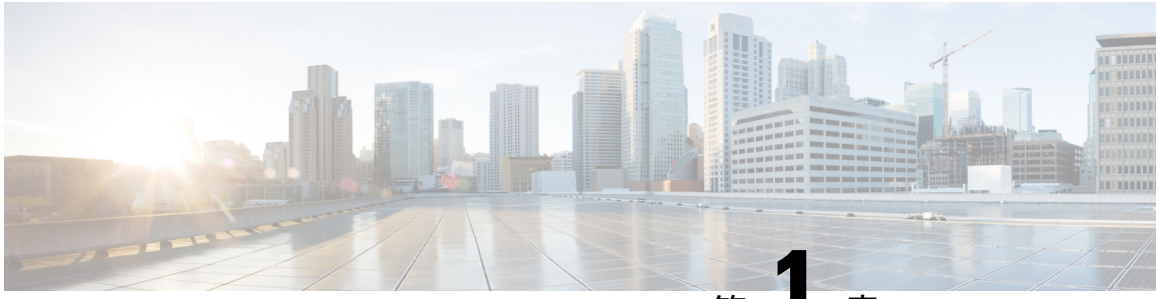
Cisco 適応型セキュリティ仮想アプライアンス（ASAv）9.17 スタートアップガイド

初版：2021 年 6 月 21 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（ **www.cisco.com/jp/go/safety_warning/** ）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

目 次



第 1 章

ASAv の概要

適応型セキュリティ仮想アプライアンス（ASAv）は、仮想化環境に包括的なファイアウォール機能を提供し、データセントートラフィックとマルチテナント環境のセキュリティを強化します。

ASDM または CLI を使用して、ASAv を管理およびモニタすることができます。その他の管理オプションを使用できる場合もあります。

ハイパーバイザのサポート

ハイパーバイザのサポートについては、Cisco ASA の互換性 [英語] を参照してください。

ASAv のライセンス

ASAv はシスコ スマート ソフトウェア ライセンシングを使用しています。詳細については、「Smart Software Licensing」を参照してください。

（注） ASAv にスマートライセンスをインストールする必要があります。ライセンスをインストールするまで、スループットは 100 Kbps に制限されるため、予備接続テストを実行できます。スマート ライセンスは、通常の操作に必要です。

9.13(1) 以降では、サポートされているすべての ASAv vCPU/メモリ構成ですべての ASAv ライセンスを使用できます。これにより、さまざまな VM リソースフットプリントに ASAv を導入できます。AnyConnect クライアント および TLS プロキシのセッション制限は、モデルタイプ

に関連付けられたプラットフォーム制限ではなく、インストールされた ASAv プラットフォームの権限付与によって決まります。

ASAv ライセンスの権限付与と、サポートされているプライベートおよびパブリック導入ターゲットのリソース仕様については、以降の各セクションを参照してください。

スマートライセンスの権限付与について

すべての ASAv ライセンスを、サポートされているすべての ASAv vCPU/メモリ構成で使用できます。これにより、さまざまな VM リソースフットプリントで ASAv を実行できます。また、サポート対象の AWS および Azure インスタンスタイプの数も増えます。ASAv マシンを構成する場合、サポートされる最大 vCPU 数は 16（ASAv100）個です。また、サポートされる最大メモリは 64GB RAM です。

重要 一度展開した ASAv インスタンスのリソース割り当て（メモリ、CPU、ディスク容量）は変更できません。何らかの理由でリソース割り当てを増やす必要がある場合（たとえば、ライセンス付与された権限を ASAv30/2Gbps から ASAv50/10Gbps に変更する場合）、必要なリソースを使用して新しいインスタンスを作成する必要があります。

- vCPU：ASAv は 1 ～ 16 個の vCPU をサポートします。
- メモリ：ASAv は 2 ～ 64 GB の RAM をサポートします。
- ディスクストレージ：ASAv はデフォルトで最大 8 GB の仮想ディスクをサポートします。ディスクサイズを 8 GB を超えて増やすことはできません。VM リソースをプロビジョニングする場合は、この点に注意してください。

重要 ASAv の最小メモリ要件は 2GB です。現在の ASAv が 2GB 未満のメモリで動作している場合、ASAv マシンのメモリを増やさないと、以前のバージョンからバージョン 9.13(1) 以降にアップグレードできません。また、最新バージョンを使用して新しい ASAv マシンを再導入できます。

1 つ以上の vCPU を使用して ASAv を導入する場合、ASAv の最小メモリ要件は 4GB です。

ライセンスされた機能のセッション制限

AnyConnect クライアントおよび TLS プロキシのセッション制限は、インストールされた ASAv プラットフォームの権限付与層によって決まり、レート制限の適用を受けます。次の表は、権限付与層とレート制限に基づくセッション制限をまとめたものです。

表 *1 :* 権限付与による *ASAv* セッションの制限

権限付与	AnyConnect クライアント Premium ピア	合計 TLS プロキシセッション	レートリミッタ
標準層、100M	50	500	150 Mbps
標準層、1G	250	500	1 Gbps
標準層、2G	750	1000	[2 Gbps]
標準層、10G	10,000	10,000	10 Gbps
標準層、20G	20,000	20,000	20 Gbps

前の表に示したように、権限付与によって付与されたセッション制限は、プラットフォームのセッション制限を超えることはできません。プラットフォームのセッション制限は、ASAv 用にプロビジョニングされたメモリ量に基づいて決まります。ASAv マシンの最大サイズは、8 個の vCPU と 64 GB のメモリです。

表 *2 :* メモリ要件による *ASAv* セッション制限

プロビジョニングされたメモリ	AnyConnect クライアント Premium ピア	合計 TLS プロキシセッション
2 GB ～ 7.9 GB	250	500
8 GB ～ 15.9 GB	750	1000
16 GB ～ 31.9 GB	10,000	10,000
32 GB ～ 64 GB	20,000	20,000

プラットフォームの制限

ファイアウォール接続、同時接続、および VLAN は、ASAv メモリに基づくプラットフォームの制限です。

（注） ASAv がライセンスされていない状態にある場合、ファイアウォール接続は 100 に制限されます。権限付与によってライセンスが付与されると、接続はプラットフォームの制限に移行します。ASAv の最小メモリ要件は 2GB です。

表 *3 :* プラットフォームの制限

ASAv のメモリ	ファイアウォールの接続、同時	VLANs
2 GB ～ 7.9 GB	100,000	50

ASAv のメモリ	ファイアウォールの接続、同時	VLANs
8 GB ～ 15.9 GB	500,000	200
16 GB ～ 31.9	2,000,000	1024
32 GB ～ 64 GB	4,000,000	1024

ASAv プライベートクラウドの権限付与（VMware、KVM、Hyper-V）

すべての ASAv ライセンスは、サポートされているすべての ASAv vCPU/メモリ構成で使用できるため、プライベートクラウド環境（VMware、KVM、Hyper-V）に ASAv を導入する場合の柔軟性が高まります。

（注） ASAv50 と ASAv100 は、HyperV ではサポートされません。

AnyConnect クライアントおよび TLS プロキシのセッション制限は、インストールされた ASAv プラットフォームの権限付与層によって決まり、レート制限の適用を受けます。次の表は、プライベートクラウド環境に導入された ASAv の権限付与層に基づくセッション制限と、適用されるレート制限をまとめたものです。

（注） ASAv セッション制限は、ASAv 用にプロビジョニングされたメモリの量に基づいています。表 2 : メモリ要件による ASAv セッション制限 （3 ページ）を参照してください。

