） サーバーレス Auto Scale スクリプトは CloudWatch イベントによってのみ呼び出されるため、インスタンスの起動時にのみ実行されます。

Auto Scale ソリューションのコンポーネント

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 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。
- ASA 構成ファイルを使用して新しい ASA 仮想 を設定し展開します。

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、ASA 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 関数をトリガーします。

- 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 関数をトリガーし、ターゲットグループから ASA 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。

Auto Scale ソリューションの前提条件

展開ファイルのダウンロード

ASA Virtual Auto Scale for AWS ソリューションの起動に必要なファイルをダウンロードします。該当する ASA バージョン用の展開スクリプトとテンプレートは、GitHub リポジトリから入手できます。

注目 Auto Scale 用のシスコ提供の導入スクリプトおよびテンプレートは、オープンソースの例として提供されており、通常の Cisco TAC サポートの範囲内ではカバーされないことに注意してください。更新と ReadMe の手順については、GitHub を定期的に確認してください。

インフラストラクチャ設定

複製/ダウンロードされた GitHub リポジトリでは、**infrastructure.yaml** ファイルはテンプレートフォルダ内にあります。この CFT は、バケットポリシーを使用して VPC、サブネット、ルート、ACL、セキュリティグループ、VPC エンドポイント、および S3 バケットを展開するために使用できます。この CFT は、要件に合わせて変更できます。

次の項では、これらのリソースと Auto Scale での使用について詳しく説明します。これらのリソースを手動で展開し、Auto Scale で使用することもできます。

（注） **infrastructure.yaml** テンプレートは、VPC、サブネット、ACL、セキュリティグループ、S3 バケット、および VPC エンドポイントのみを展開します。SSL 証明書、Lambda レイヤ、または KMS キーリソースは作成されません。

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。

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。導入例に示されているように、ASA Virtual マシン の動作には 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、ASA 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。ASA 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 AutoScale 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。ただし、ロードバランサから ASA 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これを変更できます。この場合、AWS メタデータ サーバー オブジェクトをそれぞれの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、ASA Virtual 管理インターフェイスが含まれます。 デフォルトルートを設定することは任意です。

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。Lambda サブネットのベストプラクティスについては、AWS のドキュメントを参照してください。

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。AWS Elastic Load Balancing ユーザーガイド [英語] 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。

表 *1* : 必須のポート

ポート	使用方法	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

Amazon 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。ファイアウォール テンプレートとアプリケーション テンプレートの両方に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 関数が作成されます。したがって、S3 バケットはユーザーアカウントにアクセス可能である必要があります。

SSL サーバー証明書

インターネットに面したロードバランサが TLS/SSL をサポートしている必要がある場合、証明書 ARN が必要です。詳細については、次のリンクを参照してください。

- サーバー証明書の使用
- テスト用の秘密キーと自己署名証明書の作成

- 自己署名 SSL 証明書を使用した AWS ELB の作成（サードパーティリンク）

ARN の例：arn:aws:iam::[AWS Account]:server-certificate/[Certificate Name]

Lambda レイヤ

autoscale_layer.zip は、Python 3.9 がインストールされた Ubuntu 18.04 などの Linux 環境で作成できます。

```
#!/bin/bash
mkdir -p layer
virtualenv -p /usr/bin/python3.9 ./layer/
source ./layer/bin/activate
pip3 install cffi==1.15.1
pip3 install cryptography==2.9.1
pip3 install paramiko==2.7.1
pip3 install requests==2.23.0
pip3 install scp==0.13.2
pip3 install jsonschema==3.2.0
pip3 install pycryptodome==3.15.0
echo "Copy from ./layer directory to ./python\n"
cp -r ./layer/lib/python3.9/site-packages/* ./python/
zip -r autoscale_layer.zip ./python
```

作成された ***autoscale_layer.zip*** ファイルは、*lambda-python-files* フォルダにコピーする必要があります。

KMS マスターキー

これは、ASA 仮想パスワードが暗号化形式の場合に必要です。それ以外の場合、このコンポーネントは必要ありません。パスワードは、ここで提供される KMS のみを使用して暗号化する必要があります。KMS ARN が CFT で入力される場合、パスワードを暗号化する必要があります。それ以外の場合、パスワードはプレーンテキストである必要があります。

