AWS クラウドへの ASA 仮想 の導入

Amazon Web Services（AWS）クラウドに ASA 仮想 を導入できます。

重要 9.13(1) 以降では、サポートされているすべての ASA 仮想 vCPU/メモリ構成ですべての ASA 仮想 ライセンスを使用できるようになりました。これにより、ASA 仮想 を使用しているお客様は、さまざまなVMリソースフットプリントで実行できるようになります。また、サポート対象の AWS インスタンスタイプの数も増えます。

- 概要 （1 ページ）
- 前提条件 （4 ページ）
- 注意事項と制約事項 （5 ページ）
- 設定の移行と SSH 認証 （7 ページ）
- ネットワークトポロジの例 （7 ページ）
- AWS での ASA Virtual に対するインスタンス メタデータ データ サービス （9 ページ）
- ASA 仮想 の導入 （10 ページ）
- Amazon GuardDuty サービスと Threat Defense Virtual の統合 （15 ページ）
- Secure Firewall ASA Virtual と GuardDuty の統合について （15 ページ）
- サポートされるソフトウェア プラットフォーム （18 ページ）
- Amazon GuardDuty と Secure Firewall ASA Virtual の統合のガイドラインと制限事項 （18 ページ）
- Amazon GuardDuty との統合 ASA 仮想 （19 ページ）
- 既存のソリューション展開構成の更新 （32 ページ）
- パフォーマンスの調整 （34 ページ）

概要

ASA 仮想 は、物理 ASA と同じソフトウェアを実行して、仮想フォームファクタにおいて実証済みのセキュリティ機能を提供します。ASA 仮想 は、パブリック AWS クラウドに導入できます。その後設定を行うことで、時間の経過とともにロケーションを展開、契約、またはシフトする仮想および物理データセンターのワークロードを保護できます。

ASA 仮想 は、次の AWS インスタンスタイプをサポートしています。

表 *1: AWS* でサポートされているインスタンス タイプ

インスタンス	属性		インターフェイスの最大数
	vCPU	メモリ（**GB**）	
c3.large	2	3.75	3
c3.xlarge	4	7.5	4
c3.2xlarge	8	15	4
c4.large	2	3.75	3
c4.xlarge	4	7.5	4
c4.2xlarge	8	15	4
c5.large	2	4	3
c5.xlarge	4	8	4
c5.2xlarge	8	16	4
c5.4xlarge	16	32	8
c5a.large	2	4	3
c5a.xlarge	4	8	4
c5a.2xlarge	8	16	4
c5a.4xlarge	16	32	8
c5ad.large	2	4	3
c5ad.xlarge	4	8	4
c5ad.2xlarge	8	16	4
c5ad.4xlarge	16	32	8
c5d.large	2	4	3
c5d.xlarge	4	8	4
c5d.2xlarge	8	16	4
c5d.4xlarge	16	32	8
c5n.large	2	5.3	3
c5n.xlarge	4	10.5	4

インスタンス	属性		インターフェイスの最大数
	vCPU	メモリ（GB）	
c5n.2xlarge	8	21	4
c5n.4xlarge	16	54	8
m4.large	2	8	2
m4.xlarge	4	16	4
m4.2xlarge	8	32	4
m5n.large	2	8	3
m5n.xlarge	4	16	4
m5n.2xlarge	8	32	4
m5n.4xlarge	16	64	8
m5zn.large	2	8	3
m5zn.xlarge	4	16	4
m5zn.2xlarge	8	32	4
c6i.large	2	4	3
c6i.xlarge	4	8	4
c6i.2xlarge	8	16	4
c6i.4xlarge	16	32	8
C6a.large	2	4	3
C6a.xlarge	4	8	4
C6a.2xlarge	8	16	4
C6a.4xlarge	16	32	8
c6in.large	2	4	3
c6in.xlarge	4	8	4
c6in.2xlarge	8	16	4
c6in.4xlarge	16	32	8

ヒント　M4 または C4 インスタンスタイプを使用している場合は、パフォーマンスを向上させるために、Nitro ハイパーバイザと Elastic Network Adapter（ENA）インターフェイスドライバを使用する M5 または C5 インスタンスタイプに移行することを推奨します。

表 *2: ASA* 仮想 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	インスタンスタイプ（コア/RAM）	レート制限（Rate Limit）	RA VPN セッション制限
ASAv5	c5.large 2 コア/4 GB	100 Mbps	50
ASAv10	c5.large 2 コア/4 GB	1 Gbps	250
ASAv30	c5.xlarge 4 コア/8 GB	[2 Gbps]	750
ASAv50	c5.2xlarge 8 コア/16 GB	10 Gbps	10,000
ASAv100	c5n.4xlarge 16 コア/42 GB	16 Gbps	20,000

AWS にアカウントを作成し、AWS ウィザードを使用して ASA 仮想 をセットアップして、Amazon Machine Image（AMI）を選択します。AMI はインスタンスを起動するために必要なソフトウェア構成を含むテンプレートです。

重要　AMI イメージは AWS 環境の外部ではダウンロードできません。

前提条件

- aws.amazon.com でアカウントを作成します。
- ASA 仮想 へのライセンス付与。ASA 仮想 にライセンスを付与するまでは、100 回の接続と 100 Kbps のスループットのみが許可される縮退モードで実行されます。「ASA 仮想 のライセンス」を参照してください。

（注） これまでASA Virtual向けにシスコが提供していたすべてのデフォルトのソフトウェア利用資格でIPv6の設定がサポートされます。

- インターフェイスの要件：
  - 管理インターフェイス
  - 内部および外部インターフェイス
  - （任意）追加のサブネット（DMZ）
- 通信パス：
  - 管理インターフェイス：ASDM に ASA 仮想 を接続するために使用され、トラフィックの通過には使用できません。
  - 内部インターフェイス（必須）：内部ホストに ASA 仮想 を接続するために使用されます。
  - 外部インターフェイス（必須）：ASA 仮想 をパブリック ネットワークに接続するために使用されます。
  - DMZ インターフェイス（任意）：c3.xlarge インターフェイスを使用する場合、DMZ ネットワークに ASA 仮想 を接続するために使用されます。
- ASA 仮想 システム要件については、Cisco Secure Firewall ASA の互換性 [英語] を参照してください。

注意事項と制約事項

サポートされる機能

AWS 上の ASA 仮想 は、次の機能をサポートしています。

- 次世代の Amazon EC2 Compute Optimized インスタンスファミリである Amazon EC2 C5 インスタンスのサポート
- 仮想プライベート クラウド（VPC）への導入
- 拡張ネットワーク（SR-IOV）（使用可能な場合）
- Amazon マーケットプレイスからの導入
- L3 ネットワークのユーザー導入
- ルーテッド モード（デフォルト）
- IPv6

- Amazon CloudWatch
- クラスタリング

サポートされない機能

AWS 上の ASA 仮想 は、以下の機能をサポートしていません。

- コンソール アクセス（管理は、ネットワーク インターフェイスを介して SSH または ASDM を使用して実行される）
- VLAN
- 無差別モード（スニファなし、またはトランスペアレント モードのファイアウォールのサポート）
- マルチ コンテキスト モード
- ASA 仮想 ネイティブ HA
- EtherChannel は、ダイレクト物理インターフェイスのみでサポートされる
- VM のインポート/エクスポート
- ハイパーバイザに非依存のパッケージ
- VMware ESXi
- ブロードキャスト/マルチキャスト メッセージ

  これらのメッセージは AWS 内で伝播されないため、ブロードキャスト/マルチキャストを必要とするルーティング プロトコルは AWS で予期どおりに機能しません。VXLAN はスタティック ピアでのみ動作できます。

- Gratuitous/非要請 ARP

  これらの ARPS は AWS 内では受け入れられないため、Gratuitous ARP または非要請 ARP を必要とする NAT 設定は期待どおりに機能しません。

