Google Cloud Platform への Threat Defense Virtual の展開

Google Cloud Platform（GCP）上で Threat Defense Virtual を展開できます。GCP は、Google が提供する可用性の高いホスト環境でアプリケーションを実行できるパブリック クラウド コンピューティング サービスです。

GCP コンソールの **[ダッシュボード（Dashboard）]** に GCP プロジェクト情報が表示されます。

- まだ選択していない場合は、**[ダッシュボード（Dashboard）]** で GCP プロジェクトを選択してください。
- ダッシュボードにアクセスするには、**[ナビゲーションメニュー（Navigation menu）]** > **[ホーム（Home）]** > **[ダッシュボード（Dashboard）]** をクリックします。

GCP コンソールにログインし、GCP Marketplace で Cisco Firepower NGFW 仮想ファイアウォール（NGFWv）製品を検索し、Threat Defense Virtual インスタンスを起動します。次の手順では、GCP 環境を準備し、Threat Defense Virtual インスタンスを起動して Threat Defense Virtual を展開する方法について説明します。

- 概要 （2 ページ）
- エンドツーエンドの手順 （3 ページ）
- 前提条件 （4 ページ）
- Threat Defense Virtual および GCP のガイドラインと制限事項 （5 ページ）
- データインターフェイス への NIC マッピング （7 ページ）
- ネットワークトポロジの例 （8 ページ）
- Secure Firewall Threat Defense Virtual デバイスの管理方法 （9 ページ）
- GCP 環境の設定 （10 ページ）
- ファイアウォールルールの作成 （10 ページ）
- Threat Defense Virtual の導入 （11 ページ）
- 外部 IP を使用した Threat Defense Virtual インスタンスへの接続 （13 ページ）
- シリアルコンソールを使用した Threat Defense Virtual インスタンスへの接続 （14 ページ）
- Gcloud を使用した Threat Defense Virtual インスタンスへの接続 （15 ページ）
- Auto Scale ソリューション （15 ページ）

• 導入パッケージのダウンロード （18 ページ）
• システム要件 （19 ページ）
• 前提条件 （23 ページ）
• Auto Scale ソリューションの展開 （31 ページ）
• Auto Scale ロジック （38 ページ）
• ロギングとデバッグ （38 ページ）
• トラブルシューティング （39 ページ）

概要

Threat Defense Virtual は、物理的な Secure Firewall Threat Defense（旧称 Firepower Threat Defense）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。Threat Defense Virtualは、パブリック GCP に展開できます。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。

システム要件

Threat Defense Virtual のニーズに合わせて Google 仮想マシンのタイプとサイズを選択します。現在、Threat Defense Virtual はコンピューティング最適化マシンと汎用マシン（標準タイプ、大容量メモリタイプ、高性能 CPU タイプ）のいずれもサポートしています。

（注） サポートされるマシンタイプは、予告なく変更されることがあります。

表 *1*: サポートされるコンピューティング最適化マシンタイプ

コンピューティング最適化マシンタイプ	属性		
	vCPU	**RAM（GB）**	**vNIC**
c2-standard-4	4	16 GB	4
c2-standard-8	8	32 GB	8
c2-standard-16	16	64 GB	8

表 *2*: サポートされる汎用マシンタイプ

汎用マシンタイプ	属性		
	vCPU	**RAM（GB）**	**vNIC**
n1-standard-4	4	15	4
n1-standard-8	8	30	8

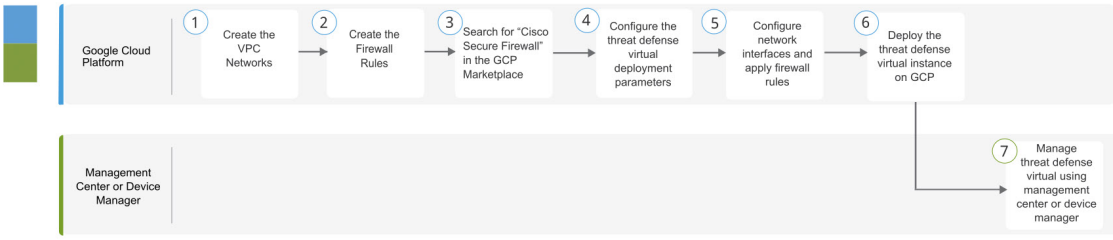
汎用マシンタイプ	属性		
	vCPU	**RAM（GB）**	**vNIC**
n1-standard-16	16	60	8
n2-standard-4	4	16	4
n2-standard-8	8	32	8
n2-standard-16	16	64	8
n2-highmem-4	4	32	4
n2-highmem-8	8	64	8

- Threat Defense Virtual には、少なくとも 4 つのインターフェイスが必要です。
- サポートされる vCPU の最大数は 16 です。

ユーザーは、GCP でアカウントを作成し、GCP Marketplace の Cisco Firepower NGFW 仮想ファイアウォール（NGFWv）製品を使用して VM インスタンスを起動し、GCP マシンタイプを選択します。

エンドツーエンドの手順

次のフローチャートは、Google Cloud Platform に Threat Defense Virtual を展開する際のワークフローを示しています。



	ワークスペース	手順
1	GCP	GCP 環境の設定：VPC ネットワークを作成します（**[VPCネットワーク（VPC Networks）]** > **[サブネット（Subnet）]** > **[リージョン（Region）]** > **[IPアドレス範囲（IP address range）]**）。
2	GCP	ファイアウォールルールの作成：ファイアウォールルールを作成します（**[ネットワーキング（Networking）]** > **[VPCネットワーク（VPC networks）]** > **[ファイアウォール（Firewall）]** > **[ファイアウォールルールの作成（Create Firewall Rule）]**）。

	ワークスペース	手順
3	GCP	Threat Defense Virtual の導入：GCP Marketplace で「Cisco Secure Firewall」を検索します。
4	GCP	Threat Defense Virtual の導入：Threat Defense Virtual の展開パラメータを設定します。
5	GCP	Threat Defense Virtual の導入：ネットワーク インターフェイスを設定し、ファイアウォールルールを適用します。
6	GCP	Threat Defense Virtual の導入：GCP に Threat Defense Virtual を展開します。
7	Management Center または Device Manager	Threat Defense Virtual を管理します。 • Management Center を使用した Threat Defense Virtual の管理 • Device Manager を使用した Threat Defense Virtual の管理

前提条件

- https://cloud.google.com で GCP アカウントを作成します。
- GCP プロジェクトを作成します。Google ドキュメントの『Creating Your Project』を参照してください。
- Cisco スマートアカウント。Cisco Software Central（https://software.cisco.com/）で作成できます。
- Threat Defense Virtual へのライセンス付与。
  - Management Center からセキュリティ サービスのすべてのライセンス資格を設定します。
  - ライセンスの管理方法の詳細については、『Cisco Secure Firewall Management Center アドミニストレーション ガイド』の「*Licensing*」の章を参照してください。
- Threat Defense Virtual のシステム要件については、『Cisco Secure Firewall Threat Defense Compatibility Guide』を参照してください。

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。
- トラフィック インターフェイス（2）：Threat Defense Virtual を内部の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。

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

サポートされる機能

- GCP Compute Engine での展開
- インスタンスあたり最大 16 個の vCPU
- ルーテッド モード（デフォルト）
- ライセンス：BYOL のみをサポート
- クラスタリング（7.2 以降）詳細については、『パブリッククラウドにおける Threat Defense Virtual のクラスタリング』を参照してください。
- Cisco Secure Firewall 7.1 以前のバージョンでは、 Management Center のみがサポートされています。Cisco Secure Firewall バージョン 7.2 以降では、Device Manager もサポートされます。

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 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。

表 *3 : 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、100Mbps	4 コア/8 GB	100Mbps	50
FTDv10、1Gbps	4 コア/8 GB	1Gbps	250
FTDv20、3Gbps	4 コア/8 GB	3 Gbps	250
FTDv30、5Gbps	8 コア/16 GB	5 Gbps	250
FTDv50、10Gbps	12 コア/24 GB	10 Gbps	750
FTDv100、16 Gbps	16 コア/32 GB	16 Gbps	10,000

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、『Cisco Secure Firewall Management Center アドミニストレーション ガイド』の「Licensing」の章を参照してください。

（注） vCPU/メモリの値を変更するには、最初に Threat Defense Virtual デバイスの電源をオフにする必要があります。

パフォーマンスの最適化

Threat Defense Virtual の最高のパフォーマンスを実現するために、VM とホストの両方を調整することができます。詳細については、「GCP での仮想化の調整と最適化」を参照してください。

Receive Side Scaling：Threat Defense Virtual は Receive Side Scaling（RSS）をサポートしています。これは、ネットワークアダプタによって複数のプロセッサコアにネットワーク受信トラフィックを分散するために使用されるテクノロジーです。バージョン 7.0 以降でサポートされています。詳細については、「Receive Side Scaling（RSS）用の複数の RX キュー」を参照してください。

送受信キューの割り当て

ネットワークパケットを処理するために、特定の数の受信（RX）および送信（TX）キューが各 vNIC に割り当てられます。Google Cloud では、使用されるネットワーク インターフェイスのタイプ（VirtIO または gVNIC）に基づき、アルゴリズムを使用して、vNIC ごとにデフォルトの数の RX および TX キューが割り当てられます。

vNIC にキューを割り当てるために GCP で使用される方法は次のとおりです。

- VirtIO：vCPU の数が vNIC の数で除算され、残りの値は破棄されます。

  たとえば、VM に 16 個の vCPU と 4 個の vNIC がある場合、vNIC ごとに割り当てられるキューの数は 16/4 = 4 です。

- gVNIC：vCPU の数が vNIC の数で除算され、結果がさらに 2 で除算されます。

  たとえば、VM に 128 個の vCPU と 2 個の vNIC がある場合、割り当てられるキューの数は 128/2/2 = 2 です。

また、Compute Engine API を使用して新しい VM を作成すると場合、各 vNIC に割り当てられるキューの数をカスタマイズできます。ただし、カスタマイズする場合は、次のルールに従う必要があります。