表 *4 : VMware/KVM/HyperV* プライベートクラウドの *ASAv* : 権限付与に基づいてライセンスされた機能の制限

RAM（GB）		権限付与のサポート *				
最小	最大	標準層、100M	標準層、1G	標準層、2G	標準層、10G	標準層、20G
2	7.9	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500/20G
8	159	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G
16	319	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K/20G
32	64	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	20K/20K/20G
* AnyConnect クライアント セッション/TLS プロキシセッション/権限付与またはインスタンスごとのレート制限。						

ASAv パブリッククラウドの権限付与（AWS）

すべての ASAv ライセンスは、サポートされているすべての ASAv vCPU/メモリ構成で使用できるため、さまざまな AWS インスタンスタイプに ASAv を導入できます。AnyConnect クライアント および TLS プロキシのセッション制限は、インストールされた ASAv プラットフォームの権限付与層によって決まり、レート制限の適用を受けます。

次の表は、AWS インスタンスタイプの権限付与層に基づくセッション制限とレート制限をまとめたものです。サポートされているインスタンスの AWS VM の規模（vCPU とメモリ）の内訳については、「AWS クラウドへの ASAv の導入について」を参照してください。

表 *5 : AWS* 上の *ASAv* : 権限付与に基づくライセンス機能の制限

インスタンス	BYOL 権限付与のサポート *				PAYG **
	標準層、100M	標準層、1G	標準層、2G	標準層、10G	
c5.xlarge	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000
c5.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K
c4.large	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
c4.xlarge	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
c4.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	750/1000
c3.large	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
c3.xlarge	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
c3.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	750/1000
m4.large	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
m4.xlarge	50/500/100M	250/500/1G	250/500/2G	250/500/10G	10K/10K
m4.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K

* AnyConnect クライアント セッション/TLS プロキシセッション/権限付与またはインスタンスごとのレート制限。

** AnyConnect クライアント セッション/TLS プロキシセッション。PAYG モードではレート制限は使用されません。

Pay-As-You-Go（PAYG）モード

次の表に、毎時課金（PAYG）モードにおける各層のスマートライセンス権限付与の概要を示します。PAYG モードは、割り当てられたメモリに基づきます。

表 *6 : AWS* 上の *ASAv* : *PAYG* のスマートライセンス権限付与

RAM（GB）	毎時課金モードの権限付与
2 GB ～ 8 GB 未満	標準層、1G
8 GB ～ 16 GB 未満	標準層、2G
16 GB ～ 64 GB	標準層、10G

ASAv パブリッククラウドの権限付与（Azure）

すべての ASAv ライセンスは、サポートされているすべての ASAv vCPU/メモリ構成で使用できるため、さまざまな Azure インスタンスタイプに ASAv を導入できます。AnyConnect クライアント および TLS プロキシのセッション制限は、インストールされた ASAv プラットフォームの権限付与層によって決まり、レート制限の適用を受けます。

次の表は、Azure インスタンスタイプの権限付与層に基づくセッション制限とレート制限をまとめたものです。サポートされているインスタンスの Azure VM の規模（vCPU とメモリ）の内訳については、「Microsoft Azure クラウドへの ASAv の導入について」を参照してください。

（注） Pay-As-You-Go（PAYG）モードは現在、Azure 上の ASAv ではサポートされていません。

表 *7 : Azure* 上の *ASAv* : 権限付与に基づくライセンス機能の制限

インスタンス	BYOL 権限付与のサポート *				
	標準層、100M	標準層、1G	標準層、2G	標準層、10G	標準層、20G
D1、D1_v2DS1、DS1_v2	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500/20G
D2、D2_v2、DS2、DS2_v2	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500/20G
D3、D3_v2、DS3、DS3_v2	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G
D4、D4_v2、DS4、DS4_v2	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K/20G
D5、D5_v2、DS5、DS5_v2	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/20K/20G
D2_v3	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G

インスタンス	BYOL 権限付与のサポート *				
	標準層、100M	標準層、1G	標準層、2G	標準層、10G	標準層、20G
D4_v3	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K/20G
D8_v3	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K/20G
F4、F4s	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G
F8、F8s	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/20K/20G
F16、F16s	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/20K/20G
* AnyConnect クライアント セッション/TLS プロキシセッション/権限付与またはインスタンスごとのレート制限。					

注意事項と制約事項

ASAv ファイアウォール機能は ASA ハードウェア ファイアウォールとよく似ていますが、次のガイドラインと制限事項があります。

ASAv（すべての権限付与）のガイドラインと制限事項

スマートライセンスのガイドライン

- サポートされる vCPU の最大数は 8 です。また、サポートされる最大メモリは 64 GB RAM です。すべての ASAv ライセンスを、サポートされているすべての ASAv vCPU/メモリ構成で使用できます。
- ライセンスされた機能およびライセンスされていないプラットフォーム機能のセッション制限は、VM メモリの量に基づいて設定されます。
- AnyConnect クライアントおよび TLS プロキシのセッション制限は、ASAv プラットフォームの権限付与によって決定されます。セッション制限は、ASAv モデルタイプ（ASAv5/10/30/50/100）に関連付けられなくなりました。
- セッション制限には最小メモリ要件があります。VM メモリが最小要件を下回っている場合、セッション制限はそのメモリ量でサポートされる最大数に設定されます。
- 既存の権限付与に変更はありません。権限付与 SKU と表示名には、引き続きモデル番号（ASAv5/10/30/50/100）が含まれます。
- 権限付与は、レート制限を介して最大スループットを設定します。
- お客様の発注プロセスに変更はありません。

ディスク ストレージ

ASAv は、デフォルトで最大 8 GB の仮想ディスクをサポートします。ディスクサイズを 8 GB を超えて増やすことはできません。VM リソースをプロビジョニングする場合は、この点に注意してください。

コンテキスト モードのガイドライン

シングル コンテキスト モードでだけサポートされます。マルチ コンテキスト モードをサポートしません。

ハイ アベイラビリティ ガイドラインのためのフェールオーバー

フェールオーバー配置の場合は、スタンバイ装置が同じライセンス権限付与を備えていることを確認してください（たとえば、両方の装置が 2Gbps の権限付与であることなど）。

重要 ASAv を使用して高可用性ペアを作成する場合は、データインターフェイスを各 ASAv に同じ順序で追加する必要があります。完全に同じインターフェイスが異なる順序で各 ASAv に追加されると、ASAv コンソールにエラーが表示されることがあります。また、フェールオーバー機能にも影響が出ることがあります。

サポートしない ASA 機能

ASAv は、次の ASA 機能をサポートしません。

- クラスタリング（KVM と VMware を除くすべての権限付与）
- マルチ コンテキスト モード
- アクティブ/アクティブ フェールオーバー
- EtherChannel
- AnyConnect Premium（共有）ライセンス

制限事項

- ASAv は、x710 NIC の 1.9.5 i40en ホストドライバと互換性がありません。これより古いバージョンまたは新しいバージョンのドライバは動作します。（VMware のみ）

1 GB 権限付与のガイドラインと制限事項

パフォーマンスのガイドライン

- 9 つ以上の設定済み e1000 インターフェイスを使用した 1 GB プラットフォームのジャンボフレーム予約によって、デバイスがリロードされる場合があります。**ジャンボ フレーム予約**が有効になっている場合は、インターフェイスの数を 8 つ以下に減らしてください。