インスタンス メタデータ データ サービス（IMDS）サービスに関する ASA Virtual の制限事項

- たとえば、IMDS モードはいつでも変更できます。
- IMDSv2 必須モードに切り替える前に、製品バージョンでこのモードがサポートされていることを確認してください。サポートされていない場合、IMDS に依存する一部のサービスが失敗する可能性があります。
- 古いバージョン（IMDSv2 サポート対象外）の場合、IMDSv2 オプションモードでのみ展開可能です。
- 新しいバージョン（IMDSv2 サポート対象）では、IMDSv2 オプションモードと IMDSv2 必須モードの両方で展開できます。ただし、IMDSv2 必須モードが推奨されます。

設定の移行と SSH 認証

SSH 公開キー認証使用時のアップグレードの影響：SSH 認証が更新されることにより、SSH 公開キー認証を有効にするための新たな設定が必要となります。そのため、アップグレード後は、公開キー認証を使用した既存の SSH 設定は機能しません。公開キー認証は、Amazon Web Services（AWS）の ASA 仮想 のデフォルトであるため、AWS ユーザーにはこの問題が表示されます。SSH 接続を失なう問題を避けるには、アップグレードの前に設定を更新します。または（ASDM アクセスが有効になっている場合）アップグレード後に ASDM を使用して設定を修正できます。

次は、ユーザー名「admin」の元の設定例です。

```
username admin nopassword privilege 15
username admin attributes
  ssh authentication publickey 55:06:47:eb:13:75:fc:5c:a8:c1:2c:bb:
  07:80:3a:fc:d9:08:a9:1f:34:76:31:ed:ab:bd:3a:9e:03:14:1e:1b hashed
```

ssh authentication コマンドを使用するには、アップグレードの前に次のコマンドを入力します。

```
aaa authentication ssh console LOCAL
username admin password <password> privilege 15
```

nopassword キーワードが存在している場合、これを維持するのではなく、代わりにユーザー名に対応したパスワードを設定することを推奨します。**nopassword** キーワードは、パスワードは入力不可を意味するのではなく、任意のパスワードを入力できます。9.6(2) より前のバージョンでは、**aaa** コマンドは SSH 公開キー認証に必須ではありませんでした。このため、**nopassword** キーワードはトリガーされませんでした。9.6(2) では **aaa** コマンドが必須となり、**password**（または **nopassword**）キーワードが存在する場合、自動的に **username** の通常のパスワード認証を許可するようになりました。

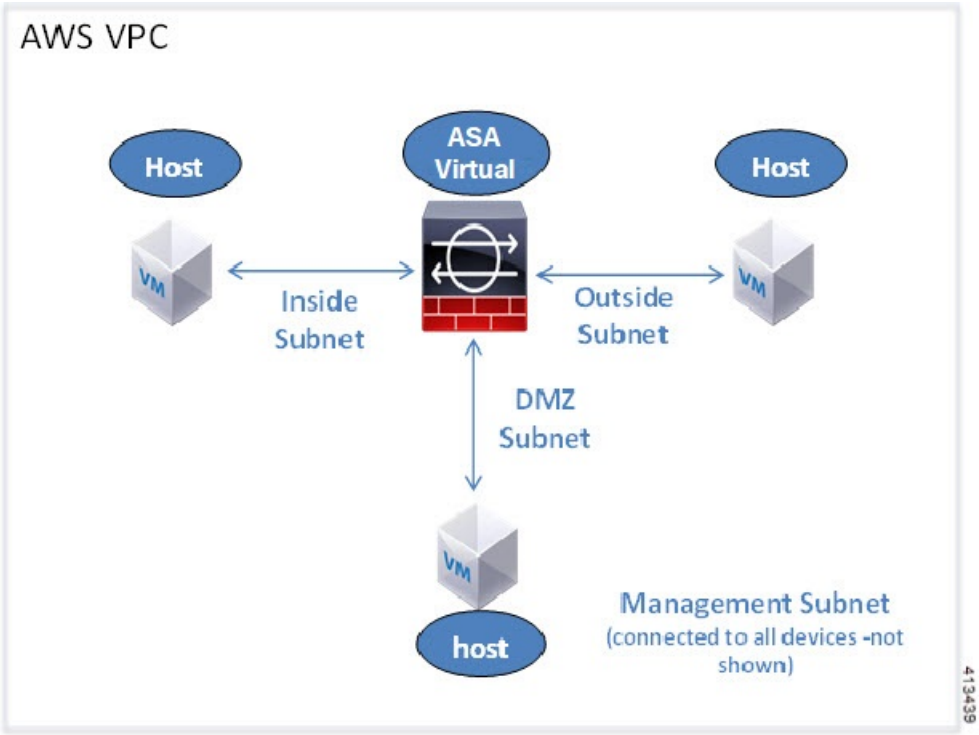
アップグレード後は、 **username** コマンドに対する **password** または **nopassword** キーワードの指定は任意となり、ユーザーがパスワードを入力できなくするよう指定できるようになります。よって、公開キー認証のみを強制的に使用する場合は、**username** コマンドを入力しなおします。

```
username admin privilege 15
```

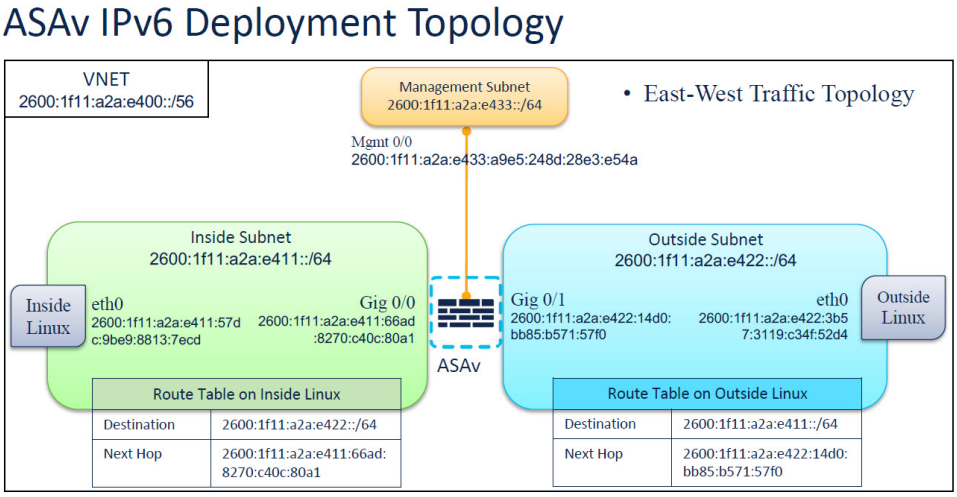
ネットワークトポロジの例

次の図は、ASA 仮想 用に AWS 内で設定された 4 つのサブネット（管理、内部、外部、および DMZ）を備えたルーテッド ファイアウォール モードの ASA 仮想 の推奨トポロジを示しています。

図 *1 : AWS* への *ASA* 仮想 の導入例



IPv6 トポロジ



AWS での ASA Virtual に対するインスタンス メタデータ データ サービス

インスタンス メタデータ データ サービス（IMDS）では、AWS に展開されている ASA Virtual インスタンスデータに関する情報が提供されます。これには、仮想インスタンスのネットワーク、ストレージ、およびその他のデータに関する詳細情報が含まれます。このメタデータを使用して、設定の決定（Day 0 構成）を自動化したり、インスタンスタイプやリージョンなどのインスタンス情報を表示したりできます。

IMDS API は、デバイスの起動時に AWS から ASA Virtual インスタンスのメタデータを収集し、後でインスタンスを設定します。現在、ASA Virtual インスタンスは、IMDSv1 API を使用してインスタンスのメタデータを取得し、検証します。IMDSv2 API は、ASA Virtualバージョン 9.20.3 以降でサポートされています。

AWS での ASA Virtual インスタンス用 IMDS の設定

AWS は、ASA Virtual に対して次の IMDSv2 モードをサポートしています。

- **V1 と V2（トークンはオプション）**：IMDSv1 API、IMDSv2 API、または IMDSv1 API と IMDSv2 API の両方の組み合わせを有効にする ASA Virtual インスタンスを展開します。
- **V2 のみ（トークンが必要）**：（推奨）IMDSv2 API のみを有効にする ASA Virtual インスタンスを展開します。