マスターキーと暗号化の詳細については、パスワードの暗号化と KMS に関する AWS のドキュメントのキーの作成 [英語] と AWS CLI コマンドリファレンス [英語] を参照してください。

例：

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

CiphertextBlob キーの値をパスワードとして使用する必要があります。

Python 3 環境

make.py ファイルは、複製されたリポジトリの最上位ディレクトリにあります。これにより、python ファイルが Zip ファイルに圧縮され、ターゲットフォルダにコピーされます。これらのタスクを実行するには、Python 3 環境が使用可能である必要があります。

Auto Scale の展開

準備

アプリケーションが展開されているか、アプリケーションの展開プランが利用可能である必要があります。

入力パラメータ

導入前に、次の入力パラメータを収集する必要があります。

（注） AWS Gateway Load Balancer（GWLB）の場合**LoadBalancerType**、**LoadBalancerSG**、**LoadBalancerPort**、および**SSLcertificate** パラメータは対象外です。

表 *2 : Auto Scale* 入力パラメータ

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（ASA 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 *ASA 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-ASA 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

パラメータ	使用できる値/タイプ	説明
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。ASA 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 ASA 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、ASA 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

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 ASA 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
ASAvInstanceType	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。 ASA 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
ASAvLicenseType	文字列	ASA 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
ASAvAmiId	文字列	ASA Virtual AMI ID（有効な Cisco ASA 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。

パラメータ	使用できる値/タイプ	説明
ConfigFileURL	文字列	ASA 仮想 構成ファイルの HTTP URL。各 AZ の構成ファイルはURL で使用できる必要があります。Lambda 関数が正しいファイルの選択を処理します。 HTTP サーバーをホスト構成ファイルに展開することも、AWS S3 の静的な Web ホスティング機能を使用することもできます。 （注） インポート時に構成ファイル名がURL に付加されるため、末尾の「/」も必要です。 「*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：https://myserver/asavconfig/asaconfig .txt/
NoOfAZs	整数	ASA 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

パラメータ	使用できる値/タイプ	説明
ASAvMgmtSubnetId	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。
ASAvInsideSubnetId	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。
ASAvOutsideSubnetId	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、ASA 仮想 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
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

ASA 構成ファイルの更新

ASA 構成ファイルを準備し、ASA 仮想 インスタンスからアクセス可能な HTTP/HTTPS サーバーに保存します。これは標準の ASA 構成ファイル形式です。スケールアウトされた ASA 仮想 により、構成ファイルがダウンロードされて構成が更新されます。

以下のセクションでは、Auto Scale ソリューション用に ASA 構成ファイルを変更する方法の例を示します。

オブジェクト、デバイスグループ、NAT ルール、アクセスポリシー

ASA 仮想 構成のロードバランサの正常性プローブのオブジェクト、ルート、および NAT ルールの例については、次を参照してください。

```
! Load Balancer Health probe Configuration
object network aws-metadata-server
host 169.254.169.254
object service aws-health-port
service tcp destination eq 7777
object service aws-metadata-http-port
service tcp destination eq 80
route inside 169.254.169.254 255.255.255.255 10.0.100.1 1
nat (outside,inside) source static any interface destination static interface
aws-metadata-server service aws-health-port aws-metadata-http-port
!
```

（注） 上記の正常性プローブ接続がアクセスポリシーで許可されている必要があります。

ASA 仮想 構成のデータプレーンの構成例については、次を参照してください。

```
! Data Plane Configuration
route inside 10.0.0.0 255.255.0.0 10.0.100.1 1
object network http-server-80
host 10.0.50.40
object network file-server-8000
host 10.0.51.27
object service http-server-80-port
service tcp destination eq 80
nat (outside,inside) source static any interface destination static interface
```

```
http-server-80 service http-server-80-port http-server-80-port
object service file-server-8000-port
service tcp destination eq 8000
nat (outside,inside) source static any interface destination static interface
file-server-8000 service file-server-8000-port file-server-8000-port
object service https-server-443-port
service tcp destination eq 443
nat (outside,inside) source static any interface destination static interface
http-server-80 service https-server-443-port http-server-80-port
!
```

構成ファイルの更新

ASA 仮想 構成は、*az1-connfiguration.txt*、*az2-configuration.txt*、および *az3-configuration.txt* ファイルで更新する必要があります。

（注） 3 つの構成ファイルがあると、可用性ゾーン（AZ）に基づいて構成を変更できます。たとえば、aws-metadata-server へのスタティックルートには、各 AZ に異なるゲートウェイがあります。