- 最小キュー数：vNIC ごとに 1 つ。
- 最大キュー数：この数は、ドライバタイプに基づいて、vCPU 数または vNIC あたりの最大キュー数のうち、小さい方です。
  - VirtIO またはカスタムドライバを使用している場合、最大キュー数は 32 です。

  - gVNIC を使用している場合、最大キュー数は 16 です。

- VM のすべての vNIC に割り当てられるキューの数をカスタマイズする場合、キュー割り当ての総数は、VM インスタンスに割り当てられた vCPU の数以下である必要があります。

デフォルトおよびカスタムキュー割り当ての詳細と例については、「デフォルトキューの割り当て」および「カスタムキューの割り当て」を参照してください。

Snort

- Snort のシャットダウンに時間がかかったり、VM が全体的に遅くなったりといった異常な動作が見られる場合や、特定のプロセスが実行されるときには、Threat Defense Virtual および VM ホストからログを収集します。全体的な CPU 使用率、メモリ、I/O 使用率、および読み取り/書き込み速度のログの収集は、問題のトラブルシューティングに役立ちます。
- Snort のシャットダウン時には、CPU と I/O の使用率が高くなります。十分なメモリがなく、専用の CPU がない単一のホスト上に多数の Threat Defense Virtual インスタンスが作成されている場合は、Snort のシャットダウンに時間がかかって Snort コアが作成されます。

アップグレード

GCP の Threat Defense Virtual のアップグレードは、Cisco Secure Firewall バージョン 7.1 から 7.2 へはサポートされていません。Cisco Secure Firewall バージョン 7.1 から 7.2 にアップグレードする場合は、再イメージ化を実行します。

サポートされない機能

- IPv6
- Threat Defense Virtual ネイティブ HA
- トランスペアレント/インライン/パッシブ モード
- ジャンボ フレーム

データインターフェイス への NIC マッピング

Cisco Secure Firewall バージョン 7.1 以前のリリースにおけるネットワーク インターフェイス カード（NIC）とデータインターフェイスのマッピングは次のとおりです。

- nic0 – 管理インターフェイス
- nic1 – 診断インターフェイス
- nic2 – ギガビットイーサネット 0/0
- nic3 – ギガビットイーサネット 0/1

Cisco Secure Firewall バージョン 7.2 以降、外部ロードバランサ（ELB）はパケットを nic0 にのみ転送するため、North-South トラフィックの移動を容易にするために nic0 にデータインターフェイスが必要です。

Cisco Secure Firewall バージョン 7.2 の NIC とデータインターフェイスのマッピングは次のとおりです。

- nic0 – ギガビットイーサネット 0/0
- nic1 – ギガビットイーサネット 0/1
- nic2 – 管理インターフェイス
- nic3 – 診断インターフェイス
- nic4 – ギガビットイーサネット 0/2

  .

  .

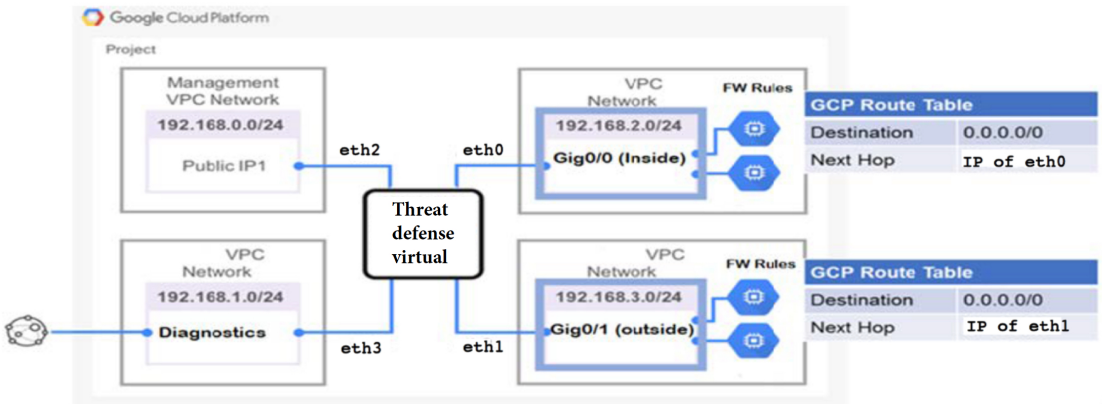
  .

- nic(N-2) – ギガビットイーサネット 0/N-4
- nic(N-1) – ギガビットイーサネット 0/N-3

ネットワークトポロジの例

次の図は、Threat Defense Virtual 用に 4 つのサブネット（管理、診断、内部、外部）が GCP 内に設定されたルーテッドファイアウォールモードの Threat Defense Virtual の推奨トポロジを示しています。

図 *1 : GCP* 展開での *Threat Defense Virtual* の例



Secure Firewall Threat Defense Virtual デバイスの管理方法

Secure Firewall Threat Defense Virtual を管理するには、2 つのオプションがあります。

Secure Firewall Management Center

多数のデバイスを管理している場合、または Threat Defense で許可される、より複雑な機能や設定を使用したい場合は、組み込みの Device Manager の代わりに Management Center を使用してデバイスを設定します。

重要 Device Manager と Management Center の両方を使用して Threat Defense デバイスを管理することはできません。いったん Device Manager の統合管理を有効にすると、ローカル管理を無効にして、Management Center を使用するように管理を再設定しない限り、Management Center を使用して Threat Defense デバイスを管理することはできなくなります。一方、Threat Defense デバイスを Management Center に登録すると、Device Manager のオンボード管理サービスは無効になります。

注意 現在、シスコには Device Manager の設定を Management Center に移行するオプションはありません。その逆も同様です。Threat Defense デバイス用に設定する管理のタイプを選択する際は、このことを考慮してください。

Secure Firewall Device Manager

Device Manager はオンボード統合マネージャです。

Device Manager は一部の Threat Defense デバイスに搭載された Web ベースの設定インターフェイスです。Device Manager では、小規模ネットワークで最も一般的に使用されるソフトウェアの基本機能を設定できます。また、これは多くの Threat Defense デバイスを含む大規模なネットワークを制御するために強力な複数デバイスのマネージャを使用することがない、単一のデバイスまたは限られた数のデバイスを含むネットワークのために特に設計されています。

（注） Device Manager をサポートする Threat Defense デバイスのリストについては、「Cisco Secure Firewall Device Manager Configuration Guide」を参照してください。

GCP 環境の設定

Threat Defense Virtual の展開には、Threat Defense Virtualを展開する前に 4 つのネットワークを作成する必要があります。ネットワークは次のとおりです。

- 管理サブネットの管理 VPC。
- 診断 VPC または診断サブネット。
- 内部サブネットの内部 VPC。
- 外部サブネットの外部 VPC。

さらに、Threat Defense Virtual を通過するトラフィックフローを許可するようにルートテーブルと GCP ファイアウォールルールを設定します。ルートテーブルとファイアウォールルールは、Threat Defense Virtual 自体に設定されているものとは別になっています。関連するネットワークと機能に応じて、GCP ルートテーブルとファイアウォールルールに名前を付けます。ガイドとして、ネットワークトポロジの例 を参照してください。

手順

ステップ 1 GCP コンソールで、**[VPC ネットワーク（VPC networks）]** を選択し、**[VPC ネットワークの作成（Create VPC Network）]** をクリックします。

ステップ 2 **[名前（Name）]** フィールドに、特定の名前を入力します。

ステップ 3 サブネット作成モードで、[カスタム（Custom）] をクリックします。

ステップ 4 **新しいサブネット**で **[名前（Name）]** フィールドに、特定の名前を入力します。

ステップ 5 **[地域（Region）]** ドロップダウンリストから、展開に適した地域を選択します。4 つのネットワークはすべて同じリージョン内にある必要があります。

ステップ 6 [IPアドレス範囲（IP address range）] フィールドで、最初のネットワークのサブネットを CIDR 形式（10.10.0.0/24 など）で入力します。

ステップ 7 その他すべての設定はデフォルトのままで、 **[作成（Create）]** をクリックします。

ステップ 8 ステップ 1 ー 7 を繰り返して、残りの 3 つの VPC ネットワークを作成します。

ファイアウォールルールの作成

Threat Defense Virtual インスタンスの展開中に、管理インターフェイスのファイアウォールルールを適用します（Management Centerとの SSH および SFTunnel 通信を許可するため）。Threat Defense Virtual の導入 （11 ページ） を参照してください。要件に応じて、内部、外部、および診断インターフェイスのファイアウォールルールを作成することもできます。

手順

ステップ 1 GCP コンソールで、**[ネットワーキング（Networking）]** > **[VPC ネットワーク（VPC network）]** > **[ファイアウォール（Firewall）]**を選択し、**[ファイアウォールルールの作成（Create Firewall Rule）]** をクリックします。

ステップ 2 **[名前（Name）]** フィールドに、ファイアウォールルールのわかりやすい名前を入力します（例：*vpc-asiasouth-inside-fwrule*）。

ステップ 3 **[ネットワーク（Network）]** ドロップダウンリストから、ファイアウォールルールを作成する VPC ネットワークの名前を選択します（例：*ftdv-south-inside*）。

ステップ 4 **[ターゲット（Targets）]** ドロップダウンリストから、ファイアウォールルールに適用可能なオプションを選択します（例：**[ネットワーク内のすべてのインスタンス（All instances in the network）]**）。