次の導入シナリオのインスタンスに対して、AWS で IMDS を設定できます。

新規導入：ASA Virtualインスタンスを新たに導入する場合は、**IMDSv2 必須**モードを設定します。新規導入の場合は、次のいずれかの方法を使用して IMDSv2 を有効にします。

- AWS EC2 コンソール：スタンドアロンインスタンスの導入の場合は、AWS EC2 コンソールの [高度な詳細（Advance Details）] セクションで [V2のみ（トークンが必要）（V2 only (token required)）] を有効にします。
- CloudFormation テンプレート：テンプレートの [MetadataOptions] で HttpEndpoint: enabled および HttpTokens: required プロパティを使用して、[V2のみ（トークンが必要）（V2 only (token required)）] - IMDSv2 必須モードを有効にします。これは、Auto Scale およびクラスタリングの展開に適用されます。

既存の展開：ASA Virtual を IMDSv2 API 対応バージョンにアップグレードした後、IMDSv2 オプションモードを IMDSv2 必須モードに設定します。

ASA 仮想 の導入

次の手順は、ASA 仮想 で AWS をセットアップする手順の概略を示しています。詳細な手順については、『Getting Started with AWS』を参照してください。

手順

ステップ 1 aws.amazon.com にログインし、地域を選択します。

（注）
AWS は互いに分かれた複数の地域に分割されています。地域は、ページの右上隅に表示されます。ある地域で利用可能なリソースは、別の地域には表示されません。目的の地域内に存在していることを定期的に確認してください。

ステップ 2 **[マイアカウント（My Account）]** > **[AWS管理コンソール（AWS Management Console）]** をクリックして [ネットワーキング（Networking）] で **[VPC]** > **[VPCウィザードの起動（Start VPC Wizard）]** をクリックし、単一のパブリックサブネットを選択して VPC を作成し、次を設定します（特記のないかぎり、デフォルト設定を使用します）。

- 内部および外部のサブネット：VPC およびサブネットの名前を入力します。
- インターネットゲートウェイ：インターネットゲートウェイの名前を入力します。これにより、インターネットを介した直接接続が可能になります。
- 外部テーブル：インターネットへの発信トラフィックを有効にするためのエントリを追加します（インターネットゲートウェイに 0.0.0.0/0 を追加します）。

（注）
IPv6 だけを使用して、仮想ネットワーク、サブネット、インターフェースなどを構築することはできません。デフォルトでは IPv4 が使用され、IPv6 も一緒に有効にできます。IPv6 の詳細については、「AWS IPv6 Overview」と「AWS VPC Migration」を参照してください。

ステップ 3 **[My Account]** > **[AWS Management Console]** > **[EC2]** をクリックし、さらに、[Create an Instance] をクリックします。

- AMI（Ubuntu Server 14.04 LTS など）を選択します。

  イメージ配信通知で識別された AMI を使用します。

- ASA 仮想 でサポートされるインスタンスタイプ（c3.large など）を選択します。
- インスタンスを設定します（CPU とメモリは固定です）。
- [高度な詳細（Advanced Details）] セクションを展開し、オプションの [ユーザーデータ（User data）] フィールドに第 0 日用構成を入力できます。これは、ASA 仮想 の起動時に適用される ASA 仮想 構成を含むテキスト入力です。スマートライセンスなどの詳細情報を持つ第 0 日用構成の詳細については、「第 0 日のコンフィギュレーション ファイルの準備」を参照してください。
  - **管理インターフェイス**：第 0 日用構成の詳細を指定することを選択する場合は、管理インターフェイスの詳細を指定する必要があります。これは DHCP を使用するように設定する必要があります。
  - **データインターフェイス**：データインターフェイスの IP アドレスは、その情報を第 0 日用構成の一部として指定した場合にのみ割り当てられ、設定されます。データインターフェイスは、DHCP を使用するように設定できます。または、接続するネットワーク インターフェイスがすでに作成されていて、IP アドレスがわかっている場合は、第 0 日用構成で IP アドレスの詳細を指定できます。
  - **第 0 日用構成なし**：第 0 日用構成を指定せずに ASA 仮想 を導入すると、ASA 仮想 はデフォルトの ASA 仮想 構成を適用し、AWS メタデータサーバーから接続されたインターフェイスの IP アドレスを取得し、IP アドレスを割り当てます（データインターフェイスに IP アドレスは割り当てられますが、ENI はダウンします）。Management0/0 インターフェイスが起動し、DHCP アドレスで設定された IP アドレスを取得します。Amazon EC2 および Amazon VPC の IP アドレッシングについては、「VPC での IP アドレッシング」を参照してください。

    第 0 日用構成の例：

```
! ASA Version 9.x.1.200
!
interface management0/0
management-only
nameif management
security-level 100
ip address dhcp setroute
ipv6 enable
ipv6 address dhcp default
no shutdown
!
!
GWLB facing VTEP interface
interface TenGigabitEthernet0/0
nameif data-interface-in
security-level 100
ip address dhcp
no shut

!
Internet-facing outside interface
interface TenGigabitEthernet0/1
```

```
nameif data-interface-out
security-level 0
ip address dhcp
no shut

nve 1
encapsulation geneve
source-interface data-interface-in
interface vni1
proxy dual-arm
nameif vni-in
security-level 0
vtep-nve 1
! NAT for internet-bound traffic
nat (vni-in, data-interface-out) source dynamic any interface
!Default route to internet gateway= 10.1.200.1 (Outside gateway)
!Route East-West traffic (Application subnet CIDR) back to vni interface (U-turn)
route data-interface-out 0.0.0.0 0.0.0.0 10.1.200.1
route vni-in 192.168.1.0 255.255.255.0 10.1.100.1 1
!
mtu data-interface-in 1826
jumbo-frame reservation
same-security-traffic permit inter-interface
same-security-traffic permit intra-interface


crypto key generate rsa modulus 2048
ssh 0 0 management
ssh ::/0 management
ssh timeout 60
ssh version 2
username admin password Q1w2e3r4 privilege 15
username admin attributes
service-type admin
aaa authentication ssh console LOCAL
!
same-security-traffic permit inter-interface
same-security-traffic permit intra-interface
access-list allow-all extended permit ip any any
access-list allow-all extended permit ip any6 any6
access-group allow-all global
!
interface G0/0
nameif outside
ip address dhcp setroute
ipv6 enable
ipv6 address dhcp default
no shutdown
!
interface G0/1
nameif inside
ip address dhcp
ipv6 enable
ipv6 address dhcp default
no shutdown
!
```

- ストレージ：デフォルト値を保持します。
- タグインスタンス：デバイスを分類するため、多数のタグを作成できます。デバイスに名前を付けると、デバイスを容易に特定できます。

- セキュリティグループ：セキュリティグループを作成して名前を付けます。セキュリティ グループは、着信および発信トラフィックを制御するためのインスタンスの仮想ファイアウォールです。

  デフォルトでは、セキュリティ グループはすべてのアドレスに対して開かれています。ASA 仮想 のアクセスに使用するアドレスからの SSH 接続だけを許可するように、ルールを変更します。

  セキュリティグループでトラフィックを制御する方法については、AWS のドキュメント『Control traffic to your AWS resources using security groups』を参照してください。

- [高度な詳細（Advanced Details）] セクションを導入し、[ユーザーデータ（User data）] フィールドに、オプションで第 0 日用構成を入力できます。これは、ASA 仮想 の起動時に適用される ASA 仮想 構成を含むテキスト入力です。第 0 日用構成にスマート ライセンスなどの詳細情報を設定する方法の詳細については、「第 0 日のコンフィギュレーション ファイルの準備」を参照してください。

  - **管理インターフェイス**：第 0 日用構成を選択する場合は、管理インターフェイスの詳細を指定する**必要**があります。これは DHCP を使用するように設定する必要があります。
  - **データインターフェイス**：データインターフェイスの IP アドレスは、その情報を第 0 日用構成の一部として指定した場合にのみ割り当てられ、設定されます。データインターフェイスは、DHCP を使用するように設定できます。または、接続するネットワーク インターフェイスがすでに作成されていて、IP アドレスがわかっている場合は、第 0 日用構成で IP の詳細を指定できます。
  - **第 0 日用構成なし**：第 0 日用構成を指定**せず**に ASA 仮想 を導入すると、ASA 仮想 はデフォルトの ASA 仮想 構成を適用し、AWS メタデータサーバーから接続されたインターフェイスの IP を取得し、IP アドレスを割り当てます（データインターフェイスに IP は割り当てられますが、ENI はダウンします）。Management0/0 インターフェイスが起動し、DHCP アドレスで設定された IP を取得します。Amazon EC2 および Amazon VPC の IP アドレッシングについては、「VPC での IP アドレッシング」を参照してください。