テンプレートの更新

deploy_autoscale.yaml テンプレートは慎重に変更する必要があります。***LaunchTemplate*** の [ユーザーデータ（UserData）] フィールドを変更する必要があります。[ユーザーデータ（UserData）] は必要に応じて更新できます。*name-server* を適宜更新する必要があります。たとえば、VPC DNS IP にすることができます。利用するライセンスが BYOL の場合、ライセンスの *idtoken* をここで共有する必要があります。

```
!
dns domain-lookup management
DNS server-group DefaultDNS
name-server <VPC DNS IP>
!
! License configuration
        call-home
        profile License
        destination transport-method http
        destination address http <url>
        license smart
        feature tier standard
        throughput level <entitlement>
        license smart register idtoken <token>
```

Amazon Simple Storage Service（S3）へのファイルのアップロード

target ディレクトリ内のすべてのファイルを Amazon S3 バケットにアップロードする必要があります。必要に応じて、CLI を使用して、*target* ディレクトリ内のすべてのファイルを Amazon S3 バケットにアップロードできます。

```
$ cd ./target
$ aws s3 cp . s3://<bucket-name> --recursive
```

スタックの展開

展開のすべての前提条件が完了すると、AWS CloudFormation スタックを作成できます。

target ディレクトリ内の *deploy_autoscale.yaml* ファイルを使用します。

target ディレクトリ内の *deploy_ngfw_autoscale_with_gwlb.yaml* ファイルを使用します。

（注） *deploy_ngfw_autoscale_with_gwlb.yaml* ファイルを展開する前に、AWS GWLB 自動スケール ソリューション用に **infrastructure_gwlb.yaml** ファイルを展開する必要があります。

deploy_autoscale_with_gwlb.yaml テンプレートの展開時に作成される GWLB を選択して、ゲートウェイ ロードバランサー エンドポイント（GWLB-E）を作成する必要があります。GWLBe を作成したら、アプリケーションサブネットとデフォルトルートテーブルで GWLBe を使用するようにデフォルトルートを更新する必要があります。

詳細については、「https://docs.amazonaws.cn/en_us/vpc/latest/privatelink/create-endpoint-service-gwlbe.html」を参照してください。

入力パラメータ （9 ページ）で収集されたパラメータを入力します。

展開の検証

テンプレートの展開が成功したら、Lambda 関数と CloudWatch イベントが作成されていることを検証する必要があります。デフォルトでは、Auto Scale グループのインスタンスの最小数と最大数はゼロです。AWS EC2 コンソールで必要な数のインスタンスを使用して、Auto Scale グループを編集する必要があります。これにより、新しい ASA Virtual インスタンスがトリガーされます。

1 つのインスタンスのみを起動してワークフローを確認し、そのインスタンスが期待どおりに動作しているかどうかを検証することを推奨します。その後に ASA Virtual の実際の要件を展開でき、動作を確認することもできます。AWS スケーリングポリシーによる削除を回避するために、最小数の ASA Virtual インスタンスをスケールイン保護としてマークできます。

Auto Scale 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 ASA Virtual VM に属する異常な IP がある場合、ASA 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 ASA 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、ASA 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

ASA 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 ASA 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、ASA 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、ターゲットグループからのターゲットの登録解除 （19 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。「ターゲットグループへのターゲットの登録 （19 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。

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 （19 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。

ASA Virtual の管理者パスワードを変更します。

ASA 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ASA Virtualデバイスをオンボードする場合、ASA 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。

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。

Auto Scale のトラブルシューティングとデバッグ

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。

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。

ASA 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

ASA 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 ASA 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。

ASA 仮想 が設定に失敗

ASA 仮想 の設定に失敗した場合は、Amazon S3 の静的な HTTP Web サーバーのホスティング構成への接続を確認してください。詳細については、「Amazon S3 での静的な Web サイトのホスティング（Hosting a static website on Amazon S3）」https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.htmlを参照してください。

ASA 仮想 でライセンス交付に失敗

ASA 仮想 がライセンスに失敗した場合は、CSSM サーバーへの接続、ASA 仮想 セキュリティグループの構成、アクセス制御リストを確認します。

ASA Virtual に SSH 接続できない

ASA 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 ASA Virtual に渡されたかどうかを確認します。

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。