ステップ 5 **[送信元 IP 範囲（Source IP Ranges）]** フィールドに、送信元 IP アドレスの範囲を CIDR 形式で入力します（例：0.0.0.0/0）。

トラフィックは、これらの IP アドレス範囲内の送信元からのみ許可されます。

ステップ 6 **[プロトコルとポート（Protocols and ports）]** の下で、**[指定されたプロトコルとポート（Specified protocols and ports）]** を選択します。

ステップ 7 セキュリティルールを追加します。

ステップ 8 [作成（Create）] をクリックします。

Threat Defense Virtual の導入

以下の手順に従って、GCP マーケットプレイスから提供される Cisco Firepower NGFW 仮想ファイアウォール（NGFWv）を使用して Threat Defense Virtual インスタンスを展開できます。

手順

ステップ 1 GCP コンソールにログインします。

ステップ 2 **ナビゲーションメニュー** > **の [マーケットプレイス（Marketplace）]**をクリックします。

ステップ 3 マーケットプレイスで「Cisco Firepower NGFW 仮想ファイアウォール（NGFWv）（Cisco Firepower NGFW virtual firewall (NGFWv)）」を検索して、製品を選択します。

ステップ 4 [作成（Launch）] をクリックします。

a) **[展開名（Deployment name）]**：インスタンスの一意の名前を指定します。

b) **[ゾーン（Zone）]**：Threat Defense Virtualを展開するゾーンを選択します。

c) **[マシンタイプ（Machine type）]**：システム要件 （2 ページ）に基づいて正しいマシンタイプを選択します。

d) **[SSH キー（SSH key）]（オプション）**：SSH キーペアから公開キーを貼り付けます。

キーペアは、GCP が保存する公開キーと、ユーザーが保存する秘密キーファイルで構成されます。これらを一緒に使用すると、インスタンスに安全に接続できます。キーペアはインスタンスへの接続に必要となるため、必ず既知の場所に保存してください。

e) このインスタンスにアクセスするためのプロジェクト全体の SSH キーを許可するかブロックするかを選択します。Google ドキュメント『Allowing or blocking project-wide public SSH keys from a Linux instance』を参照してください。

f) **[起動スクリプト（Startup script）]**：インスタンスが起動するたびに自動化されたタスクを実行するために、Threat Defense Virtual インスタンスの起動スクリプトを作成できます。

次に、**[起動スクリプト（Startup script）]** フィールドにコピーして貼り付ける day0 構成の例を示します。

```
{
"AdminPassword": "Cisco@123123",
"Hostname": "ftdv-gcp",
"DNS1": "8.8.8.8",
"FirewallMode": "routed",
"IPv4Mode": "dhcp",
"ManageLocally": "No"
}
```

ヒント
実行エラーを防ぐには、JSON 検証ツールを使用して Day0 構成を検証する必要があります。

g) **[ネットワークインターフェイス（Network interfaces）]**：1）管理、2）診断、3）内部、4）外部のインターフェイスを設定します。

（注）
インスタンスを作成した後では、インスタンスにインターフェイスを追加できません。不適切なインターフェイス構成でインスタンスを作成した場合は、インスタンスを削除し、適切なインターフェイス構成で再作成する必要があります。

1. **[ネットワーク（Network）]** ドロップダウンリストから、[VPC network（VPC ネットワーク）]（*vpc-asiasouth-mgmt* など）を選択します。
2. **[外部 IP（External IP）]** ドロップダウンリストから、適切なオプションを選択します。

   管理インターフェイスには、[外部 IP からエフェメラルへ（External IP to Ephemeral）] を選択します。内部および外部インターフェイスでは、これはオプションです。
3. [完了（Done）] をクリックします。

h) **[ファイアウォール（Firewall）]**：ファイアウォールルールを適用します。

- **[インターネットからの TCP ポート 22 のトラフィックを許可する（SSH アクセス）（Allow TCP port 22 traffic from the Internet (SSH access)）]** チェックボックスをオンにして、SSH を許可します。
- **[インターネットからの HTTPS のトラフィックを許可する（FMC access）（Allow HTTPS traffic from the Internet (FMC access)）]** チェックボックスをオンにして、Management Centerおよび管理対象デバイスが双方向の SSL 暗号化通信チャネル（SFTunnel）を使用して通信できるようにします。

i) **[詳細（More）]** をクリックしてビューを展開し、**[IP 転送（IP Forwarding）]**が**[オン（On）]**に設定されていることを確認します。

ステップ 5 [展開（Deploy）] をクリックします。

（注）
起動時間は、リソースの可用性など、さまざまな要因によって異なります。初期化が完了するまでに 7 ～ 8 分かかることがあります。初期化は中断しないでください。中断すると、アプライアンスを削除して、最初からやり直さなければならないことがあります。

次のタスク

GCP コンソールの [VM インスタンス（VM instance）] ページからインスタンスの詳細を表示します。インスタンスを停止および開始するための内部 IP アドレス、外部 IP アドレス、およびコントロールが表示されます。編集する場合は、インスタンスを停止する必要があります。

外部 IP を使用した Threat Defense Virtual インスタンスへの接続

Threat Defense Virtualインスタンスには、内部 IP と外部 IP が割り当てられます。外部 IP を使用してThreat Defense Virtualインスタンスにアクセスできます。

手順

ステップ 1 GCP コンソールで、**[コンピューティングエンジン（Compute Engine）]**>**[VM インスタンス（VM instances）]** を選択します。

ステップ 2 Threat Defense Virtualのインスタンス名をクリックすると、**[VM インスタンスの詳細（VM instance details）]** ページが開きます。

ステップ 3 **[詳細（Details）]** タブで、**[SSH]** フィールドのドロップダウンメニューをクリックします。

ステップ 4 **[SSH]** ドロップダウンメニューから、目的のオプションを選択します。

次の方法を使用してThreat Defense Virtualインスタンスに接続できます。

- その他の SSH クライアントまたはサードパーティ製ツール：詳細については、Google ドキュメントの「Connecting using third-party tools」を参照してください。

SSH を使用した Threat Defense Virtual インスタンスへの接続

UNIX スタイルのシステムから Threat Defense Virtual インスタンスに接続するには、SSH を使用してインスタンスにログインします。

手順

ステップ 1 次のコマンドを使用して、ファイルの権限を設定し、自分だけがファイルを読み取れるようにします。

```
$ chmod 400 <private_key>
```

ここで、

`<private_key>` は、アクセスするインスタンスに関連付けられた秘密キーを含むファイルのフルパスと名前です。

ステップ 2 インスタンスにアクセスするには、次の SSH コマンドを使用します。

```
$ ssh -i <private_key> <username>@<public-ip-address>
```

`<private_key>` は、アクセスするインスタンスに関連付けられた秘密キーを含むファイルのフルパスと名前です。

`<username>` は、Threat Defense Virtual インスタンスのユーザー名です。

`<public-ip-address>` は、コンソールから取得したインスタンスの IP アドレスです。

シリアルコンソールを使用した Threat Defense Virtual インスタンスへの接続

手順

ステップ 1 GCP コンソールで、**[コンピューティングエンジン（Compute Engine）]** > **[VM インスタンス（VM instances）]** を選択します。

ステップ 2 Threat Defense Virtualのインスタンス名をクリックすると、**[VM インスタンスの詳細（VM instance details）]** ページが開きます。

ステップ 3 **[詳細（Details）]** タブで、**[シリアルコンソールへの接続（Connect to serial console）]** をクリックします。

詳細については、Google ドキュメントの「シリアルコンソールとのやり取り」を参照してください。

Gcloud を使用した Threat Defense Virtual インスタンスへの接続

手順

ステップ 1 GCP コンソールで、**[コンピューティングエンジン（Compute Engine）]** > **[VM インスタンス（VM instances）]** を選択します。

ステップ 2 Threat Defense Virtualのインスタンス名をクリックすると、**[VM インスタンスの詳細（VM instance details）]** ページが開きます。

ステップ 3 **[詳細（Details）]** タブで、**[SSH]** フィールドのドロップダウンメニューをクリックします。

ステップ 4 **[gcloud コマンドを表示（View gcloud command）]** > **[Cloud Shell で実行（Run in Cloud Shell）]**をクリックします。

[Cloud Shell] ターミナルウィンドウが開きます。詳細については、Google ドキュメントの「gcloud コマンドラインツールの概要」、および「gcloud compute ssh」を参照してください。

Auto Scale ソリューション

次の項では、Auto Scale ソリューションのコンポーネントが GCP の Threat Defense Virtual でどのように機能するかについて説明します。

概要

Threat Defense Virtual Auto Scale for GCP は、GCP によって提供されるサーバーレス インフラストラクチャ（クラウド機能、ロードバランサ、Pub/Sub、インスタンスグループなど）を利用した完全なサーバーレス導入です。

Threat Defense Virtual Auto Scale for GCP 導入の主な特徴は次のとおりです。

- GCP Deployment Manager のテンプレートをベースとした導入。
- CPU 使用率などに基づくスケーリングメトリックをサポート。
- Threat Defense Virtual 展開とマルチ可用性ゾーンのサポート。
- Threat Defense Virtual の自動登録および登録解除をサポート。
- 完全に自動化された設定をスケールアウトされた Threat Defense Virtual インスタンスに自動適用。

- NAT ポリシー、アクセスポリシー、およびルートを自動的に Threat Defense Virtual に適用。
- ロードバランサとマルチ可用性ゾーンのサポート。
- 他のプラットフォームで Management Center Virtual をサポート。
- シスコでは、導入を容易にするために、Auto Scale for GCP の導入パッケージを提供しています。