  - [高度な詳細（Advanced Details）] の下で、デフォルトのログイン情報を追加します。デバイス名とパスワードの要件に合わせて、以下の例を変更してください。
  - [高度な詳細（Advanced Details）] で、IMDSv2 メタデータを有効にします。

    1. [メタデータにアクセス可能（Metadata accessible）] ドロップダウンリストから [有効化（Enabled）] を選択します。
    2. [メタデータバージョン（Metadata version）] ドロップダウンリストから [V2のみ（トークンが必要）（V2 only (token required)）] を選択します。

    次の手順を実行して、AWS CLI から IMDSv2 を有効にすることもできます。

  - AWS CLI コンソールを開き、次の引数を追加して IMDSv2 必須モードを有効にします。
    --metadata-options "HttpEndpoint=enabled,HttpTokens=required”

    IMDSv2 の設定例：

```
aws ec2 run-instances \,
--image-id ami-0abcdef1234567890 \
--instance-type c5x.large \
...
--metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

- 設定を確認し、[Launch] をクリックします。

ステップ 4 キー ペアを作成します。

注意
キーペアにわかりやすい名前を付け、キーを安全な場所にダウンロードします。再度、ダウンロードすることはできません。キーペアを失った場合は、インスタンスを破棄し、それらを再度導入する必要があります。

ステップ 5 [インスタンスの起動（Launch Instance）] をクリックして、ASA 仮想 を導入します。

ステップ 6 **[My Account]** > **[AWS Management Console]** > **[EC2]** > **[Launch an Instance]** > **[My AMIs]** をクリックします。

ステップ 7 ASA 仮想 のインターフェイスごとに [送信元または宛先の確認（Source/Destination Check）] が無効になっていることを確認します。

AWS のデフォルト設定では、インスタンスはその IP アドレス（IPv4 および IPv6）のトラフィックのみを受信でき、インスタンスは独自の IP アドレス（IPv4 および IPv6）からのみトラフィックを送信できます。ASA 仮想 のルーテッドホップとしての動作を有効にするには、ASA 仮想 の各トラフィック インターフェイス（内部、外部、および DMZ）の [送信元または宛先の確認（Source/Destination Check）] を無効にする必要があります。

既存の ASA Virtual インスタンスへの IMDSv2 必須モードの設定

AWS にすでに展開されている ASA Virtual インスタンスの IMDSv2 必須モードを設定できます。

始める前に

IMDSv2 必須モードは、ASA Virtual バージョン 9.20.3 以降でのみサポートされています。展開またはインスタンスに IMDSv2 必須モードを設定する前に、既存のインスタンスの ASA Virtual バージョン（9.20.3 以降）で IMDSv2 API をサポートしていることを確認する必要があります。

手順

ステップ 1 http://aws.amazon.com/ にログインし、地域を選択します。

ステップ 2 **[EC2]** > **[インスタンス（Instances）]**の順にクリックします。

ステップ 3 インスタンスを右クリックし、**[インスタンス設定（Instance Settings）]** > **[インスタンスメタデータオプションの変更（Modify instance metadata options）]**を選択します。[インスタンスメタデータオプションの変更（Modify instance metadata options）] ダイアログボックスが表示されます。

ステップ 4 [インスタンスメタデータサービス（Instance metadata service）] セクションで、[有効化（Enable）] をクリックします。

ステップ 5 [IMDSv2] オプションで、[必須（Required）] をクリックします。

これにより、選択したインスタンスの IMDSv2 必須モードが有効になります。

ステップ 6 [Save] をクリックします。

Amazon GuardDuty サービスと Threat Defense Virtual の統合

Amazon GuardDuty は AWS 環境において、VPC ログ、CloudTrail 管理イベントログ、CloudTrail S3 データイベントログ、DNS ログといったさまざまなソースからのデータを処理して、不正の可能性がある悪意のあるアクティビティを特定する監視サービスです。

Secure Firewall ASA Virtual と GuardDuty の統合について

シスコは、SSH を介した CLI を使用して Amazon GuardDuty サービスと Secure Firewall ASA Virtual を統合するソリューションを提供しています。

このソリューションでは、Amazon GuardDuty から受け取った脅威分析データや検出結果（脅威、攻撃などを生成する悪意のある IP）を使用して、その情報（悪意のある IP）を Secure Firewall ASA Virtual にフィードし、これらのソース（悪意のある IP）が発生源となる将来の脅威から基盤となるネットワークやアプリケーションを保護します。

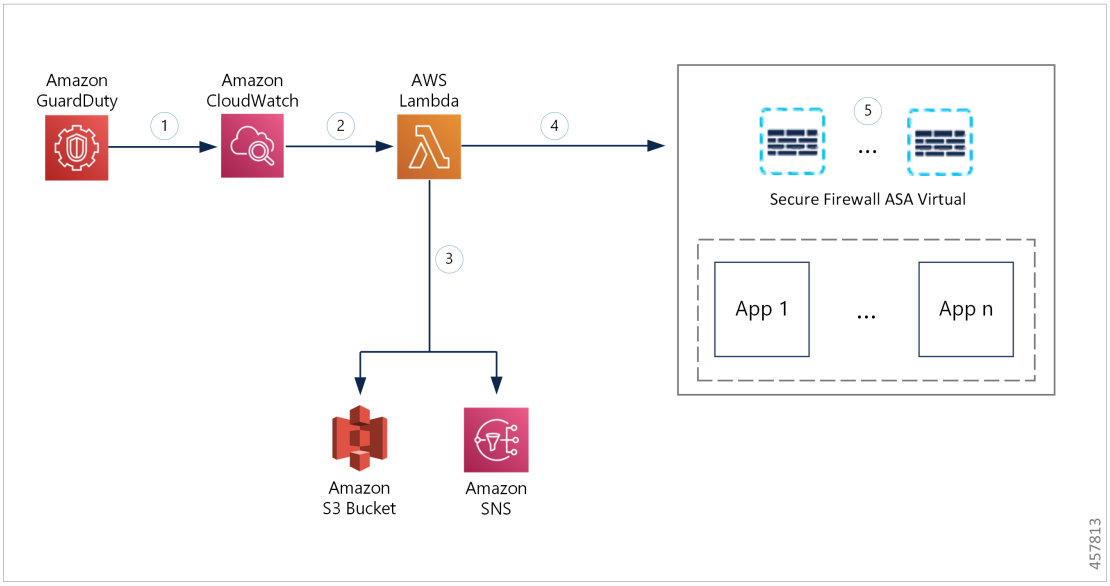
エンドツーエンドの手順

次の統合ソリューションとワークフローの図は、Amazon GuardDuty の Secure Firewall Threat Defense Virtual との統合を理解するのに役立ちます。

次のワークフロー図は、Amazon GuardDuty と ASA virtual の統合ソリューションを示しています。

ネットワーク オブジェクト グループを使用した Secure Firewall Device Manager との統合

次のワークフロー図は、ネットワーク オブジェクト グループを使用した Secure Firewall Device Manager と Amazon GuardDuty の統合ソリューションを示しています。



1	GuardDuty サービスは、悪意のあるアクティビティを検出すると、脅威の検出結果を CloudWatch に送信します。
2	CloudWatch イベントにより、AWS Lambda 関数がアクティブ化されます。
3	Lambda 関数は、S3 バケットのレポートファイル内に記載された悪意のあるホストを更新し、SNS 経由で通知を送信します。
4	Lambda 関数は、悪意のあるホスト IP アドレスを追加して Secure Firewall Device Manager のネットワーク オブジェクト グループを設定または更新します。
5	Secure Firewall Device Manager のアクセス コントロール ポリシーは、設定されたアクションに基づいてトラフィックを処理するように管理対象デバイスに指示します。たとえば、GuardDuty によって報告された悪意のあるホストからのトラフィックをブロックします。 このアクセス コントロール ポリシーは、Lambda 関数によって検出された悪意のある IP アドレスが追加されたネットワーク オブジェクト グループを使用します。

この統合の主要コンポーネント

コンポーネント	説明
Amazon GuardDuty	特定のリージョン（EC2、S3、IAM など）のさまざまな AWS リソースについて、脅威検出結果の生成を行う Amazon サービス。

Amazon Simple Storage Service（S3）	ソリューションに関連するさまざまなアーティファクトを保存するために使用される Amazon サービスは以下のとおりです。 • Lambda 関数の zip ファイル • Lambda レイヤの zip ファイル • ASA virtual 構成の入力ファイル（.ini） • Lambda 関数によって報告された悪意のある IP アドレスのリストが保存された出力レポートファイル（.txt）
Amazon CloudWatch	Amazon サービスは次の目的で使用されます。 • GuardDuty サービスで報告された検出結果についてモニタリングし、Lambda 関数をトリガーして検出結果を処理します。 • CloudWatch ロググループで Lambda 関数に関連するアクティビティをロギングします。
Amazon Simple Notification Service（SNS）	電子メール通知をプッシュするために使用される Amazon サービスです。この電子メール通知には、次の内容が含まれます。 • Lambda 関数によって正常に処理された GuardDuty 検出結果の詳細。 • Lambda 関数によって Cisco Secure Firewall Manager で実行された更新の詳細。 • Lambda 関数によって発生した重大なエラー。
AWS Lambda 関数	AWS サーバーレス コンピューティング サービスはイベントに応じてコードを実行し、基盤となるコンピューティングリソースを自動的に管理します。CloudWatch イベントルールが GuardDuty の検出結果に基づいて Lambda 関数をトリガーします。Lambda 関数はこの連携で以下を実行します。 • GuardDuty の検出結果を処理して、重大度、接続方向、悪意のある IP アドレスの存在など、必要なすべての基準が満たされていることを確認します。 • （設定に応じて）悪意のある IP アドレスを追加して、Cisco Secure Firewall Manager のネットワーク オブジェクト グループを更新します。 • S3 バケットのレポートファイルで悪意のある IP アドレスを更新します。 • Cisco Secure Firewall の管理者に対して、さまざまなマネージャの更新やエラーについて通知します。