インターフェイスの正確な数は、その他の構成済み機能の操作で必要となるメモリの量によって異なりますが、8 つより少なくすることはできます。

10 GB 権限付与のガイドラインと制限事項

パフォーマンスのガイドライン

- 集約トラフィックで 10 Gbps がサポートされます。
- ASAv のパフォーマンスを向上させるために、次のプラクティスがサポートされています。
  - NUMA ノード
  - 複数の RX キュー
  - SR-IOV プロビジョニング
  - 詳細については、VMware での ASAv のパフォーマンス調整（40 ページ）および KVM での ASAv のパフォーマンス調整 （60 ページ）を参照してください。
- フルスループットレートを実現するため、CPU ピンニングを推奨します。ESXi 構成でのパフォーマンスの向上 （40 ページ）および KVM 構成でのパフォーマンスの向上 （60 ページ）を参照してください。
- ジャンボフレーム予約で e1000 インターフェイスと i40e-vf インターフェイスが混在していると、i40e-vf インターフェイスがダウン状態のままになる場合があります。**ジャンボフレーム予約**が有効になっている場合は、e1000 ドライバと i40e-vf ドライバを使用するインターフェイスのタイプが混在しないようにしてください。

制限事項

- トランスペアレント モードはサポートされていません。
- ASAv は、x710 NIC の 1.9.5 i40en ホストドライバと互換性がありません。これより古いバージョンまたは新しいバージョンのドライバは動作します。（VMware のみ）
- Hyper-V ではサポートされていません。

20 GB 権限付与のガイドラインと制限事項

パフォーマンスのガイドライン

- 集約トラフィックで 20 Gbps がサポートされます。
- ASAv のパフォーマンスを向上させるために、次のプラクティスがサポートされています。
  - NUMA ノード
  - 複数の RX キュー

- SR-IOV プロビジョニング
- 詳細については、VMware での ASAv のパフォーマンス調整（40 ページ）およびKVM での ASAv のパフォーマンス調整 （60 ページ）を参照してください。

- フルスループットレートを実現するため、CPU ピンニングを推奨します。ESXi 構成でのパフォーマンスの向上 （40 ページ）およびKVM 構成でのパフォーマンスの向上 （60 ページ）を参照してください。

制限事項

- ASAv は、x710 NIC の 1.9.5 i40en ホストドライバと互換性がありません。これより古いバージョンまたは新しいバージョンのドライバは動作します。（VMware のみ）
- トランスペアレント モードはサポートされていません。
- Amazon Web Services（AWS）および Hyper-V ではサポートされません。

ASAv インターフェイスおよび仮想 NIC

ASAv は、仮想プラットフォーム上のゲストとして、基盤となる物理プラットフォームのネットワーク インターフェイスを利用します。ASAv の各インターフェイスは仮想 NIC（vNIC）にマッピングされます。

- ASAv のインターフェイス
- サポートされている vNIC

ASAv のインターフェイス

ASAv は、次のギガビット イーサネット インターフェイスがあります。

- Management 0/0

  AWS と Azure の場合は、Management 0/0 をトラフィック伝送用の「外部」インターフェイスにすることができます。

- GigabitEthernet 0/0 ～ 0/8。ASAv をフェールオーバー ペアの一部として展開する場合は GigabitEthernet 0/8 がフェールオーバー リンクに使用されることに注意してください。

（注） 構成を簡単に移行できるように、Ten GigabitEthernet（VMXNET3 ドライバで使用可能なインターフェイスなど）には GigabitEthernet というラベルが付いています。これは表面的なものであり、実際のインターフェイス速度には影響しません。

ASAv では、E1000 ドライバを 1 Gbps リンクとして使用してギガビットイーサネットインターフェイスが定義されます。VMware では E1000 ドライバの使用が推奨されなくなっていることに注意してください。

- Hyper-V は最大 8 つのインターフェイスをサポートします。Management 0/0 および GigabitEthernet 0/0 ～ 0/6。フェールオーバー リンクとして GigabitEthernet 0/6 を使用できます。

サポートされている vNIC

ASAv では次の vNIC がサポートされています。同じ ASAv での vNIC の混在（e1000 と vmxnet3 など）はサポートされていません。

表 *8*：サポートされている *vNIC*

vNIC のタイプ	ハイパーバイザのサポート		ASAv バージョン	注意
	VMware	KVM		
VMXNET3	対応	×	9.9(2) 以降	VMware のデフォルト vmxnet3 を使用する場合は、TCP パフォーマンスの低下を避けるために大量受信オフロード（LRO）を無効にする必要があります。VMware および VMXNET3 の LRO を無効にします （12 ページ）を参照してください。
e1000	対応	対応	9.2(1) 以降	VMware では推奨されません。
virtio	×	対応	9.3(2.200) 以降	KVM のデフォルト
ixgbe-vf	対応	対応	9.8(1) 以降	AWS のデフォルト。SR-IOV サポート用の ESXi と KVM。
i40e-vf	×	対応	9.10(1) 以降	SR-IOV サポート用の KVM。

VMware および VMXNET3 の LRO を無効にします

Large Receive Offload（LRO）は、CPU オーバーヘッドを削減することによって、高帯域幅ネットワーク接続のインバウンドスループットを向上させる手法です。これは、1 つのストリームからの複数の着信パケットを大きなバッファに集約してから、ネットワークスタックの上位に渡されるようにすることによって、処理する必要があるパケットの数を減らすことによって機能します。ただし、LRO は、ネットワークパケット配信のフローが一貫せず、輻輳しているネットワークで「バースト」する可能性がある場合に、TCP パフォーマンスの問題を引き起こす可能性があります。

重要 VMware は、デフォルトで LRO を有効にして、全体的なスループットを向上させます。したがって、このプラットフォームで ASAv 導入の LRO を無効にする必要があります。

ASAv マシンで LRO を直接無効化できます。設定変更を行う前に、仮想マシンの電源をオフにします。

1. vSphere Web Client インベントリで ASAv マシンを検索します。
    1. 仮想マシンを検索するには、データセンター、フォルダ、クラスタ、リソースプール、またはホストを選択します。
    2. [Related Objects] タブをクリックし、[Virtual Machines] タブをクリックします。
2. 仮想マシンを右クリックして、[Edit Settings] をクリックします。
3. [VM Options] をクリックします。
4. [Advanced] を展開します。
5. [Configuration Parameters] の下で、[Edit Configuration] ボタンをクリックします。
6. [Add Parameter] をクリックし、LRO パラメータの名前と値を入力します。
    - Net.VmxnetSwLROSL | 0
    - Net.Vmxnet3SwLRO | 0
    - Net.Vmxnet3HwLRO | 0
    - Net.Vmxnet2SwLRO | 0
    - Net.Vmxnet2HwLRO | 0

（注） オプションで、LRO パラメータが存在する場合は、値を調べて必要に応じて変更できます。パラメータが 1 に等しい場合、LRO は有効です。0 に等しい場合、LRO は無効です。

7. [OK] をクリックして変更を保存し、[Configuration Parameters] ダイアログボックスを終了します。

8. [保存（Save）] をクリックします。

詳細については、次の VMware サポート記事を参照してください。

- VMware KB 1027511
- VMware KB 2055140

ASAv と SR-IOV インターフェイスのプロビジョニング

Single Root I/O Virtualization（SR-IOV）により、さまざまなゲスト オペレーティング システムを実行している複数の VM が、ホストサーバー内の単一の PCIe ネットワークアダプタを共有できるようになります。SR-IOV では、VM がネットワーク アダプタとの間で直接データを移動でき、ハイパーバイザをバイパスすることで、ネットワークのスループットが増加しサーバーの CPU 負荷が低下します。最近の x86 サーバー プロセッサには、SR-IOV に必要なダイレクト メモリの転送やその他の操作を容易にする Intel VT-d テクノロジーなど、チップセットの拡張機能が搭載されています。

SR-IOV 仕様では、次の 2 つのデバイス タイプが定義されています。

- 物理機能（PF）：基本的にスタティック NIC です。PF は、SR-IOV 機能を含む完全な PCIe デバイスです。PF は、通常の PCIe デバイスとして検出、管理、設定されます。単一 PF は、一連の仮想関数（VF）の管理および設定を提供できます。
- Virtual Function（VF）：ダイナミック vNIC に似ています。VF は、データ移動に必要な最低限のリソースを提供する、完全または軽量の仮想 PCIe デバイスです。VF は直接的には管理されず、PF を介して配信および管理されます。1 つ以上の VF を 1 つの VM に割り当てることができます。