注意事項と制約事項

- IPv4 だけがサポートされます。
- ライセンス：BYOL のみをサポートしています。PAYG ライセンスはサポートされていません。
- デバイス機能エラーはログに表示されません。
- サポートされるデバイスの最大数は 25 です。これは、Management Center Virtual インスタンスの上限です。
- すべての Cisco Secure Firewall バージョンで、提供されているテンプレートを使用して、Threat Defense Virtual Auto Scale ソリューションを展開できます。Threat Defense Virtual インスタンスは、少なくとも 4 つのインターフェイス（1 つの管理インターフェイス、1 つの診断インターフェイス、および 2 つのデータインターフェイス）で展開されます。
- スケールアウト時間を短縮するコールドスタンバイまたはスナップショットメソッドはサポートされていません。
- スケジュールベースのスケーリングはサポートされていません。
- 平均メモリ使用率に基づく自動スケーリングはサポートされていません。
- スケールイン/スケールアウトにより、インスタンスの数が 1 よりも多く減少/増加する場合がありますが、Management Center Virtual での Threat Defense Virtual インスタンスの登録解除/登録は順次 1 つずつ実行されます。
- スケールイン時に 300 秒の接続ドレイン時間があります。ドレイン時間を必要な時間に手動で設定することもできます。
- 外部ロードバランサは、提供されているテンプレートによって作成されます。ロードバランサのパブリック IP の DNS 要件をカスタマイズすることはできません。
- ユーザーは、既存のインフラストラクチャを導入のサンドイッチモデルに適合させる必要があります。
- スケールアウトおよびスケールインのプロセス中に発生したエラーの詳細については、クライド機能のログを分析してください。

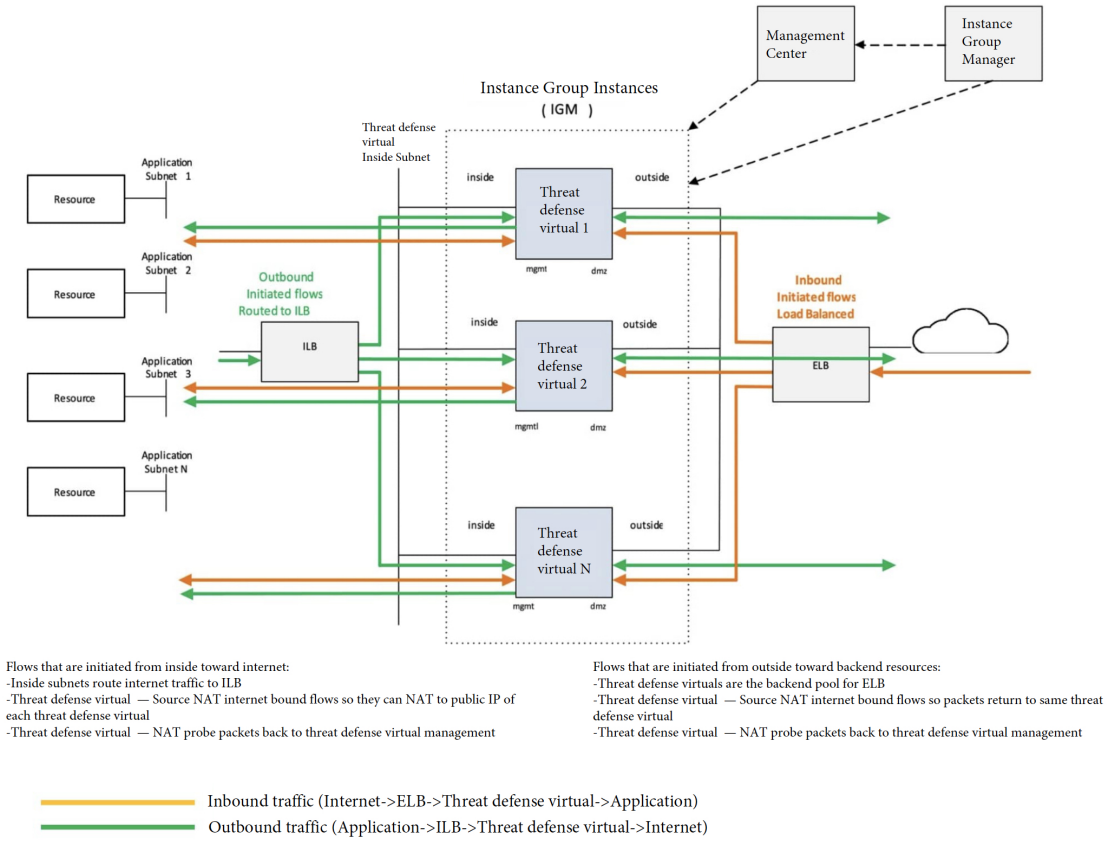
- NAT、デバイスグループに付加されたセキュリティポリシー、および静的ルートは、新しく作成された 脅威に対する防御 ファイルに適用されます。
- ソリューションを複数の Threat Defense Virtual に対して展開する場合、Management Center Virtual は一度に 1 つの登録要求しか処理できないため、展開時間が長くなります。スケールアウトによって複数の Threat Defense Virtual インスタンスが追加されると、展開時間も長くなります。現在、すべての登録と登録解除は連続して実行されます。
- 自動スケーリングを開始する前に、Management Center Virtual でデバイスグループ、NAT ルール、およびネットワークオブジェクトを作成する必要があります。ILB および ELB IP は、ソリューションを展開してからのみ使用できることに注意してください。したがって、ダミーオブジェクトを作成し、実際の IP を取得した後にオブジェクトを更新できます。

Auto Scale の導入例

Threat Defense Virtual Auto Scale for GCP は、Threat Defense Virtual インスタンスグループを GCP の内部ロードバランサ（ILB）と GCP の外部ロードバランサ（ELB）の間に配置する水平方向の自動スケーリングソリューションです。

- ELB は、インターネットからのトラフィックをインスタンスグループ内の Threat Defense Virtual インスタンスに分散させます。その後、Threat Defense Virtual がアプリケーションにトラフィックを転送します。
- ILB は、アプリケーションからのインターネットトラフィックをインスタンスグループ内の Threat Defense Virtual インスタンスに分散させます。その後、Threat Defense Virtual がインターネットにトラフィックを転送します。
- ネットワークパケットが、単一の接続で両方（内部および外部）のロードバランサを通過することはありません。
- スケールセット内の Threat Defense Virtual インスタンスの数は、負荷条件に基づいて自動的にスケーリングおよび設定されます。

図 *2 : Threat Defense Virtual* 自動スケールのユースケース



スコープ

このドキュメントでは、Threat Defense Virtual Auto Scale for GCP ソリューションのサーバーレスコンポーネントを展開する際の詳細な手順について説明します。

重要

- 導入を開始する前に、ドキュメント全体をお読みください。
- 導入を開始する前に、前提条件を満たしていることを確認します。
- ここに記載されている手順と実行順序に従っていることを確認します。

導入パッケージのダウンロード

Threat Defense Virtual Auto Scale for GCP は、GCP によって提供されるサーバーレス インフラストラクチャ（クラウド機能、ロードバランサ、Pub/Sub、インスタンスグループなど）を利用した GCP Deployment Manager のテンプレートベースの導入です。

Threat Defense Virtual Auto Scale for GCP 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 Threat Defense Virtual 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。

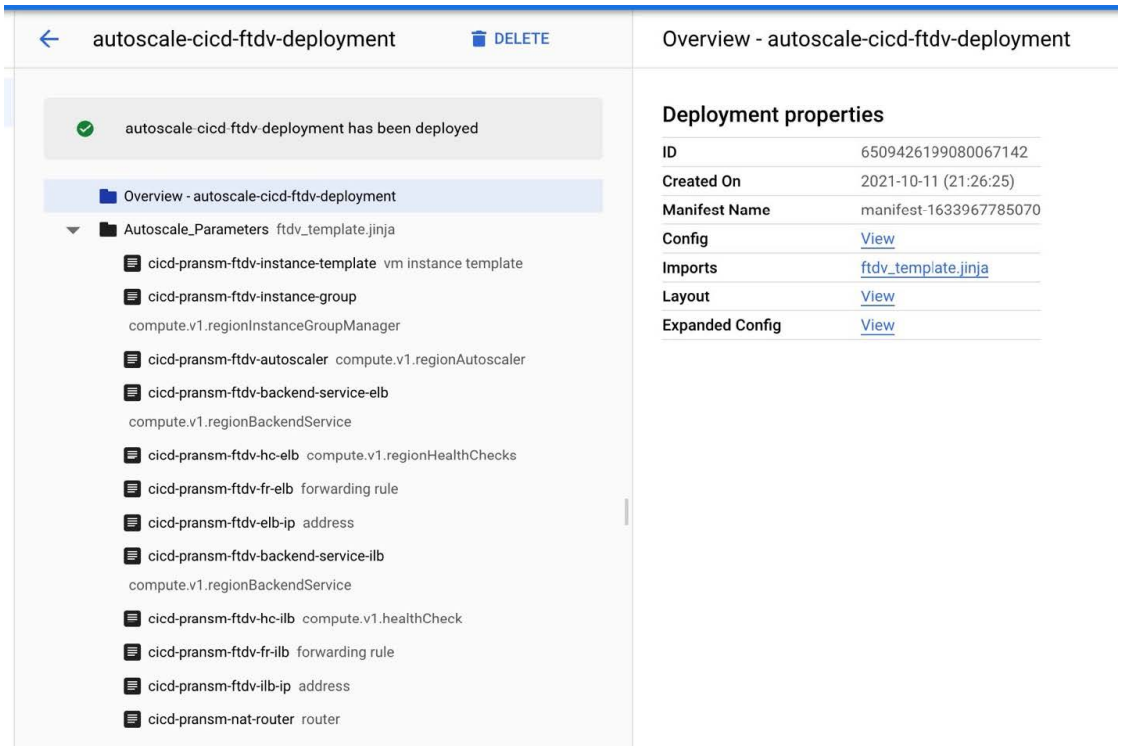
システム要件

Threat Defense Virtual Auto Scale for GCP 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。