CloudFormation テンプレート	AWS での連携に必要なさまざまなリソースを展開するために使用されます。 CloudFormation テンプレートには、次のリソースが含まれています。 • **AWS::SNS::Topic**：電子メール通知をプッシュするための SNS トピック。 • **AWS::Lambda::Function, AWS::Lambda::LayerVersion**：Lambda 関数とレイヤファイル。 • **AWS::Events::Rule**：GuardDuty の検出結果イベントに基づいて Lambda 関数をトリガーする CloudWatch イベントルール。 • **AWS::Lambda::Permission**：Lambda 関数をトリガーする CloudWatch イベントルールのアクセス許可。 • **AWS::IAM::Role、AWS::IAM::Policy**：各種 AWS リソースの Lambda 関数へのさまざまなアクセス許可を付与する IAM ロールとポリシーリソース。 このテンプレートは、展開をカスタマイズするためのユーザー入力を取り込みます。

サポートされるソフトウェア プラットフォーム

- GuardDuty 統合ソリューションは、SSH を介した CLI を使用して管理される Secure Firewall ASA Virtual に適用できます。
- Lambda 関数を使用して、Secure Firewall ASA Virtual 内のネットワーク オブジェクト グループを更新できます。Lambda 関数がパブリック IP アドレスを介して Secure Firewall ASA Virtual に接続できることを確認してください。

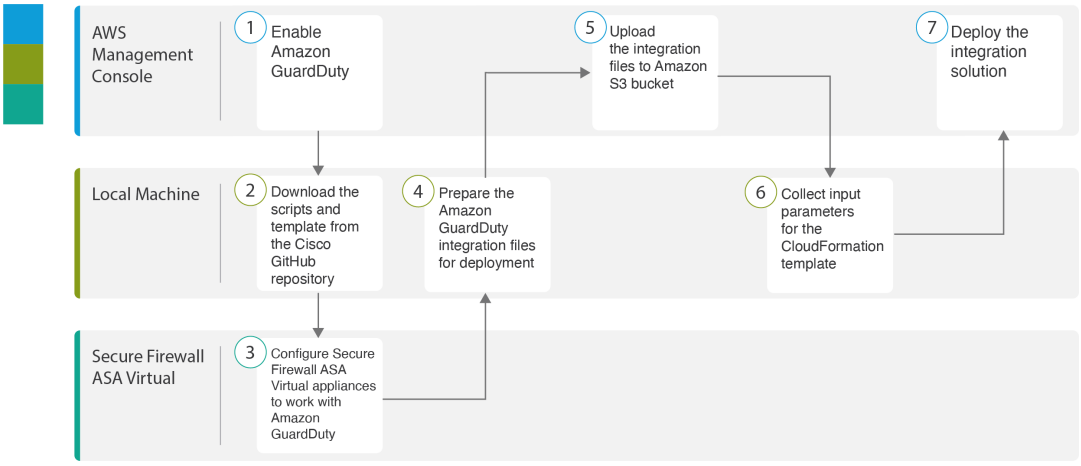
Amazon GuardDuty と Secure Firewall ASA Virtual の統合のガイドラインと制限事項

- Lambda 関数は、悪意のある IP アドレスが含まれるネットワーク オブジェクト グループの更新のみを実行します。要件に応じて、不要なトラフィックをブロックするアクセスルールとアクセスポリシーを作成します。
- この統合で使用される AWS のサービスはリージョン固有です。異なるリージョンの GuardDuty 検出結果を使用する場合は、リージョン固有のインスタンスを展開する必要があります。

- SSH 経由で CLI を使用して、ASA Virtual のアップデートを設定できます。ASDM、CSM、および CDO はサポートされていません。
- パスワードベースのログインのみを使用できます。他の認証方式はサポートされていません。
- 入力ファイルで暗号化されたパスワードを使用している場合は、次の点に注意してください。
  - 対称 KMS キーを使用した暗号化のみがサポートされます。
  - すべてのパスワードは、Lambda 関数にアクセス可能な単一の KMS キーを使用して暗号化する必要があります。

Amazon GuardDuty との統合 ASA 仮想

次のタスクを実行して、Amazon GuardDuty と ASA 仮想 を統合します。



	ワークスペース	手順
1	AWS 管理コンソール	AWS での Amazon GuardDuty サービスの有効化 （20 ページ）
2	Local Machine	Secure Firewall ASA Virtual と Amazon GuardDuty ソリューションテンプレートのダウンロード （20 ページ）
3	ASA 仮想	Amazon GuardDuty と連携するように管理対象デバイスを設定する （21 ページ）
4	Local Machine	展開に向けた Amazon GuardDuty リソースファイルの準備 （23 ページ）

	ワークスペース	手順
5	AWS 管理コンソール	Amazon Simple Storage Service へのファイルのアップロード （27 ページ）
6	Local Machine	CloudFormation テンプレートの入力パラメータの収集 （28 ページ）
7	AWS 管理コンソール	スタックの展開 （30 ページ）

AWS での Amazon GuardDuty サービスの有効化

ここでは、AWS で Amazon GuardDuty サービスを有効にする方法について説明します。

始める前に

すべての AWS リソースが同じリージョンにあることを確認します。

手順

ステップ 1 https://aws.amazon.com/marketplace（Amazon マーケットプレイス）に移動してサインインします。

ステップ 2 [サービス（Services）] > [GuardDuty] を選択します。

ステップ 3 [GuardDuty] ページで [利用を開始する（Get Started）] をクリックします。

ステップ 4 [GuardDutyの有効化（Enable GuardDuty）] をクリックして、Amazon GuardDuty サービスを有効にします。

GuardDuty の有効化の詳細については、AWS ドキュメントの『Getting started with GuardDuty』[英語] を参照してください。

次のタスク

Cisco GitHub リポジトリから Amazon GuardDuty ソリューションファイル（テンプレートとスクリプト）をダウンロードします。を参照してください。Secure Firewall ASA Virtual と Amazon GuardDuty ソリューションテンプレートのダウンロード （20 ページ）

Secure Firewall ASA Virtual と Amazon GuardDuty ソリューションテンプレートのダウンロード

Amazon GuardDuty ソリューションに必要なファイルをダウンロードします。Secure Firewall ASA Virtual の該当するバージョン用の導入スクリプトとテンプレートは、次の Cisco GitHub リポジトリから入手できます。

https://github.com/CiscoDevNet/cisco-asav

以下は、Cisco GitHub リポジトリリソースのリストです。

ファイル	説明
READ.MD	ReadMe ファイル
configuration/	Secure Firewall ASA Virtual 構成ファイルテンプレート。
images/	Secure Firewall ASA Virtual および Amazon GuardDuty 統合ソリューションの図が格納されています。
lambda/	Lambda 関数の Python ファイル。
templates/	導入用の CloudFormation テンプレート

Amazon GuardDuty と連携するように管理対象デバイスを設定する

Lambda 関数は Amazon GuardDuty の検出結果を処理し、CloudWatch イベントをトリガーした悪意のある IP アドレスを特定します。次に、Lambda 関数を使用して、悪意のある IP アドレスが含まれた ASAv のネットワーク オブジェクト グループを更新します。次に、このネットワーク オブジェクト グループを使用してトラフィックを処理するアクセス コントロール ポリシーを設定できます。

ネットワーク オブジェクト グループの作成

ASA virtual で Lambda 関数のネットワーク オブジェクト グループを設定または作成して、Amazon GuardDuty によって検出された悪意のある IP アドレスを更新する必要があります。

Lambda 関数でネットワーク オブジェクト グループを設定しない場合、デフォルト名 **aws-gd-suspicious-hosts** のネットワーク オブジェクト グループが Lambda 関数によって作成され、悪意のある IP アドレスが更新されます。

Secure Firewall ASA Virtual でのネットワーク オブジェクト グループの作成

Secure Firewall ASA Virtual で Lambda 関数のネットワーク オブジェクト グループを作成し、Amazon GuardDuty によって検出された悪意のある IP アドレスを更新する必要があります。

Lambda 関数でネットワーク オブジェクト グループを設定しない場合、デフォルト名 *aws-gd-suspicious-hosts* のネットワーク オブジェクト グループが Lambda 関数によって作成され、悪意のある IP アドレスが更新されます。

最初に、ACL ルールでネットワーク オブジェクト グループを使用するには、ダミーの IP アドレスを使用してオブジェクトグループを作成する必要があります。1 つの ASAv で複数のネットワーク オブジェクト グループを作成できます。

ネットワーク オブジェクト グループとアクセスポリシーの詳細については、『Cisco ASA Series Firewall CLI Configuration Guide』を参照してください。

ネットワーク オブジェクト グループを作成するには、次の手順を実行します。

手順

ステップ 1 Secure Firewall ASA Virtual にログインします。

ステップ 2 説明を使用してネットワーク オブジェクト グループを作成します。この例では、作成されたネットワーク オブジェクト グループにダミーのホスト IP アドレス 12.12.12.12 が追加されます。

例：