SR-IOV は、PCI 標準の開発および管理が公認されている業界組織である Peripheral Component Interconnect Special Interest Group（PCI SIG）によって定義および管理されています。SR-IOV の詳細については、『PCI-SIG SR-IOV Primer: An Introduction to SR-IOV Technology』を参照してください。

ASAv 上で SR-IOV インターフェイスをプロビジョニングするには、適切なオペレーティング システム レベル、ハードウェアと CPU、アダプタタイプ、およびアダプタの設定から始める計画が必要です。

SR-IOV インターフェイスに関するガイドラインと制限事項

ASAv の導入に使用する具体的なハードウェアは、サイズや使用要件によって異なります。ASAv のライセンス （1 ページ） には、さまざまな ASAv プラットフォームに関するライセンスの権限付与条件に準拠するリソースシナリオが説明されています。加えて、SR-IOV 仮想機能には特定のシステム リソースが必要です。

ホスト オペレーティング システムとハイパーバイザ サポート

SR-IOV サポートと VF ドライバは、以下で使用できます。

- Linux 2.6.30 カーネル以降

SR-IOV インターフェイスを備えた ASAv は、現在、次のハイパーバイザでサポートされています。

- VMware vSphere/ESXi
- QEMU/KVM
- AWS

ハードウェア プラットフォーム サポート

（注） サポートされている仮想化プラットフォームを実行できる任意のサーバークラスの x86 CPU デバイスに ASAv を導入する必要があります。

このセクションでは、SR-IOV インターフェイスに関するハードウェア ガイドラインについて説明します。以下はガイドラインであって要件ではありませんが、このガイドラインに従っていないハードウェアを使用すると、機能の問題や性能の低下につながる可能性があります。

SR-IOV をサポートしており、SR-IOV 対応 PCIe アダプタを搭載したサーバーが必要です。以下のハードウェア検討事項に留意する必要があります。

- 使用可能な VF の数を含む SR-IOV NIC の機能は、ベンダーやデバイスによって異なります。
- すべての PCIe スロットが SR-IOV をサポートしているわけではありません。
- SR-IOV 対応 PCIe スロットは機能が異なる場合があります。

（注） メーカーのマニュアルで、お使いのシステムの SR-IOV サポートを確認する必要があります。

- VT-d 対応のチップセット、マザーボード、および CPU については、『virtualization-capable IOMMU supporting hardware』を参照してください。VT-d は、SR-IOV システムに必須の BIOS 設定です。
- VMware の場合は、オンラインの『Compatibility Guide』で SR-IOV サポートを検索できます。
- KVM の場合は、『CPU compatibility』を確認できます。KVM 上の ASAv では、x86 ハードウェアしかサポートされないことに注意してください。

（注） シスコでは、ASAv を Cisco UCS C シリーズ ラックサーバー でテストしました。Cisco UCS-B サーバーは ixgbe-vf vNIC をサポートしていないことに注意してください。

SR-IOV でサポートされている NIC

- Intel イーサネット ネットワーク アダプタ X710

注目 ASAv は、x710 NIC の 1.9.5 i40en ホストドライバと互換性がありません。これより古いバージョンまたは新しいバージョンのドライバは動作します。（VMware のみ）

- Intel Ethernet Server Adapter X520 - DA2

CPU

- x86_64 マルチコア CPU

  Intel Sandy Bridge 以降（推奨）

（注） シスコでは、ASAv を 2.3GHz の Intel Broadwell CPU（E5-2699-v4）でテストしました。

- コア
  - CPU ソケットあたり 8 個以上の物理コア
  - 単一のソケット上で 8 コアにする必要があります。

（注） CPU ピンニングは、ASAv50 および ASAv100 上でフルスループットレートを実現するために推奨されています。ESXi 構成でのパフォーマンスの向上 （40 ページ）とKVM 構成でのパフォーマンスの向上 （60ページ）を参照してください。

BIOS 設定

SR-IOV は、BIOS だけでなく、ハードウェアで実行しているオペレーティング システム インスタンスまたはハイパーバイザのサポートも必要です。システム BIOS で次の設定をチェックします。

- SR-IOV が有効になっている。
- VT-x（仮想化テクノロジー）が有効になっている。
- VT-d が有効になっている。
- （オプション）ハイパースレッディングが無効になっている。

システムごとに BIOS 設定にアクセスして変更する方法が異なるため、ベンダーのマニュアルでプロセスを確認することをお勧めします。

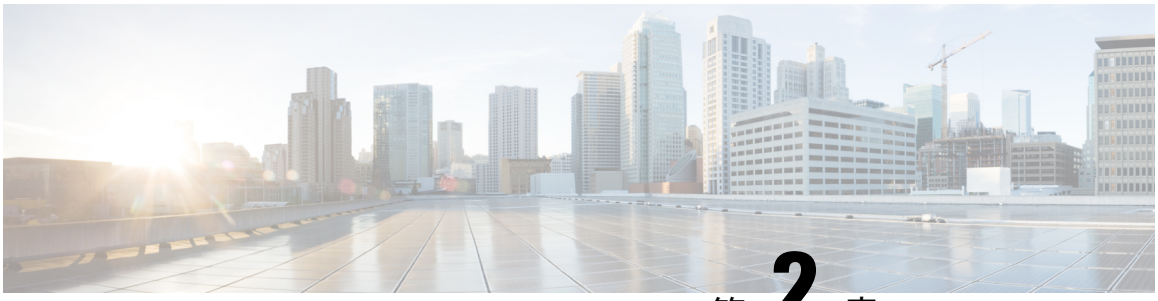
制限事項

ixgbe-vf インターフェイスを使用する場合、次の制限事項があります。

- ゲスト VM では、VF を無差別モードに設定できません。そのため、ixgbe-vf の使用時はトランスペアレント モードがサポートされません。
- ゲスト VM では、VF 上で MAC アドレスを設定できません。そのため、HA 中は MAC アドレスが転送されません。他の ASA プラットフォームや他のインターフェイス タイプを使用した場合は転送されます。HA フェールオーバーは、IP アドレスをアクティブからスタンバイに移行することによって機能します。

（注） この制限は、i40e-vf インターフェイスにも適用されます。

- Cisco UCSB サーバーは ixgbe-vf の vNIC をサポートしません。
- フェールオーバー セットアップでは、ペアになっている ASAv（プライマリ装置）に障害が発生すると、スタンバイ ASAv 装置がプライマリ装置のロールを引き継ぎ、そのインターフェイス IP アドレスがスタンバイ ASAv 装置の新しい MAC アドレスで更新されます。その後、ASAv は Gratuitous Address Resolution Protocol（ARP）更新を送信して、インターフェイス IP アドレスの MAC アドレスの変更を同じネットワーク上の他のデバイスに通知します。ただし、インターフェイスタイプの非互換性により、Gratuitous ARP 更新は、インターフェイス IP アドレスをグローバル IP アドレスに変換するための NAT または PAT ステートメントで定義されているグローバル IP アドレスに送信されません。



第 2 章

VMware を使用した ASAv の導入

ASAv は、VMware ESXi を実行できる任意のサーバークラスの x86 CPU デバイスに導入できます。

重要 ASAv の最小メモリ要件は 2GB です。現在の ASAv が 2GB 未満のメモリで動作している場合、ASAv マシンのメモリを増やさないと、以前のバージョンから 9.13(1) 以降にアップグレードできません。また、最新バージョンを使用して新しい ASAv マシンを再導入できます。