導入マネージャ

- 構成をコードとして扱い、反復可能な展開を実行します。Google Cloud Deployment Manager では、YAML を使用して、アプリケーションに必要なすべてのリソースを宣言形式で指定できます。また、Jinja2 テンプレートを使用して構成をパラメータ化し、一般的な導入パラダイムを再利用可能にすることもできます。
- リソースを定義する構成ファイルを作成します。リソースを作成するプロセスを繰り返し実行することで、一貫した結果を得ることができます。詳細については、https://cloud.google.com/deployment-manager/docsを参照してください。

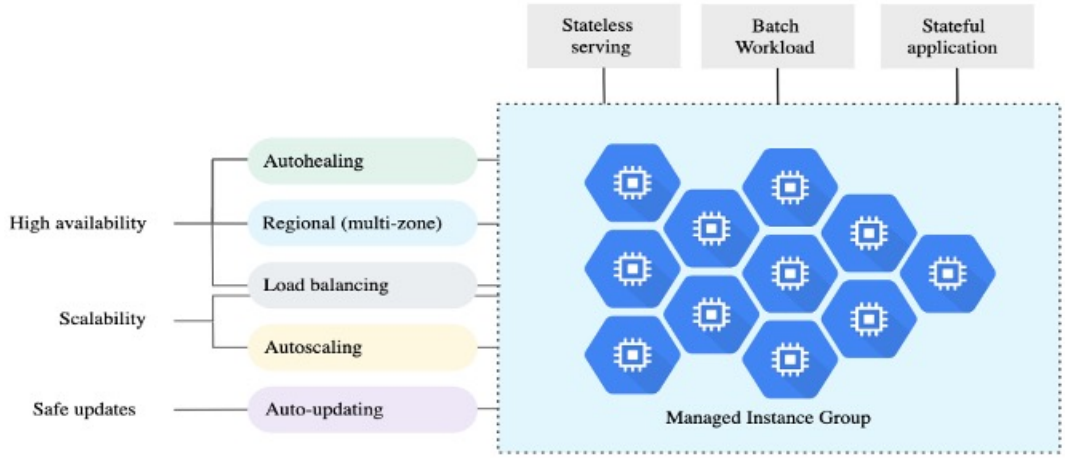
図 *3*: 導入マネージャビュー



GCP のマネージド インスタンス グループ

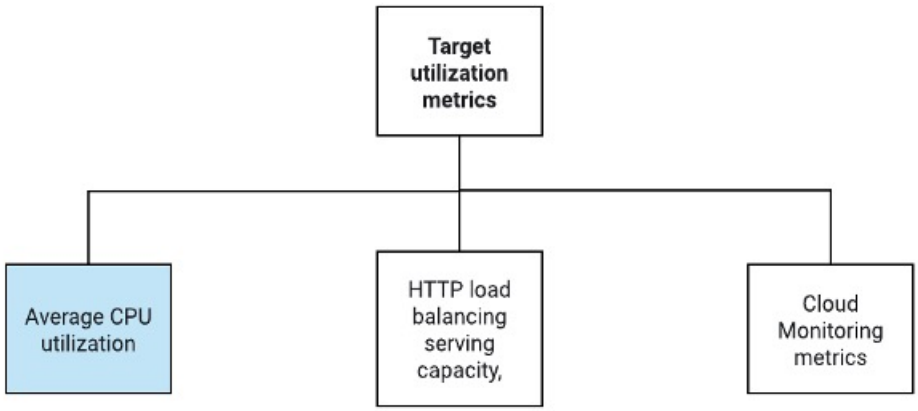
マネージドインスタンス グループ（MIG）は、指定したインスタンステンプレートとオプションのステートフル構成に基づいて、各マネージドインスタンスを作成します。詳細については、https://cloud.google.com/compute/docs/instance-groupsを参照してください。

図 *4*: インスタンスグループの機能



ターゲット使用率メトリック

- 次の図は、ターゲット使用率のメトリックを示しています。自動スケーリングを決定する際、平均 CPU 使用率メトリックのみが使用されます。
- オートスケーラは、選択された使用率メトリクスに基づいて使用状況の情報を継続的に収集し、実際の使用率を希望するターゲット使用率と比較します。次に、この情報を使用して、グループがインスタンスを削除する必要があるか（スケールイン）またはインスタンスを追加する必要があるか（スケールアウト）を判断します。
- ターゲット使用率レベルとは、仮想マシン（VM）インスタンスをどのレベルで維持するかを示します。たとえば、CPU使用率に基づいてスケーリングする場合、ターゲット使用率レベルを75%に設定すると、オートスケーラは指定されたインスタンス グループで75%またはそれに近いCPU使用率を維持します。各メトリックの使用率レベルは、自動スケーリング ポリシーに基づいてさまざまに解釈されます。詳細については、https://cloud.google.com/compute/docs/autoscalerを参照してください。



サーバーレスクラウド機能

SSH パスワードの変更、マネージャの設定、Management Center Virtual への Threat Defense Virtual の登録、Management Center Virtual から Threat Defense Virtual の登録解除などのタスクには、サーバーレスの Google Cloud 機能を使用します。

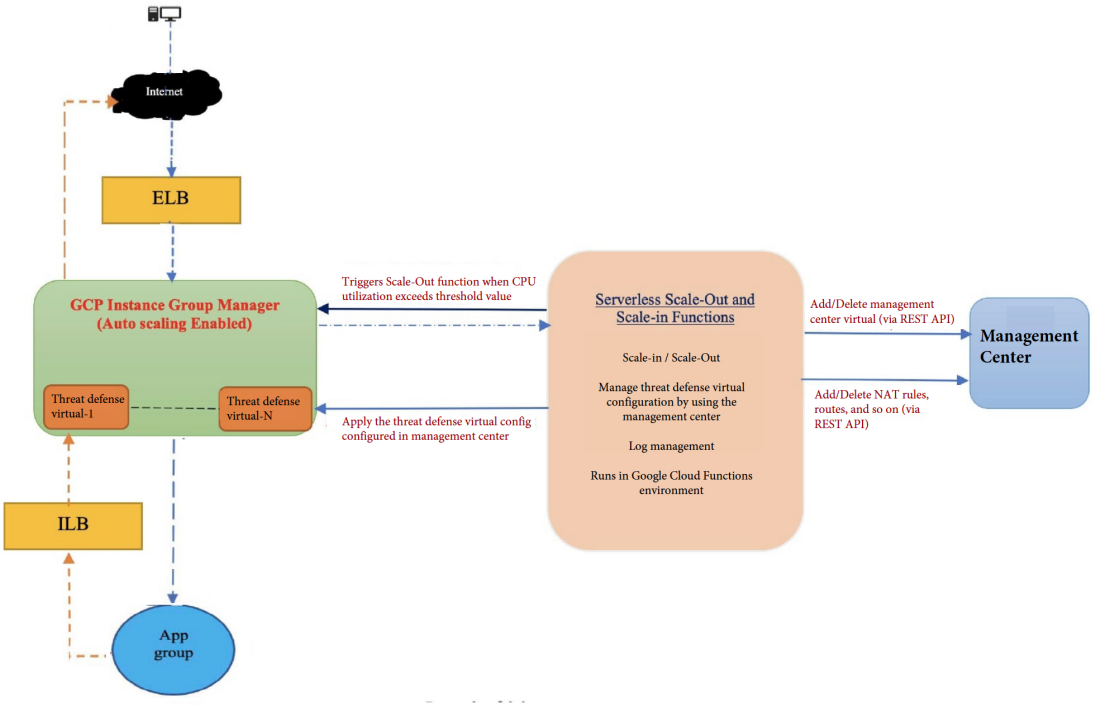
- スケールアウト中に新しい Threat Defense Virtual インスタンスがインスタンスグループに追加された場合、SSH パスワードの変更、マネージャの設定、Management Center Virtual への Threat Defense Virtual の登録、Management Center Virtual から Threat Defense Virtual の登録解除などのタスクを実行する必要があります。
- クラウド機能は、スケールアウトプロセス中にクラウドのパブリック/サブトピックを介してトリガーされます。また、スケールアウト時のインスタンス追加専用のフィルタを備えたログシンクもあります。

クラウド機能を使用したサーバーレスのライセンス登録解除

- スケールイン時のインスタンス削除中に、Threat Defense Virtual インスタンスからライセンスの登録を解除し、Management Center Virtual から Threat Defense Virtual の登録を解除する必要があります。
- クラウド機能は、クラウドのパブリック/サブトピックを介してトリガーされます。特に削除プロセスについては、スケールイン時のインスタンス削除専用のフィルタを備えたログシンクがあります。
- クラウド機能がトリガーされると、削除対象の Threat Defense Virtual インスタンスに SSH で接続し、ライセンス登録解除のコマンドを実行します。

Auto Scale ソリューションの概要

図 *5 : Auto Scale* ソリューションの概要



前提条件

GCP リソース

GCP プロジェクト

このソリューションのすべてのコンポーネントを展開するには、既存または新しく作成されたプロジェクトが必要です。

VPC ネットワーク

4 つの VPC が使用可能/作成されていることを確認します。Auto Scale 展開では、ネットワークリソースの作成、変更、管理は行われません。

既存のサブネットワークに加えて、/28 サブネットワークを使用して管理 VPC ネットワークに新しい VPC コネクタを作成します。クラウド機能はVPC コネクタを使用して、プライベート IP アドレスで Threat Defense Virtual にアクセスします。