```
hostname(config)# object-group network aws-gd-suspicious-hosts
hostname(config)# description Malicious Hosts reported by AWS GuardDuty
hostname(config)# network-object host 12.12.12.12
```

ステップ 3 ネットワーク オブジェクト グループを使用してトラフィックを処理するためのアクセスポリシーやアクセス制御ルールを作成または更新します。\

ヒント
また、Lambda 関数を使用して悪意のある IP アドレスが含まれるネットワーク オブジェクト グループを更新していることを確認した後に、アクセス コントロール ポリシーやアクセス制御ルールを作成または更新することもできます。

例：

```
hostname(config)# access-list out-iface-access line 1 extended deny ip object-group
aws-gd-suspicious-hosts any
```

ASAv で Lambda 関数を利用するためのユーザーアカウントの作成

Lambda 関数では、設定の更新を処理するために ASAv の専用ユーザーが必要です。権限レベル 15 は、ユーザーがすべての権限を持っていることを保証します。

ユーザー作成の詳細については、『Cisco ASA Series Firewall CLI Configuration Guide』を参照してください。

手順

ステップ 1 ユーザーを作成します。

username *name* [**password** *password*] **privilege** *level*

例：

```
hostname(config)# username aws-gd password MyPassword@2021 privilege 15
```

ステップ 2 ユーザー名属性を設定します。

username *username* **attributes**

例：

```
hostname(config)# username aws-gd attributes
```

ステップ 3 すべてのサービスへの管理者レベルのアクセスをユーザーに付与します。

service-type admin

例：

```
hostname(config)# service-type admin
```

（任意）パスワードの暗号化

必要に応じて、入力構成ファイルに暗号化されたパスワードを指定できます。プレーンテキスト形式でパスワードを指定することもできます。

Lambda関数にアクセスできる単一のKMSキーを使用して、すべてのパスワードを暗号化します。**aws kms encrypt --key-id** <*KMS-ARN*> **--plaintext** <*password*> コマンドを使用して暗号化されたパスワードを生成します。このコマンドを実行するには、AWS CLI をインストールして設定する必要があります。

（注） パスワードが対称 KMS キーを使用して暗号化されていることを確認します。

AWS CLI については、AWS のコマンド ライン インタフェース [英語] を参照してください。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 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 <password>
{
    "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。

展開に向けた Amazon GuardDuty リソースファイルの準備

Amazon GuardDuty ソリューションの展開リソースファイルは、Cisco GitHub リポジトリで入手できます。

AWS に Amazon GuardDuty ソリューションを展開する前に、次のファイルを準備する必要があります。

- Cisco Secure FirewallASA virtual マネージャの構成入力ファイル
- Lambda 関数の zip ファイル
- Lambda レイヤの zip ファイル

構成入力ファイルの準備

構成テンプレートでは、Amazon GuardDuty ソリューションと連携する ASAv の詳細を定義する必要があります。

始める前に

- 構成ファイルにユーザーアカウントの詳細を指定する前に、デバイスマネージャのユーザーアカウントを認証および検証します。
- 構成ファイルで ASAv を 1 つだけ設定してください。ASAv を複数設定すると、Lambda 関数はファイルで設定されたすべての ASAv を同時に更新する可能性があり、その結果、競合状態や非確定的な動作が発生します。
- ASAv の IP アドレスと名前をメモしておく必要があります。
- ASAv でこれらのネットワーク オブジェクト グループにアクセスして更新するには、Lambda 関数の管理者権限を持つユーザーアカウントを作成しておく必要があります。

手順

ステップ 1 Amazon GuardDuty リソースファイルをダウンロードしたローカルマシンにログインします。

ステップ 2 asav-template > configuration フォルダを参照します。

ステップ 3 テキストエディタツールで `asav-manager-config-input.ini` ファイルを開きます。このファイルには、Amazon GuardDuty ソリューションの統合と展開を計画している ASAv の詳細を入力する必要があります。

ステップ 4 次の ASAv パラメータを入力します。

パラメータ	説明
[asav-1]	セクション名：ファイル内の一意の ASAv 識別子
public-ip	ASAv のパブリック IP アドレス
ユーザー名	ASAv にログインするユーザー名。
パスワード	ASAvにログインするパスワード。パスワードには、プレーンテキスト形式、または KMS を使用して暗号化された文字列を使用できます。

パラメータ	説明
enable_password	ASAv のイネーブルパスワード。パスワードには、プレーンテキスト形式、または KMS を使用して暗号化された文字列を使用できます。
object-group-name	Lambda 関数が悪意のあるホスト IP を追加して更新するネットワーク オブジェクト グループの名前。複数のネットワーク オブジェクト グループ名を入力する場合は、カンマ区切り値になっていることを確認してください。

ステップ 5 `asav-manager-config-input.ini` ファイルを保存して閉じます。

次のタスク

Lambda 関数のアーカイブファイルを作成します。

Lambda 関数のアーカイブファイルの準備

ここでは、Linux 環境で Lambda 関数ファイルをアーカイブする方法について説明します。

（注） アーカイブプロセスは、ファイルのアーカイブを実行するローカルマシンのオペレーティングシステムによって異なる場合があります。

始める前に

Linux ホストで Python バージョン 3.6 以降がインストールされた Ubuntu バージョン 18.04 を実行していることを確認します。

手順

ステップ 1 Amazon GuardDuty リソースをダウンロードしたローカルマシンで CLI コンソールを開きます。

ステップ 2 /lambda フォルダに移動し、ファイルをアーカイブします。以下は、Linux ホストからのサンプルトランスクリプトです。