VMware での ASAv のガイドラインと制限事項

ESXi サーバーに ASAv の複数のインスタンスを作成して導入できます。ASAv の導入に使用される特定のハードウェアは、導入されるインスタンスの数や使用要件によって異なります。作成する各仮想アプライアンスには、ホストマシン上での最小リソース割り当て（メモリ、CPU 数、およびディスク容量）が必要です。

重要 ASAv は、8GB のディスクストレージサイズで導入されます。ディスク容量のリソース割り当てを変更することはできません。

ASAv を導入する前に、次のガイドラインと制限事項を確認します。

VMWare ESXi での ASAv のシステム要件

最適なパフォーマンスを確保するために、以下の仕様に準拠していることを確認してください。ASAvASAv には、次の要件があります。

- ホスト CPU は、仮想化拡張機能を備えたサーバークラスの x86 ベースの Intel または AMD CPU である必要があります。

  たとえば、ASAv パフォーマンステストラボでは、2.6GHz で動作する Intel® Xeon® CPU E5-2690v4 プロセッサを搭載した Cisco Unified Computing System™（Cisco UCS®）C シリーズ M4 サーバーを最低限使用しています。

- ASAv は、ESXi バージョン 6.0、6.5、6.7、および 7.0 をサポートしています。

推奨される vNIC

最適なパフォーマンスを得るためには、次の vNIC を推奨します。

- PCI パススルーでの i40e：サーバーの物理 NIC を VM に関連付け、DMA（ダイレクトメモリアクセス）を介して NIC と VM の間でパケットデータを転送します。パケットの移動に CPU サイクルは必要ありません。

- i40evf/ixgbe-vf：実質的に上記と同じですが（NIC と VM 間の DMA パケット）、NIC を複数の VM 間で共有できます。SR-IOV は、導入の柔軟性が高いため、一般的に推奨されます。注意事項と制約事項 （45 ページ）を参照してください。

- vmxnet3：10Gbps の動作をサポートしますが、CPU サイクルも必要な準仮想化ネットワークドライバです。これが VMware のデフォルトです。

  vmxnet3 を使用する場合は、TCP パフォーマンスの低下を避けるために大量受信オフロード（LRO）を無効にする必要があります。

パフォーマンスの最適化

ASAv 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、VMware での ASAv のパフォーマンス調整 （40 ページ）を参照してください。

- **NUMA**：ゲスト VM の CPU リソースを単一の Non-Uniform Memory Access（NUMA）ノードに分離することで、ASAv のパフォーマンスを向上できます。詳細については、NUMA のガイドライン （40 ページ）を参照してください。

- **Receive Side Scaling**：ASAv 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 9.13(1) 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 （42 ページ）を参照してください。
- **VPN の最適化**：ASAv で VPN パフォーマンスを最適化するための追加の考慮事項については、VPN の最適化 （66 ページ）を参照してください。

クラスタリング

バージョン 9.17 以降、クラスタリングは VMware で展開された ASA 仮想インスタンスでサポートされます。詳細については、「ASA Cluster for the ASAv」を参照してください。

OVF ファイルのガイドライン

導入対象に基づいて、asav-vi.ovf ファイルまたは asav-esxi.ovf ファイルを選択します。

- asav-vi：vCenter に導入する場合
- asav-esxi：ESXi に導入する場合（vCenter なし）
- ASAv OVF の導入は、ローカリゼーション（非英語モードでのコンポーネントのインストール）をサポートしません。ご自身の環境の VMware vCenter と LDAP サーバーが ASCII 互換モードでインストールされていることを確認してください。
- ASAv をインストールして VM コンソールを使用する前に、キーボードを [United States English] に設定する必要があります。
- ASAv を導入すると、2 つの異なる ISO イメージが ESXi ハイパーバイザにマウントされます。
  - マウントされた最初のドライブには、vSphere によって生成された OVF 環境変数が備わってます。
  - マウントされた 2 番目のドライブは day0.iso です。

注目 ASAv マシンが起動したら、両方のドライブのマウントを解除できます。ただし、[電源投入時に接続（Connect at Power On）] がオフになっている場合でも、ドライブ 1（OVF 環境変数を使用）は、ASAv の電源をオフ/オンにするたびに常にマウントされます。

OVF テンプレートのガイドラインのエクスポート

vSphere の OVF テンプレートのエクスポート機能は、既存の ASAv インスタンスパッケージを OVF テンプレートとしてエクスポートするのに役立ちます。エクスポートされた OVF テンプレートを使用して、同じ環境または異なる環境に ASAv インスタンスを導入できます。エクス

ポートされた OVF テンプレートを使用して vSphere に ASAv インスタンスを導入する前に、OVF ファイルの構成の詳細を変更して、導入の失敗を防ぐ必要があります。

ASAv のエクスポートされた OVF ファイルを変更するには、次の手順を実行します。

1. OVF テンプレートをエクスポートしたローカルマシンにログインします。
2. テキストエディタで OVF ファイルを参照して開きます。
3. `<vmw:ExtraConfig vmw:key="monitor_control.pseudo_perfctr" vmw:value="TRUE"></vmw:ExtraConfig>` タグが存在することを確認します。
4. `<rasd:ResourceSubType>vmware.cdrom.iso</rasd:ResourceSubType>` タグを削除します。

   または

   `<rasd:ResourceSubType>vmware.cdrom.iso</rasd:ResourceSubType>` タグと `<rasd:ResourceSubType>vmware.cdrom.remotepassthrough</rasd:ResourceSubType>` タグを交換します。

   詳細については、VMware が公開した「Deploying an OVF fails on vCenter Server 5.1/5.5 when VMware tools are installed (2034422)」を参照してください。
5. `UserPrivilege`、`OvfDeployment`、および `ControllerType` のプロパティ値を入力します。

   次に例を示します。

```
- <Property ovf:qualifiers="ValueMap{"ovf", "ignore", "installer"}" ovf:type="string"
 ovf:key="OvfDeployment">
+ <Property ovf:qualifiers="ValueMap{"ovf", "ignore", "installer"}" ovf:type="string"
 ovf:key="OvfDeployment" ovf:value="ovf">

- <Property ovf:type="string" ovf:key="ControllerType">
+ <Property ovf:type="string" ovf:key="ControllerType" ovf:value="ASAv">

- <Property ovf:qualifiers="MinValue(0) MaxValue(255)" ovf:type="uint8"
ovf:key="UserPrivilege">
+ <Property ovf:qualifiers="MinValue(0) MaxValue(255)" ovf:type="uint8"
ovf:key="UserPrivilege" ovf:value="15">
```

6. OVF ファイルを保存します。
7. OVF テンプレートを使用して、ASAv を導入します。VMware vSphere Web Client を使用した ASAv の導入 [英語] を参照してください。

ハイ アベイラビリティ ガイドラインのためのフェールオーバー

フェールオーバー配置の場合は、スタンバイ装置が同じライセンス権限付与を備えていることを確認してください（たとえば、両方の装置が 2Gbps の権限付与であることなど）。

重要 ASAv を使用して高可用性ペアを作成する場合は、データインターフェイスを各 ASAv に同じ順序で追加する必要があります。完全に同じインターフェイスが異なる順序で各 ASAv に追加されると、ASAv コンソールにエラーが表示されることがあります。また、フェールオーバー機能にも影響が出ることがあります。

ASAv 内部インターフェイスまたは ASAv フェールオーバーの高可用性リンクに使用される ESX ポートグループについては、2 つの仮想 NIC を使用して ESX ポートグループのフェールオーバー順序を設定します（1 つはアクティブアップリンク、もう 1 つはスタンバイアップリンク）。この設定は、2 つの VM が相互に ping を実行したり、ASAv 高可用性リンクを稼働させたりするために必要です。

vMotion に関するガイドライン

- VMware では、vMotion を使用する場合、共有ストレージのみを使用する必要があります。ASAv の導入時に、ホストクラスタがある場合は、ストレージをローカルに（特定のホスト上）または共有ホスト上でプロビジョニングできます。ただし、ASAv を vMotion を使用して別のホストに移行する場合、ローカル ストレージを使用するとエラーが発生します。