Threat Defense Virtual には 4 つのネットワーク インターフェイスが必要なため、仮想ネットワークには次の 4 つのサブネットが必要です。

- 外部トラフィック
- 内部トラフィック
- 管理トラフィック
- 診断トラフィック

Firewall

VPC 間通信を許可し、正常性プローブも許可するファイアウォールルールを作成する必要があります。

内部、外部、管理、および診断インターフェイス用に 4 つのファイアウォールルールが必要です。また、正常性チェックプローブを許可するファイアウォールルールを作成します。

正常性チェックプローブの IP アドレスは次のとおりです。

- 35.191.0.0/16
- 130.211.0.0/22
- 209.85.152.0/22
- 209.85.204.0/22

Deployment Manager テンプレートで後に使用されるファイアウォールタグに注意する必要があります。

サブネットが接続されているネットワーク セキュリティ グループで、次のポートを開く必要があります。

- SSH（TCP/22）：ロードバランサと Threat Defense Virtual 間の正常性プローブに必要です。サーバーレス機能と Threat Defense Virtual 間の通信に必要です。
- アプリケーション固有のプロトコルまたはポート：ユーザーアプリケーションに必要です（TCP/80 など）。

GCP クラウド機能パッケージの構築

Threat Defense Virtual GCP Auto Scale ソリューションでは、圧縮形式の ZIP パッケージでクラウド関数を提供する 2 つのアーカイブファイルを作成する必要があります。

- `ftdv_scalein.zip`
- `ftdv_scaleout.zip`

`ftdv_scalein.zip` および `ftdv_scaleout.zip` パッケージの構築方法については、Auto Scale の導入手順を参照してください。

関数は、特定のタスクを実行するために可能な限り独立しており、拡張機能や新しいリリースのサポートのために必要に応じてアップグレードできます。

入力パラメータ

次の表に、テンプレートパラメータおよび例を示します。各パラメータの値を決めたら、GCP プロジェクトに GCP Deployment Manager を展開するときに、各パラメータを使用して Threat Defense Virtual デバイスを作成できます。

表 *4:* テンプレートパラメータ

パラメータ名	使用できる値/タイプ	説明
resourceNamePrefix	文字列	すべてのリソースは、このプレフィックスを含む名前で作成されます。 例：demo-test
region	GCP でサポートされている有効なリージョン [String]	プロジェクトが展開されるリージョン名。 例：us-central1
serviceAccountMailId	文字列 [Email Id]	サービスアカウントを識別するメールアドレス。

パラメータ名	使用できる値/タイプ	説明
vpcConnectorName	文字列	サーバーレス環境と VPC ネットワーク間のトラフィックを処理するコネクタの名前。 例：demo-test-vpc-connector
adminPassword	文字列	Threat Defense Virtual インスタンスの初期パスワード。後でこのパラメータは「newFtdPasswordSecret」に変更されます。
bucketName	文字列	クラウド機能の ZIP パッケージをアップロードする GCP ストレージバケットの名前。 例：demo-test-bkt
coolDownPeriodSec	整数	オートスケーラが新しいインスタンスから情報の収集を開始するまで待機する秒数。 例：30
cpuUtilizationTarget	10 進数 (0,1]	オートスケーラーが維持する必要があるインスタンスグループ内の VM の平均 CPU 使用率。 例：0.5
deployUsingExternalIP	ブール値	Threat Defense Virtual の管理にパブリック IP アドレスが必要かどうかを決定します。 例：true true に設定されている場合、Threat Defense Virtual にはパブリック IP アドレスが必要です。false に設定されている場合、パブリック IP アドレスは必要ありません。
diagFirewallRule	文字列	診断 VPC 用に作成されたファイアウォールルールの名前。 例：cisco-ftdv-diag-firewall-rule

パラメータ名	使用できる値/タイプ	説明
diagSubnetworkName	文字列	診断インターフェイスに使用される VPC サブネットの名前。 例：cisco-ftdv-diag-subnet
diagVpcName	文字列	診断インターフェイスに使用される VPC の名前。 例：custom-ftdv-diag-vpc
elbFePorts	整数	ELB ファストイーサネットポート。 例：80,22
elbIpProtocol	文字列	使用される ELB IP プロトコル。 例：TCP
elbPort	整数	ELB ポート番号。 例： 80
elbPortName	文字列	ELB ポートの名前。 例：tcp
elbPortRange	整数	ELB ポートの範囲。 例：80-80
elbProtocol	文字列	使用される ELB プロトコル。 例：TCP
elbProtocolName	文字列	ELB プロトコルの名前。 例：TCP
elbTimeoutSec	整数	秒単位の ELB タイムアウト時間。 例：5
elbUnhealthyThreshold	整数	ヘルスチェック不合格回数のしきい値。 例：2。

パラメータ名	使用できる値/タイプ	説明
fmcIP	文字列	Management Center の IP アドレス 例：10.61.1.2
fmcPasswordSecret と新しい FtdPasswordSecret	文字列	作成されたシークレットの名前。
fmcUsername	文字列	Management Center Virtual のユーザー名
ftdvCheckIntervalSec	整数	ヘルスチェックの間隔。 例：300
ftdvHealthCheckPort	整数	Threat Defense Virtual のヘルスチェックのポート番号。 例：22
ftdvHealthCheckProtocolName	文字列	ヘルスチェックに使用されるプロトコル。 例：TCP
ftdvPassword	文字列	Threat Defense Virtual のパスワード。
ftdvTimeoutSec	整数	Threat Defense Virtual 接続のタイムアウト 例：300
ftdvUnhealthyThreshold	整数	ヘルスチェック不合格回数のしきい値。 例：3
grpID	文字列	Management Center で作成されたデバイスグループの名前。 例：auto-group
healthCheckFirewallRule	文字列	ヘルスチェックプローブの IP 範囲からのパケットを許可するファイアウォールルールの名前。 例：custom-ftdv-hc-firewall-rule

パラメータ名	使用できる値/タイプ	説明
healthCheckFirewallRuleName	文字列	ヘルスチェックプローブの IP 範囲からのパケットを許可するファイアウォールルールのタグ。 例：demo-test-health-allow-all
ilbCheckIntervalSec	整数	ILB接続をチェックする間隔。 例：10
ilbDrainingTimeoutSec	整数	接続ドレインのタイムアウト時間 例：60
ilbPort	整数	ILB ポート番号。 例： 80
ilbProtocol	文字列	使用される ILB プロトコル。 例：TCP
ilbProtocolName	文字列	ILB プロトコル名。 例：TCP
ilbTimeoutSec	整数	ILB タイムアウト時間。 例：5
ilbUnhealthyThreshold	整数	ヘルスチェック不合格回数のしきい値。 例：3
insideFirewallRule	文字列	内部ファイアウォールルールの名前。 例：custom-ftdv-in-firewall-rule
insideFirewallRuleName	文字列	内部 VPC での通信を許可するファイアウォールルールのタグ。 例：demo-test-inside-allowall
insideGwName	文字列	内部ゲートウェイの名前。 例：inside-gateway

パラメータ名	使用できる値/タイプ	説明
insideSecZone	文字列	内部セキュリティゾーンの名前。 例：inside-zone
insideSubnetworkName	文字列	内部サブネットの名前。 例：custom-ftdv-inside-subnet
insideVPCName	文字列	内部 VPC の名前。 例：demo-test-inside
insideVPCSubnet	文字列	内部サブネットの名前。 例：demo-test-inside-subnet
licenseCAPS	文字列	使用するライセンスの名前 例：BASE、MALWARE、URL Filter、THREAT
マシン タイプ	文字列	Threat Defense Virtual VM のマシンタイプ。 例：n1-standard-4
maxFTDCount	整数	インスタンスグループで許可される Threat Defense Virtual インスタンスの最大数。 例：3
maxFTDReplicas	整数	自動スケーリンググループ内の Threat Defense Virtual インスタンスの最大数。 例：2。
mgmtFirewallRule	文字列	管理ファイアウォールルールの名前。 例：cisco-ftdv-mgmt-firewall-rule
mgmtFirewallRuleName	文字列	管理 VPC での通信を許可するファイアウォールルールのタグ。 例：demo-test-mgmt-allowall

パラメータ名	使用できる値/タイプ	説明
mgmtSubnetworkName	文字列	管理サブネットの名前。 例：custom-ftdv-mgmt-subnet
mgmtVPCName	文字列	管理 VPC の名前。 例：demo-test-mgmt
mgmtVPCSubnet	文字列	管理サブネットの名前。 例：demo-test-mgmt-subnt
minFTDCount	整数	任意の時点でインスタンスグループで使用可能な Threat Defense Virtual の最小インスタンス数。 例 1
minFTDReplicas	整数	自動スケーリンググループ内の Threat Defense Virtual インスタンスの最小数。 例：2。
natID	文字列	脅威に対する防御 で Management Center を登録するときに必要な一意の NAT ID。
outsideFirewallRule	文字列	外部ファイアウォールルールの名前。 例：cisco-ftdv-out-firewall-rule
outsideFirewallRuleName	文字列	外部 VPC での通信を許可するファイアウォールルールのタグ。 例：demo-test-outside-allowall
outsideGwName	文字列	外部ゲートウェイの名前。 例：outside-gateway
outsideSecZone	文字列	外部セキュリティゾーンの名前。 例：outside-zone
outsideSubnetworkName	文字列	外部サブネットの名前。 例：custom-ftdv-outside-subnet