```
$ cd lambda
$ zip asav-gd-lambda.zip *.py
adding: aws.py (deflated 71%)
adding: asav.py (deflated 79%)
adding: main.py (deflated 73%)
adding: utils.py (deflated 65%)
$
```

zip ファイル `asav-gd-lambda.zip` が作成されます。

ステップ 3 終了して CLI コンソールを閉じます。

次のタスク

zip ファイル `asav-gd-lambda.zip` を使用して、Lambda レイヤの zip ファイルを作成します。

Lambda レイヤファイルの準備

ここでは、Linux 環境で Lambda レイヤファイルをアーカイブする方法について説明します。

（注） アーカイブプロセスは、ファイルのアーカイブを実行するローカルマシンのオペレーティングシステムによって異なる場合があります。

手順

ステップ 1 Amazon GuardDuty リソースをダウンロードしたローカルマシンで CLI コンソールを開きます。

ステップ 2 CLI コンソールで次のアクションを実行します。

以下は、Python 3.9 がインストールされている Ubuntu 22.04 などの Linux ホストでのサンプルトランスクリプトです。

```
$ mkdir -p layer
$ virtualenv -p /usr/bin/python3.9 ./layer/
$ source ./layer/bin/activate
$ pip3.9 install cffi==1.15.0
$ pip3.9 install cryptography==37.0.2
$ pip3.9 install paramiko==2.7.1
$ mkdir -p ./python/.libs_cffi_backend/
$ cp -r ./layer/lib/python3.9/site-packages/* ./python/
$ zip -r asav-gd-lambda-layer.zip ./python
```

zip ファイル `asav-gd-lambda-layer.zip` が作成されます。

Lambda レイヤを作成するには、Python 3.9 とその依存関係をインストールする必要があることに注意してください。

以下は、Ubuntu 22.04 などの Linux ホストに Python 3.9 をインストールするためのサンプルトランスクリプトです。

```
$ 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 終了して CLI コンソールを閉じます。

次のタスク

Amazon S3 バケットでは、Cisco Secure FirewallASA virtual の構成ファイル、Lambda 関数の zip ファイル、および Lambda レイヤの zip ファイルをアップロードする必要があります。Amazon Simple Storage Service へのファイルのアップロード （27 ページ） を参照してください

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

すべての Amazon GuardDuty ソリューション アーティファクトを準備したら、AWS ポータルの Amazon Simple Storage Service（S3）バケットフォルダにファイルをアップロードする必要があります。

手順

ステップ 1 https://aws.amazon.com/marketplace（Amazon マーケットプレイス）に移動してサインインします。

ステップ 2 Amazon S3 コンソールを開きます。

ステップ 3 Amazon GuardDuty アーティファクトをアップロードするための Amazon S3 バケットを作成します。Amazon S3 の作成 [英語] を参照してください。

ステップ 4 次の Amazon GuardDuty アーティファクトを Amazon S3 バケットにアップロードします。

- Cisco Secure Firewall ASA virtual構成ファイル：`asav-config-input.ini`

  （注）
  管理センターでセキュリティ インテリジェンスのネットワーク フィード メソッドを使用して Amazon GuardDuty ソリューションを展開する場合、このファイルをアップロードする必要はありません。

- Lambda レイヤ zip ファイル：`asav-gd-lambda-layer.zip`
- Lambda 関数 zip ファイル：`asav-gd-lambda.zip`

次のタスク

Amazon GuardDuty リソースの展開に使用する CloudFormation テンプレートを準備します。CloudFormation テンプレートの入力パラメータの収集 （28 ページ）を参照してください。

CloudFormation テンプレートの入力パラメータの収集

シスコでは、AWS の Amazon GuardDuty ソリューションに必要なリソースを展開する際に使用する CloudFormation テンプレートを提供しています。展開する前に、次のテンプレートパラメータの値を収集します。

手順

Template Parameters

パラメータ	説明	例
展開名*	このパラメータに入力する名前は、Cloud Formation テンプレートによって作成されるすべてのリソースのプレフィックスとして使用されます。	**cisco-asav-gd**
GD 検出結果の最小の重大度レベル*	Amazon GuardDuty の検出結果で処理の対象となる最小重大度レベルは、**1.0** から **8.9** の範囲にする必要があります。報告された検出結果の重大度が最小範囲よりも低い場合は無視されます。 重大度の分類は次のとおりです。 • 低：1.0 ～ 3.9 中：4.0 ～ 6.9 高：7.0 ～ 8.9	**4.0%**
管理者の電子メール ID*	Cisco Secure Firewall ASA virtual の Lambda 関数によって実行された更新に関する通知を受信する Cisco Secure Firewall ASA virtual の管理者の電子メールアドレス。	**abc@xyz.com**
S3 バケット名*	Amazon GuardDuty アーティファクトファイル （Lambda 関数の zip ファイル、Lambda レイヤの zip ファイル、および Cisco Secure FirewallASA virtual 設定マネージャファイル）が格納された Amazon S3 バケットの名前。	例：**asav-gd-bucket**

パラメータ	説明	例
S3 バケットフォルダ/パスプレフィックス	構成ファイルが保存されている Amazon S3 バケットのパスまたはフォルダ名。フォルダがない場合は、このフィールドを空白のままにします。	例：「」または「**cisco/asav-gd/**」
Lambda レイヤの zip ファイル名*	Lambda レイヤの zip ファイル名。	例：`asav-gd-lambda-layer.zip`
Lambda 関数の zip ファイル名*	Lambda 関数の zip ファイル名。	例：`asav-gd-lambda.zip`
Cisco Secure Firewall ASA virtual マネージャの構成ファイル名	Cisco Secure Firewall ASA virtual のマネージャ設定の詳細が保存された ***.ini** ファイル（パブリック IP、ユーザー名、パスワード、デバイスタイプ、ネットワーク オブジェクト グループ名など）。	例：`asav-config-input.ini`
パスワードの暗号化に使用される KMS キーの ARN	既存の KMS（パスワードの暗号化に使用される AWS KMS キー）の ARN。Cisco Secure FirewallASA virtual の構成入力ファイルでプレーンテキストパスワードが指定されている場合は、このパラメータを空のままにしておくことができます。指定する場合、Cisco Secure FirewallASA virtual の構成入力ファイルに記載されているすべて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には、指定された ARN のみを使用する必要があります。暗号化パスワードの生成：aws kms encrypt --key-id <KMS ARN> --plaintext <password>	例：**arn:aws:kms:<region>:<awsaccountid>:key/<keyid>**
デバッグログの有効化/無効化*	CloudWatch で Lambda 関数のデバッグログを有効または無効にします。	例：**enable** または **disable**

*：必須フィールド

次のタスク

CloudFormationテンプレートを使用してスタックを展開します。スタックの展開 （30ページ）を参照してください

スタックの展開

Amazon GuardDuty ソリューションを導入するためのすべての前提条件プロセスを完了した後に、AWS CloudFormation スタックを作成します。対象ディレクトリのテンプレートファイル（`templates/cisco-asav-gd-integration.yaml`）を使用し、「CloudFormationテンプレートの入力パラメータの収集」で収集したパラメータを指定します。