スループット用のメモリと vCPU の割り当てとライセンス

- ASAv に割り当てられたメモリのサイズは、スループットレベルに合わせたものです。異なるスループットレベルのライセンスを要求する場合を除いて、[Edit Settings] ダイアログボックスのメモリ設定または vCPU ハードウェア設定は変更しないでください。アンダープロビジョニングは、パフォーマンスに影響を与える可能性があります。

（注） メモリまたは vCPU ハードウェア設定を変更する必要がある場合は、ASAv のライセンス （1 ページ）に記載されている値のみを使用してください。VMware が推奨するメモリ構成の最小値、デフォルト値、および最大値は使用しないでください。

CPU 予約

- デフォルトでは、ASAv の CPU 予約は 1000 MHz です。共有、予約、および制限の設定（[設定の編集（Edit Settings）] > [リソース（Resources）] > [CPU]）を使用することで、ASAv に割り当てられる CPU リソースの量を変更できます。より低い設定で必要なトラフィック負荷が課されている状況で ASAv が目的を達成できる場合は、CPU 予約の設定を 1000 Mhz 未満にできます。ASAv によって使用される CPU の量は、動作しているハードウェアプラットフォームだけでなく、実行している作業のタイプと量によっても異なります。

  仮想マシンの [Performance] タブの [Home] ビューに配置された [CPU Usage (MHz)] チャートから、すべての仮想マシンに関する CPU 使用率をホストの視点で確認できます。ASAv が標準的なトラフィック量を処理しているときの CPU 使用率のベンチマークを設定すると、その情報を CPU 予約の調整時の入力として使用できます。

  詳細については、VMware から発行されている『CPU Performance Enhancement Advice』を参照してください。

- リソース割り当てとオーバープロビジョニングまたはアンダープロビジョニングされたリソースを表示するには、ASAv **show vm** および **show cpu** コマンド、あるいは ASDM **[ホーム（Home）] > [デバイスダッシュボード（Device Dashboard）] > [デバイス情報（Device Information）] > [仮想リソース（Virtual Resources）]** タブまたは **[モニタリング（Monitoring）] > [プロパティ（Properties）] > [システムリソースグラフ（System Resources Graphs）] > [CPU]** ペインを使用できます。

UCS B シリーズ ハードウェアにおけるトランスペアレント モードに関するガイドライン

MAC フラップが、Cisco UCS B シリーズ ハードウェアのトランスペアレントモードで動作する一部の ASAv 設定で発生することがあります。MAC アドレスがさまざまな場所で出現した場合、パケットはドロップされます。

VMware 環境にトランスペアレントモードで ASAv を導入する場合に MAC フラップを回避するには、次のガイドラインを参考にしてください。

- VMware NIC チーミング：UCS B シリーズにトランスペアレントモードで ASAv を導入する場合、内部および外部インターフェイスに使用するポートグループにはアクティブアップリンクを 1 つだけ設定し、アップリンクは同じである必要があります。vCenter で VMware NIC チーミングを設定します。

  NIC チーミングの設定方法の詳細については、VMware ドキュメントを参照してください。

- ARP インスペクション：ASAv で ARP インスペクションを有効にし、受信インターフェイスで MAC および ARP エントリを静的に設定します。ARP インスペクションと有効化の詳細については、Cisco ASA シリーズ コンフィギュレーション ガイド（一般的な操作）[英語] を参照してください。

その他のガイドラインと制限事項

- ESXi 6.7、vCenter 6.7、ASA Virtual 9.12 以降を実行している場合、ASA Virtual は 2 つの CD/DVD IDE ドライブなしで起動します。
- vSphere Web Client は ASAv OVA の導入ではサポートされないため、vSphere Client を使用してください。

ASAv の VMware 機能のサポート

次の表に、ASAv の VMware 機能のサポートを示します。

表 *9 : ASAv の VMware* 機能のサポート

機能	説明	サポート（あり/なし）	コメント
コールド クローン	クローニング中に VM の電源がオフになります。	あり	–

機能	説明	サポート（あり/なし）	コメント
DRS	動的リソースのスケジューリングおよび分散電源管理に使用されます。	Yes	VMware のガイドラインを参照してください。
ホット追加	追加時に VM が動作しています。	なし	–
ホット クローン	クローニング中に VM が動作しています。	なし	–
ホット リムーブ	取り外し中に VM が動作しています。	なし	–
スナップショット	VM が数秒間フリーズします。	あり	使用には注意が必要です。トラフィックが失われる可能性があります。フェールオーバーが発生することがあります。
一時停止と再開	VM が一時停止され、その後再開します。	あり	–
vCloud Director	VM の自動配置が可能になります。	なし	–
VM の移行	移行中に VM の電源がオフになります。	あり	–
VMotion	VM のライブ マイグレーションに使用されます。	あり	共有ストレージを使用します。vMotion に関するガイドライン（21 ページ）を参照してください。
VMware FT	VM の HA に使用されます。	なし	ASAv マシンの障害に対して ASAv のフェールオーバーを使用します。
VMware HA	ESXi およびサーバーの障害に使用されます。	あり	ASAv マシンの障害に対して ASAv のフェールオーバーを使用します。

機能	説明	サポート（あり/なし）	コメント
VM ハートビートの VMware HA	VM 障害に使用されます。	なし	ASAv マシンの障害に対して ASAv のフェールオーバーを使用します。
VMware vSphere スタンドアロン Windows クライアント	VM を導入するために使用されます。	あり	–
VMware vSphere Web Client	VM を導入するために使用されます。	あり	–

ASAv と VMware の前提条件

VMware vSphere Web Client、vSphere スタンドアロンクライアント、または OVF ツールを使用して ASAv を導入できます。システム要件については、Cisco ASA の互換性 [英語] を参照してください。

vSphere 標準スイッチのセキュリティ ポリシー

vSphere スイッチについては、レイヤ 2 セキュリティ ポリシーを編集して、ASAv インターフェイスによって使用されるポート グループに対しセキュリティ ポリシーの例外を適用できます。次のデフォルト設定を参照してください。

- 無差別モード：**拒否**
- MAC アドレスの変更：**許可**
- 不正送信：**許可**

次の ASAv 設定の場合、これらの設定の変更が必要な場合があります。詳細については、vSphere のマニュアルを参照してください。

表 *10*：ポート グループのセキュリティ ポリシーの例外

	ルーテッド ファイアウォール モード		トランスペアレント ファイアウォール モード	
セキュリティの例外	フェールオーバーなし	フェールオーバー	フェールオーバーなし	フェールオーバー
無差別モード	<任意>	<任意>	承認	承認
MAC アドレスの変更	<任意>	承認	<任意>	承認

セキュリティの例外	ルーテッド ファイアウォール モード		トランスペアレント ファイアウォール モード	
	フェールオーバーなし	フェールオーバー	フェールオーバーなし	フェールオーバー
不正送信	<任意>	承認	承認	承認