パラメータ名	使用できる値/タイプ	説明
outsideVPCName	文字列	外部 VPC の名前。 例：demo-test-outside
outsideVPCSubnet	文字列	外部サブネットの名前。 例：demo-test-outside-subnt
policyID	文字列	ACL ポリシーの名前。
publicKey	文字列	Threat Defense Virtual VM の SSH キー。
sourceImageURL	文字列	プロジェクトで使用する Threat Defense Virtual イメージの URL。
sshUsingExternalIP	ブール値	Google 機能によってパブリック IP アドレスとプライベート IP アドレスのどちらが使用されるかを決定します。 例：true true に設定されている場合、Google 機能はパブリック IP アドレスを使用します。false に設定されている場合、Google 機能はプライベート IP アドレスを使用します。

Auto Scale ソリューションの展開

手順

ステップ 1 Git リポジトリをローカルフォルダに複製します。

```
git clone git_url -b branch_name
```

ステップ 2 gcloud CLI でバケットを作成します。

```
gsutil mb -c nearline gs://bucket_name
```

（注）
システムにインストールされている Google Cloud Shell または Google Cloud SDK で、この手順の任意の **gsutil** または **gcloud** コマンドを実行します。

ステップ 3 Zip 形式の圧縮パッケージを作成します。

a) `ftdv_scaleout` および `ftdv_scalein` フォルダから、以下のファイルで構成される Zip 形式の圧縮パッケージを作成します。

- main.py
- basic_functions.py
- fmc_functions.py
- requirements.txt

（注）
内部 IP アドレスを使用する場合は、main.py ファイルで **ssh_ip = response[‘networkInterfaces’] [2] [‘networkIP’]** コマンドを使用します。外部 IP アドレスを使用する場合は、**ssh_ip = response[‘networkInterfaces’] [2] [‘accessConfigs’] [0] [‘natIP’]** コマンドを入力します。また、この関数では2つの静的ルートが追加されます。静的ルートを変更するには、**fmc.create_static_network_route (vm_name, ‘outside’, ‘any_ipv4’, os.getenv(“OUTSIDE_GW_NAME”), metric=1)** および **fmc.create_static_network_route (vm_name, ‘inside’, ‘any_ipv4’, os.getenv(“INSIDE_GW_NAME”), metric=2)** コマンドを使用します。

b) Zip 形式の圧縮パッケージの名前を `ftdv_scaleout.zip` および `ftdv_scalein.zip` に変更します。

（注）
フォルダ内を移動して選択するファイルを右クリックし、[圧縮|アーカイブ（compress | archive）] を選択すると、GCP が読み取れる .zip が作成されます。

ステップ 4 Zip 形式の圧縮パッケージ（`ftdv_scaleout.zip` および `ftdv_scalein.zip`）をクラウドエディタのワークスペースにアップロードします。

ステップ 5 以下のファイルを Deployment Manager のテンプレートからクラウドエディタのワークスペースにアップロードします。

- ftdv_predeployment.yaml
- ftdv_predeployment.jinja
- ftdv_parameters.yaml
- ftdv_template.jinja

ステップ 6 Zip 形式の圧縮パッケージをバケットストレージにコピーします。

- `gsutil cp ftdv_scaleout.zip gs://bucket_name`
- `gsutil cp ftdv_scalein.zip gs://bucket_name`

ステップ 7 内部、外部、管理、診断インターフェイス用の VPC とサブネットを作成します。

管理 VPC では、/28 サブネット（例：10.8.2.0/28）が必要です。

ステップ 8 内部、外部、管理、診断インターフェイス用に 4 つのファイアウォールルールが必要です。また、正常性チェックプローブを許可するファイアウォールルールが必要です。

ステップ 9 Secret Manager GUI を使用して、次の 2 つのシークレットを作成します。https://console.cloud.google.com/security/secret-manager を参照してください。

- fmc-password
- ftdv-new-password

ステップ 10 VPC コネクタを作成します。

```
gcloud beta compute networks vpc-access connectors create <vpc-connector-name>
--region <region> --subnet=</28 subnet name>
```

例：

```
gcloud beta compute networks vpc-access connectors create demo-vpc-connector
--region us-central1 --subnet=outside-connect-28
Create request issued for: [demo-vpc-connector]
Waiting for operation [projects/asavgcp-poc-4krn/locations/us-central1/operations/
10595de7-837f-4c19-9396-0c22943ecf15] to complete...done.
Created connector [demo-vpc-connector].
```

ステップ 11 パブリック IP を持つ任意のパブリック クラウド プラットフォームに Management Center Virtual を展開します。各種パブリッククラウド プラットフォームに Management Center Virtual を展開する方法の詳細については、『Cisco Secure Firewall Management Center Virtual Getting Started Guide』を参照してください。

（注）
Management Center Virtual を展開したインスタンスでステップ 12 から 16 を実行します。

ステップ 12 Management Center Virtual インスタンス：fmcpassword シークレットに保存されているものと同じパスワードを使用して、Management Center Virtual でユーザー restapi を作成します。詳細については、「ユーザー」を参照してください。

ステップ 13 Management Center Virtual インスタンス：デバイスグループ、アクセス コントロール ポリシー、およびアクセス制御ルールを作成します。詳細については、「デバイスグループの追加」、「基本的なアクセス コントロール ポリシーの作成」、および「アクセスコントロールルールの作成および編集」を参照してください。

ステップ 14 Management Center Virtual インスタンス：以下のオブジェクトを作成します。Management Center Virtual でオブジェクトを作成する方法の詳細については、「オブジェクト管理」を参照してください。

- ELB-IP
- ILB-IP
- Application-IP
- ヘルスチェックの IP 範囲（4）
- メタデータ（Metadata）

```
object network hc1
  subnet 35.191.0.0 255.255.0.0
object network metadata
  host 169.254.169.254
```

```
object network ilb-ip
  host 10.52.1.218
object network hc2
  subnet 130.211.0.0 255.255.252.0
object network elb-ip
  host 34.85.214.40
object network hc3
  subnet 209.85.152.0 255.255.252.0
object network hc4
  subnet 209.85.204.0 255.255.252.0
object network inside-linux
  host 10.52.1.217
object network outside-gateway
  host <>
object network inside-gateway
  host <>
```

ステップ 15 Management Center Virtual インスタンス：セキュリティゾーン（インターフェイスオブジェクト）を作成します。「Creating Security Zone and Interface Group Objects」を参照してください。

- inside-security-zone
- outside-security-zone

ステップ 16 Management Center Virtual インスタンス：NAT ポリシーと NAT ルールを作成します。詳細については、「Network Address Translation」を参照してください。

```
nat (inside,outside) source dynamic hc1 interface destination static ilb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (inside,outside) source dynamic hc2 interface destination static ilb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (inside,outside) source dynamic any interface
nat (outside,inside) source dynamic hc1 interface destination static elb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (outside,inside) source dynamic hc2 interface destination static elb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (outside,inside) source dynamic hc3 interface destination static elb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (outside,inside) source dynamic hc4 interface destination static elb-ip metadata service
SVC_4294968559 SVC_4294968559
nat (outside,inside) source dynamic any interface destination static elb-ip inside-linux
```

	#	Direction	Type	Source Interface Objects	Destination Interface Objects	Original Sources	Original Destinations	Original Services	Translated Sources	Translated Destinations	Translated Services	Options	
	1		D...	inside-zone	outside-zone	hc1	ilb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false	
	2		D...	inside-zone	outside-zone	hc2	ilb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false	
	3		D...	inside-zone	outside-zone	any-ipv4			Interface			Dns:false	
	4		D...	outside-zone	inside-zone	hc1	elb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false	
	5		D...	outside-zone	inside-zone	hc2	elb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false	
	6		D...	outside-zone	inside-zone	hc3	elb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false	
	7		D...	outside-zone	inside-zone	hc4	elb-ip	Original HTTP	Interface	metadata	Original HTTP	Dns:false	
	8		D...	outside-zone	inside-zone	any-ipv4	elb-ip		Interface	inside-linux		Dns:false	

ステップ 17 導入前および Threat Defense Virtual Auto Scale 導入用の Jinja ファイルと YAML ファイルのパラメータを更新します。

a) `ftdv_predeployment.yaml` ファイルを開き、次のパラメータを更新します。

- **resourceNamePrefix**: <resourceNamePrefix>
- **region**: <region>
- **serviceAccountMailId**: <serviceAccountMailId>
- **vpcConnectorName**: <VPC-Connector-Name>
- **bucketName**: <bucketName>
- **fmcIP**：<Management Center-IP-address>
- **regID**：<registration-ID>
- **natID**：<unique-NAT-ID>
- **grpID**：<device-group-name>
- **policyID**：<acl-policy-name>
- **licenseCAPS**：<licenses>
- **fmcPasswordSecret**：<Management Center-password>
- **newFtdPasswordSecret**：<new-Threat Defense Virtual-password>
- **fmcUsername**：<username>
- **ftdvPassword**：<password>
- **outsideGwName**：<outside-gateway-name>
- **insideGwName**：<inside-gateway-name>
- **outsideSecZone**：<outside-security-zone>
- **insideSecZone**：<inside-security-zone>
- **sshUsingExternalIP**：<true/false>

b) `ftdv_predeployment.jinja` ファイルは、`ftdv_predeployment.yaml` ファイルからパラメータを受け取ります。

c) `ftdv_parameters.yaml` ファイルを開き、以下のパラメータを更新します。

VPC and Firewall Parameters

- **mgmtVpcName**：<mgmt-vpc-name>
- **diagVpcName**：<diagnostic-vpc-name>
- **outsideVpcName**：<outside-vpc-name>
- **insideVpcName**：<inside-vpc-name>
- **mgmtSubnetworkName**：<mgmt-subnet-name>
- **diagSubnetworkName**：<diagnostic-subnet-name>