手順

ステップ 1 AWS コンソールにログインします。

ステップ 2 [サービス（Services）]>[CloudFormation]>[スタック（Stacks）]>[スタックの作成（Create stack）]（新しいリソースを使用）>[テンプレートの準備（Prepare template）]（テンプレートはフォルダ内にあります）>[テンプレートの指定（Specify template）]>[テンプレートソース（Template source）]（ターゲットディレクトリ `templates/cisco-asav-gd-integration.yaml` からテンプレートファイルをアップロード）>[スタックの作成（Create Stack）] の順に操作を行います。

AWS でスタックを展開する方法の詳細については、AWS ドキュメント [英語] を参照してください。

次のタスク

展開を検証します。展開の検証 （31 ページ） を参照してください。

また、Amazon GuardDuty によって報告された脅威検出の更新に関する電子メール通知を受信するように登録します。電子メール通知の登録 （30 ページ）を参照してください。

電子メール通知の登録

CloudFormation テンプレートでは、GuardDutyの検出結果の更新に関する通知を受信するように、電子メール ID が設定されています。これは Lambda 関数によって実行されます。AWS に CloudFormation テンプレートを展開すると、Amazon Simple Notification Service（SNS）サービスを介してこの電子メール ID に電子メール通知が送信され、通知の更新を登録するように要求されます。

手順

ステップ 1 電子メール通知を開きます。

ステップ 2 電子メール通知で利用可能な**サブスクリプション**リンクをクリックします。

次のタスク

展開を検証します。展開の検証 （31 ページ）を参照してください。

展開の検証

この項で説明されているように、AWS には Amazon GuardDuty ソリューションを検証するオプションがあります。CloudFormation の展開が完了したら、以下に示す展開の検証手順を実行できます。

始める前に

展開を検証するためのコマンドを実行するには、AWS コマンド ライン インターフェイス（CLI）がインストールおよび設定されていることを確認します。AWS CLI のドキュメントについては、AWS のコマンド ライン インターフェイス [英語] を参照してください。

手順

ステップ 1 AWS 管理コンソールにログインします。

ステップ 2 [サービス（Services）] > [GuardDuty] > [設定（Settings）] > [GuardDutyのが概要（About GuardDuty）] > [ディテクタID（Detector ID）] に移動して、ディテクタ ID を書き留めます。

このディテクタ ID は、Amazon GuardDuty のサンプル検出結果を生成するために必要です。

ステップ 3 AWS CLI コンソールを開き、次のコマンドを実行して Amazon GuardDuty のサンプル検出結果を生成します。

```
aws guardduty create-sample-findings --detector-id <detector-id> --finding-types
UnauthorizedAccess:EC2/MaliciousIPCaller.Custom
```

```
aws guardduty create-sample-findings --detector-id <detector-id> --finding-types
UnauthorizedAccess:EC2/MaliciousIPCaller.Custom
```

ステップ 4 Amazon GuardDuty コンソールの結果リストでサンプルの検出結果を確認します。

サンプル検出結果には、プレフィックス **[sample]** が含まれています。接続方向、リモート IP アドレスなどの属性を参照して、サンプル検出結果の詳細を確認できます。

ステップ 5 Lambda 関数が実行されるのを待ちます。

Lambda 関数がトリガーされたら、以下を確認します。

- 受信した Amazon GuardDuty の検出結果と、Lambda 関数によって実行された Cisco Secure Firewall ASA virtual の更新に関する詳細が記載された電子メール通知。

- レポートファイルが Amazon S3 バケットに生成されているかどうかを確認します。レポートファイルには、サンプルの Amazon GuardDuty の検出結果によって報告された悪意のある IP アドレスが含まれています。レポートファイル名は、`<deployment-name>-report.txt` の形式になっています。
- ：設定されたマネージャ（ Cisco Secure Firewall ASA virtual）で、サンプルの検出結果から更新された悪意のある IP アドレスを追加してネットワーク オブジェクト グループが更新されていることを確認します。

ステップ 6 [AWSコンソール（AWS Console）] > [サービス（Services）] > [CloudWatch] > [ログ（Logs）] > [ロググループ（Log groups）] に移動し、ロググループを選択して、CloudWatch コンソールで Lambda ログを確認します。CloudWatch のロググループ名は、`<deployment-name>-lambda` の形式になっています。

ステップ 7 展開を検証した後、次のようにサンプル検出結果によって生成されたデータをクリーンアップすることを推奨します。

a) AWS コンソールから [サービス（Services）] > [GuardDuty] > [結果（Findings）] > [結果を選択（Select the finding）] > [アクション（Actions）] > [アーカイブ（Archive）] に移動して、サンプルの検出結果データを表示します。

b) ネットワーク オブジェクト グループに追加された悪意のある IP アドレスを削除して、キャッシュされたデータを Cisco Secure Firewall ASA virtual から消去します。

c) Amazon S3 バケットのレポートファイルをクリーンアップします。サンプルの検出結果で報告された悪意のある IP アドレスを削除することで、ファイルを更新できます。

既存のソリューション展開構成の更新

展開後に S3 バケットや S3 バケットフォルダとパスプレフィックス値を更新しないことを推奨します。ただし、展開したソリューションの構成を更新する必要がある場合は、AWS コンソールの [CloudFormation] ページで [スタックの更新（Update Stack）] オプションを使用します。

以下のパラメータを更新できます。

パラメータ	説明
Cisco Secure FirewallASA virtual マネージャの構成ファイル名	Amazon S3 バケットの構成ファイルを追加または更新します。以前のファイルと同じ名前でファイルを更新できます。構成ファイル名が変更された場合は、AWS コンソールの [スタックの更新（Update stack）] オプションを使用して、このパラメータを更新できます。
GD 検出結果の最小の重大度レベル*	AWS コンソールの [スタックの更新（Update stack）] オプションを使用して、パラメータ値を更新します。
管理者の電子メール ID*	AWS コンソールの [スタックの更新（Update stack）] オプションを使用して、電子メール

パラメータ	説明
	ID のパラメータ値を更新します。SNS サービスコンソールを介して電子メールのサブスクリプションを追加または更新することもできます。
S3 バケット名*	Amazon S3 バケット内の zip ファイルを新しい名前で更新してから、AWS コンソールの [スタックの更新（Update Stack）] オプションを使用してパラメータを更新します。
Lambda レイヤの zip ファイル名*	Amazon S3 バケット内の Lambda レイヤ zip ファイル名を新しい名前で更新してから、AWS コンソールの [スタックの更新（Update stack）] オプションを使用して、このパラメータ値を更新します。
Lambda 関数の zip ファイル名*	Amazon S3 バケット内の Lambda 関数 zip ファイルを新しい名前で更新してから、AWS コンソールの [スタックの更新（Update stack）] オプションを使用して、このパラメータ値を更新します。
パスワードの暗号化に使用される KMS キーの ARN	AWS コンソールの [スタックの更新（Update stack）] オプションを使用して、パラメータ値を更新します。
デバッグログの有効化/無効化*	AWS コンソールの [スタックの更新（Update stack）] オプションを使用して、パラメータ値を更新します。

手順

ステップ 1 AWS 管理コンソールに進みます。

ステップ 2 必要に応じて、新しいバケットとフォルダを作成します。

ステップ 3 以下に示すアーティファクトが古いバケットから新しいバケットにコピーされていることを確認します。

- Cisco Secure Firewall ASA virtual構成ファイル：`asav-config-input.ini`
- Lambda レイヤ zip ファイル：`asav-gd-lambda-layer.zip`
- Lambda 関数 zip ファイル：`asav-gd-lambda.zip`
- Output レポートファイル：`<deployment-name>-report.txt`

ステップ 4 パラメータ値を更新するには、**Services > CloudFormation > Stacks > > Update (Update Stack) > Prepare template > Use current template > Next >** <*update parameters*>**> Update Stack** に移動します。

パフォーマンスの調整

VPN の最適化

AWS c5 インスタンスは、以前の c3、c4、および m4 インスタンスよりもはるかに高いパフォーマンスを提供します。c5 インスタンスファミリでのおおよその RA VPN スループット（AES-CBC 暗号化による 450B TCP トラフィックを使用する DTLS）は、以下のようである必要があります。

- 0.5 Gbps（c5.large）
- 1 Gbps（c5.xlarge）
- 2 Gbps（c5.2xlarge）

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。