ASAv ソフトウェアの解凍と第 0 日用構成ファイルの作成

ASAv を起動する前に、第 0 日用のコンフィギュレーション ファイルを準備できます。このファイルは、ASAv の起動時に適用される ASAv の設定を含むテキストファイルです。この初期設定は、「day0-config」というテキスト ファイルとして指定の作業ディレクトリに格納され、さらに day0.iso ファイルへと処理されます。この day0.iso ファイルが最初の起動時にマウントされて読み取られます。第 0 日用コンフィギュレーション ファイルには、少なくとも、管理インターフェイスをアクティブ化するコマンドと、公開キー認証用 SSH サーバーを設定するコマンドを含める必要がありますが、すべての ASA 設定を含めることもできます。空の day0-config を含むデフォルトの day0.iso がリリースとともに提供されています。day0.iso ファイル（カスタム day0 またはデフォルトの day0.iso）は、最初の起動中に使用できなければなりません。

始める前に

この例では Linux が使用されていますが、Windows の場合にも同様のユーティリティがあります。

- 初期導入時に自動的に ASAv にライセンスを付与するには、Cisco Smart Software Manager からダウンロードした Smart Licensing Identity（ID）トークンを「idtoken」というテキスト ファイルに格納し、第 0 日用構成ファイルと同じディレクトリに保存します。
- 仮想 VGA コンソールではなく、ハイパーバイザのシリアルポートから ASAv にアクセスし、設定する場合は、第 0 日用構成ファイルに**コンソール シリアル**の設定を追加して初回ブート時にシリアルポートを使用する必要があります。
- トランスペアレントモードで ASAv を導入する場合は、トランスペアレントモードで実行される既知の ASA 構成ファイルを、第 0 日用構成ファイルとして使用する必要があります。これは、ルーテッド ファイアウォールの第 0 日用コンフィギュレーション ファイルには該当しません。
- ISO イメージが ESXi ハイパーバイザにどのようにマウントされるかの詳細については、VMware での ASAv のガイドラインと制限事項 （17 ページ）の OVF ファイルのガイドラインを参照してください。

ステップ 1 ZIP ファイルを Cisco.com からダウンロードし、ローカル ディスクに保存します。

https://www.cisco.com/go/asa-software

（注） Cisco.com のログインおよびシスコ サービス契約が必要です。

ステップ 2 ファイルを作業ディレクトリに解凍します。ディレクトリからファイルを削除しないでください。次のファイルが含まれています。

- asav-vi.ovf：vCenter への導入用。
- asav-esxi.ovf：vCenter 以外への導入用。
- boot.vmdk：ブート ディスク イメージ。
- disk0.vmdk：ASAv ディスクイメージ。
- day0.iso：day0-config ファイルおよびオプションの idtoken ファイルを含む ISO。
- asav-vi.mf：vCenter への導入用のマニフェスト ファイル。
- asav-esxi.mf：vCenter 以外への導入用のマニフェスト ファイル。

ステップ 3 「day0-config」というテキストファイルに ASAv の CLI 設定を記入します。3 つのインターフェイスの設定とその他の必要な設定を追加します。

最初の行は ASA のバージョンで始める必要があります。day0-config は、有効な ASA 構成である必要があります。day0-config を生成する最適な方法は、既存の ASA または ASAv から実行コンフィギュレーションの必要な部分をコピーする方法です。day0-config 内の行の順序は重要で、既存の **show running-config** コマンド出力の順序と一致している必要があります。

day0-config ファイルの 2 つの例を示します。1 つ目の例では、ギガビット イーサネット インターフェイスを備えた ASAv を導入する場合の day0-config を示します。2 つ目の例では、10 ギガビット イーサネット インターフェイスを備えた ASAv を導入する場合の day0-config を示します。この day0-config を使用して、SR-IOV インターフェイスを備えた ASAv を導入します。注意事項と制約事項 （45 ページ）を参照してください。

例：

```
ASA Version 9.4.1
!
console serial
interface management0/0
nameif management
security-level 100
ip address 192.168.1.1 255.255.255.0
no shutdown
interface gigabitethernet0/0
nameif inside
security-level 100
ip address 10.1.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/1
nameif outside
security-level 0
ip address 198.51.100.2 255.255.255.0
no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
```

```
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
call-home
http-proxy 10.1.1.1 port 443
license smart
feature tier standard
throughput level 2G
```

例：

```
ASA Version 9.8.1
!
console serial
interface management 0/0
management-only
nameif management
security-level 0
ip address 192.168.0.230 255.255.255.0
!
interface GigabitEthernet0/0
nameif inside
security-level 100
ip address 10.10.10.10 255.255.255.0
!
interface GigabitEthernet0/1
nameif outside
security-level 0
ip address 10.10.20.10 255.255.255.0
!
route management 0.0.0.0 0.0.0.0 192.168.0.254
!
username cisco password cisco123 privilege 15
!
aaa authentication ssh console LOCAL
ssh 0.0.0.0 0.0.0.0 management
ssh timeout 60
ssh version 2
!
http 0.0.0.0 0.0.0.0 management
!
logging enable
logging timestamp
logging buffer-size 99999
logging buffered debugging
logging trap debugging
!
dns domain-lookup management
DNS server-group DefaultDNS
name-server 64.102.6.247
!
license smart
feature tier standard
throughput level 10G
!
crypto key generate rsa modulus 2048
```

ステップ 4 （任意）Cisco Smart Software Manager により発行された Smart License ID トークン ファイルをコンピュータにダウンロードします。

ステップ 5 （任意）ダウンロード ファイルから ID トークンをコピーし、ID トークンのみを含む「idtoken」というテキスト ファイルに保存します。

この ID トークンによって、Smart Licensing サーバーに ASAv が自動的に登録されます。

ステップ 6 テキスト ファイルを ISO ファイルに変換して仮想CD-ROM を生成します。

例：

```
stack@user-ubuntu:-/KvmAsa$ sudo genisoimage -r -o day0.iso day0-config idtoken
I: input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 252
Total directory bytes: 0
Path table size (byptes): 10
Max brk space used 0
176 extents written (0 MB)
stack@user-ubuntu:-/KvmAsa$
```

ステップ 7 day0.iso 用に Linux で新しい SHA1 値を計算します。

例：

```
openssl dgst -sha1 day0.iso
SHA1(day0.iso)= e5bee36e1eb1a2b109311c59e2f1ec9f731ecb66 day0.iso
```

ステップ 8 新しいチェックサムを作業ディレクトリの asav-vi.mf ファイルに含め、day0.iso SHA1 値を新しく生成された値で置き換えます。

例：

```
SHA1(asav-vi.ovf)= de0f1878b8f1260e379ef853db4e790c8e92f2b2
SHA1(disk0.vmdk)= 898b26891cc68fa0c94ebd91532fc450da418b02
SHA1(boot.vmdk)= 6b0000ddebfc38ccc99ac2d4d5dbfb8abfb3d9c4
SHA1(day0.iso)= e5bee36e1eb1a2b109311c59e2f1ec9f731ecb66
```

ステップ 9 ZIP ファイルを解凍したディレクトリに day0.iso ファイルをコピーします。デフォルト（空）の day0.iso ファイルが上書きされます。

このディレクトリから VM が導入される場合は、新しく生成された day0.iso 内の構成が適用されます。

VMware vSphere Web Client を使用した ASAv の導入

この項では、VMware vSphere Web Client を使用して ASAv を導入する方法について説明します。Web クライアントには、vCenter が必要です。vCenter がない場合は、「VMware vSphere スタンドアロンクライアントおよび第 0 日用構成を使用した ASAv の導入」、または「OVF ツールおよび第 0 日用構成を使用した ASAv の導入」を参照してください。

- vSphere Web Client へのアクセスとクライアント統合プラグインのインストール （29 ページ）
- VMware vSphere Web Client を使用した ASAv の導入 （28 ページ）

vSphere Web Client へのアクセスとクライアント統合プラグインのインストール

この項では、vSphere Web Client にアクセスする方法について説明します。また、ASAv コンソールアクセスに必要なクライアント統合プラグインをインストールする方法についても説明します。一部の Web クライアント機能（プラグインなど）は、Macintosh ではサポートされていません。完全なクライアントのサポート情報については、VMware の Web サイトを参照してください。