- **outsideSubnetworkName**：<outside-subnet-name>
- **insideSubnetworkName**：<inside-subnet-name>
- **mgmtFirewallRule**：<mgmt-firewall-rule>
- **diagFirewallRule**：<diagnostic-firewall-rule>
- **outsideFirewallRule**：<outside-firewall-rule>
- **insideFirewallRule**：<inside-firewall-rule>
- **healthCheckFirewallRule**：<healthcheck-firewall-rule>
- **adminPassword**：<initial-Threat Defense Virtual-password>
- **deployUsingExternalIP**：<true/false>

Instance Template parameters

- **machineType**：<machine-type>
- **sourceImageURL**：<source-image-URL>

FTDv Health Check

- **ftdvHealthCheckPort**：<port-number>
- **ftdvCheckIntervalSec**：<interval-in-seconds>
- **ftdvTimeoutSec**：<timeout-in-seconds>
- **ftdvHealthCheckProtocolName**：<protocol-name>
- **ftdvUnhealthyThreshold**：<threshold-count>

FTDv Autoscaler

- **cpuUtilizationTarget**：<percentage-in-decimals（例：0.7）>
- **coolDownPeriodSec**：<cooldown-period-in-seconds>
- **minFTDReplicas**：<min-number-of-FTDv-instances>
- **maxFTDReplicas**：<max-number-of-FTDv-instances>

ELB Services

- **elbPort**：<port-number>
- **elbPortName**：<port-name>
- **elbProtocol**：<protocol-name>
- **elbTimeoutSec**：<timeout-in-seconds>
- **elbProtocolName**：<protocol-name>

- **elbUnhealthyThreshold**：<threshold-number-for-failed-health-checks>
- **elbIpProtocol**：<IP-Protocol>
- **elbPortRange**：<port-range>
- **elbFePorts**：<fast-ethernet-ports>

ILB Services

- **ilbProtocol**: <protocol-name>
- **ilbDrainingTimeoutSec**: <timeout-in-seconds>
- **ilbPort**: <port-number>
- **ilbCheckIntervalSec**: <interval-in seconds>
- **ilbTimeoutSec**：<timeout-in-seconds>
- **ilbProtocolName**：<protocol-name>
- **ilbUnhealthyThreshold**：<threshold-number-for-failed-health-checks>

（注）
Threat Defense Virtual Auto Scale の場合、**cpuUtilizationTarget: 0.5** パラメータが設定されており、必要に応じて編集できます。この値は、すべての Threat Defense Virtual インスタンスグループの CPU 使用率が 50% であることを示します。

d) `ftdv_template.jinja` ファイルは、`ftdv_parameters.yaml` ファイルからパラメータを受け取ります。

ステップ 18 導入前の YAML 構成を展開します。

```
gcloud deployment-manager deployments create <pre-deployment-name>
--config ftdv_predeployment.yaml
```

例：

```
gcloud deployment-manager deployments create demo-predeployment
--config ftdv_predeployment.yaml

The fingerprint of the deployment is b'9NOy0gsTPgg16SqUEVsBjA=='
Waiting for create [operation-1624383045917-5c55e266e596d-4979c5b6-66d1025c]...done.
Create operation operation-1624383045917-5c55e266e596d-4979c5b6-66d1025c
completed successfully
```

ステップ 19 Threat Defense Virtual Auto Scale の展開を作成します。

```
gcloud deployment-manager deployments create <deployment-name>
--config ftdv_parameters.yaml
```

例：

```
gcloud deployment-manager deployments create demo-asav-autoscale
--config ftdv_parameters.yaml
The fingerprint of the deployment is b'1JCQi7Il-laWOY7vOLza0g=='
Waiting for create [operation-1624383774235-5c55e51d79d01-1a3acf92-4f3daf16]...done.
Create operation operation-1624383774235-5c55e51d79d01-1a3acf92-4f3daf16
completed successfully.
```

ステップ 20 内部アプリケーションからインターネットにパケットを転送する ILB のルートを作成します。

```
gcloud beta compute routes create <ilb-route-name>
--network=<inside-vpc-name> --priority=1000 --destination-range=0.0.0.0/0
--next-hop-ilb=<ilb-forwarding-rule-name> --next-hop-ilb-region=<region>
```

例：

```
gcloud beta compute routes create demo-ilb --network=sdt-test-asav-inside
--priority=1000 --destination-range=0.0.0.0/0 --next-hop-ilb=demo-asav-fr-ilb
--next-hop-ilb-region=us-central1
Created [https://www.googleapis.com/compute/beta/projects/asavgcp-poc-4krn/global
/routes/demo-ilb].
```

Auto Scale ロジック

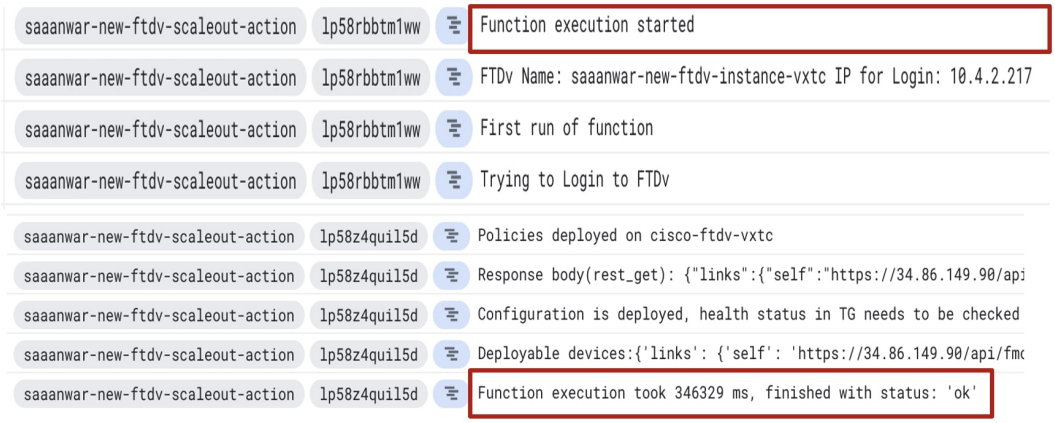
- オートスケーラは、ターゲット CPU 使用率レベルを、インスタンスグループ内の一定期間にわたるすべての vCPU の平均使用量の一部として扱います。
- 合計 vCPU の平均使用率がターゲット使用率を超えると、オートスケーラによって VM インスタンスが追加されます。合計 vCPU の平均使用率がターゲット使用率よりも低い場合、オートスケーラはインスタンスを削除します。
- たとえば、0.75 のターゲット使用率を設定すると、オートスケーラはインスタンスグループ内のすべての vCPU の平均使用率を 75% に維持するように指示されます。
- スケーリングの決定では、CPU 使用率メトリックのみが使用されます。
- このロジックは、ロードバランサがすべての Threat Defense Virtual に接続を均等に分散しようとし、平均してすべての Threat Defense Virtual が均等にロードされるという前提に基づいています。

ロギングとデバッグ

表示できるクラウド機能のログは以下のとおりです。

- スケールアウト機能のログ

図 *6*: スケールアウト機能のログ



上記のスケールアウト機能のログでは、**Function execution started** と **Function execution took 346329 ms, finish with status: 'ok'** のエントリは、機能ログの開始と終了をそれぞれ示しています。初回の機能実行、Threat Defense Virtual へのログイン、ポリシーの展開など、他の操作を追跡することもできます。

- スケールイン機能のログ

上記のスケールアウト機能のログでは、**Function execution started** と **Function execution took 50852 ms, finish with status: 'ok'** のエントリは、機能ログの開始と終了をそれぞれ示しています。登録解除プロセスの開始、登録解除のステータス、新しい認証トークンの取得など、他の操作を追跡することもできます。

トラブルシューティング

次に、Threat Defense Virtual Auto Scale for GCP の一般的なエラーシナリオとデバッグのヒントを示します。

- `main.py` が見つからない：Zip パッケージがファイルのみから作成されていることを確認します。クラウド機能に移動してファイルツリーを確認できます。フォルダがあってはいけません。
- テンプレートの展開中のエラー：「<>」内のすべてのパラメータ値が Jinja と YAML で入力されていることを確認します。または、同じ展開名が既に存在するかどうかを確認します。
- Google 関数が Threat Defense Virtual に到達できない：VPC コネクタが作成されており、YAML パラメータファイルで同じ名前が指定されていることを確認します。
- Threat Defense Virtual に SSH 接続中に認証に失敗：公開キーと秘密キーのペアが正しいことを確認します。
- 認証トークンが見つからない：シークレットの Management Center Virtual パスワードが正しいことを確認します。
- Threat Defense Virtual の異常とトラフィックの問題：ファイアウォールルールとルートに問題がないことを確認します。
- 手動で Threat Defense Virtual にログインできない：新しいパスワードを使用しているかを確認します。スケールアウト機能により旧パスワードは変更されます。
- Management Center Virtual にデバイスを登録できない：Threat Defense Virtual が Management Center Virtual から到達可能であるかを確認します。Threat Defense Virtual と Management Center Virtual の管理インターフェイスが同じサブネット内に存在する必要があります。
- 保持された接続により ILB と Threat Defense Virtual 間のループが形成されるため、正常性プローブ要求が開始されると CPU 使用率が高くなります。高い CPU 使用率を下げるには、次のいずれかのオプションを使用できます。

  オプション 1：Management Center Virtual でデータインターフェイスを無効にし、正常性プローブの NAT ルールを設定して、データインターフェイスを有効にします。データインターフェイスと NAT の詳細については、「インターフェイスの概要」と「ネットワークアドレス変換」を参照してください。

  オプション 2：正常性プローブの NAT ルールを Management Center Virtual から適用した後、Threat Defense Virtual のコンソールにログインし、**clear conn** コマンドを使用します。クラスタリングを設定している場合は、**cluster exec clear conn** コマンドを使用します。

  Threat Defense Virtual のコンソールで **show cpu** コマンドを使用して、CPU 使用率を確認します。

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。