ステップ 1 ブラウザから VMware vSphere Web Client を起動します。

https://*vCenter_server:port***/vsphere-client/**

デフォルトでは、port は 9443 です。

ステップ 2 （1 回のみ）ASAv コンソールへのアクセスを可能にするため、クライアント統合プラグインをインストールします。

1. ログイン画面で、[Download the Client Integration Plug-in] をクリックしてプラグインをダウンロードします。
2. ブラウザを閉じてから、インストーラを使用してプラグインをインストールします。
3. プラグインをインストールしたら、vSphere Web Client に再接続します。

ステップ 3 ユーザー名とパスワードを入力し、[Login] をクリックするか、[Use Windows session authentication] チェックボックスをオンにします（Windows のみ）。

VMware vSphere Web Client を使用した ASAv の導入

ASAv を導入するには、VMware vSphere Web Client（または vSphere Client）、およびオープン仮想化フォーマット（OVF）のテンプレートファイルを使用します。シスコの ASAv パッケージを展開するには、vSphere Web Client で Deploy OVF Template ウィザードを使用します。このウィザードでは、ASAv OVA ファイルを解析し、ASAv を実行する仮想マシンを作成し、パッケージをインストールします。

ウィザードの手順のほとんどは、VMware に対し標準のものです。Deploy OVF Template の詳細については、VMware vSphere Web Client のオンライン ヘルプを参照してください。

始める前に

ASAv を導入する前に、vSphere（管理用）で少なくとも 1 つのネットワークを設定しておく必要があります。

ステップ 1 ASAv ZIP ファイルを Cisco.com からダウンロードし、PC に保存します。

http://www.cisco.com/go/asa-software

（注） Cisco.com のログインおよびシスコ サービス契約が必要です。

ステップ 2 vSphere Web Client の [Navigator] ペインで、[vCenter] をクリックします。

ステップ 3 [Hosts and Clusters] をクリックします。

ステップ 4 ASAv を導入するデータセンター、クラスタ、またはホストを右クリックして、[Deploy OVF Template] を選択します。
[Deploy OVF Template] ウィザードが表示されます。

ステップ 5 ウィザード画面の指示に従って進みます。

ステップ 6 [Setup networks] 画面で、使用する各 ASAv インターフェイスにネットワークをマッピングします。

ネットワークはアルファベット順になっていない可能性があります。ネットワークを見つけることが非常に困難な場合は、[設定の編集（Edit Settings）] ダイアログボックスからネットワークを後で変更できます。展開後、ASAv インスタンスを右クリックし、[設定の編集（Edit Settings）] を選択して、[設定の編集（Edit Settings）] ダイアログボックスにアクセスします。ただし、この画面には ASAv インターフェイス ID は表示されません（ネットワーク アダプタ ID のみ）。次のネットワーク アダプタ ID と ASAv インターフェイス ID の対応一覧を参照してください。

ネットワーク アダプタ ID	ASAv インターフェイス ID
ネットワーク アダプタ 1	Management 0/0
ネットワーク アダプタ 2	GigabitEthernet 0/0
ネットワーク アダプタ 3	GigabitEthernet 0/1
ネットワーク アダプタ 4	GigabitEthernet 0/2
ネットワーク アダプタ 5	GigabitEthernet 0/3
ネットワーク アダプタ 6	GigabitEthernet 0/4
ネットワーク アダプタ 7	GigabitEthernet 0/5
ネットワーク アダプタ 8	GigabitEthernet 0/6
ネットワーク アダプタ 9	GigabitEthernet 0/7
ネットワーク アダプタ 10	GigabitEthernet 0/8

すべての ASAv インターフェイスを使用する必要はありません。ただし、vSphere Web Client ではすべてのインターフェイスにネットワークを割り当てる必要があります。使用しないインターフェイスについては、ASAv 設定内でインターフェイスを無効のままにしておくことができます。ASAv を導入した後、任意で vSphere Web Client に戻り、[Edit Settings] ダイアログボックスから余分なインターフェイスを削除することができます。詳細については、vSphere Web Client のオンライン ヘルプを参照してください。

（注） フェールオーバー/HA 配置では、GigabitEthernet 0/8 がフェールオーバー インターフェイスとして事前設定されます。

ステップ 7 インターネット アクセスに HTTP プロキシを使用する場合は、[Smart Call Home Settings] 領域でスマート ライセンスのプロキシ アドレスを設定する必要があります。このプロキシは、一般に Smart Call Home にも使用されます。

ステップ 8 フェールオーバー/HA 配置では、[Customize] テンプレート画面で次を設定します。

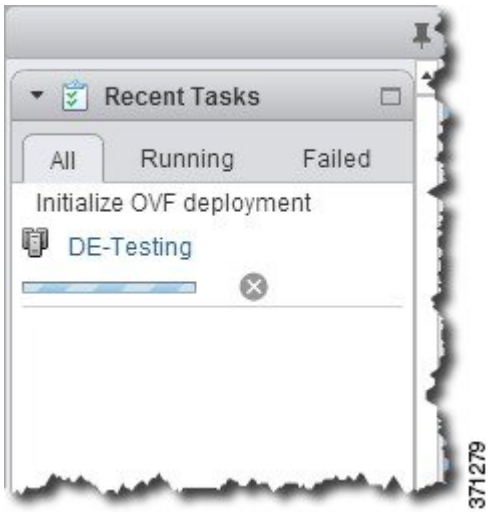
- スタンバイ管理 IP アドレスを指定します。

  インターフェイスを設定する場合、同じネットワーク上のアクティブ IP アドレスとスタンバイ IP アドレスを指定する必要があります。プライマリ装置が故障すると、セカンダリ装置はプライマリ装置の IP アドレスと MAC アドレスを引き継ぎ、トラフィックを通過させます。現在スタンバイになっている装置が、スタンバイの IP アドレスと MAC アドレスを引き継ぎます。ネットワーク デバイスは、MAC と IP アドレスの組み合わせについて変更を認識しないため、ネットワーク上のどのような場所でも ARP エントリが変更されたり、タイムアウトが生じたりすることはありません。

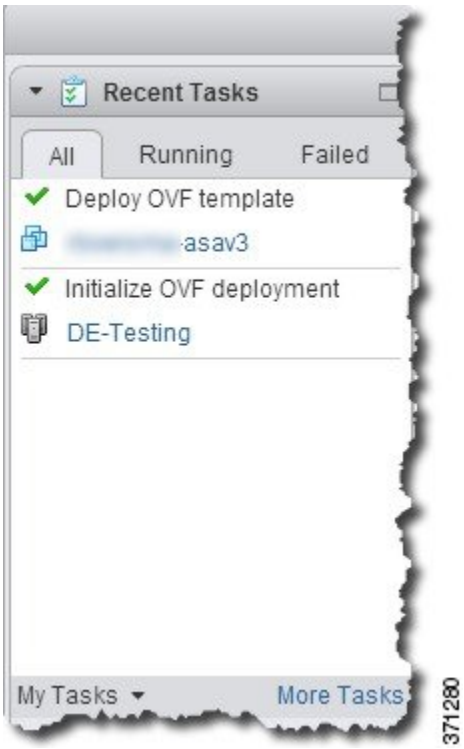
- [HA Connection Settings] 領域で、フェールオーバー リンクを設定します。

  フェールオーバー ペアの 2 台の装置は、フェールオーバー リンク経由で常に通信して、各装置の動作ステータスを確認しています。GigabitEthernet 0/8 がフェールオーバー リンクとして事前設定されています。同じネットワーク上のリンクに対するアクティブな IP アドレスとスタンバイの IP アドレスを入力します。

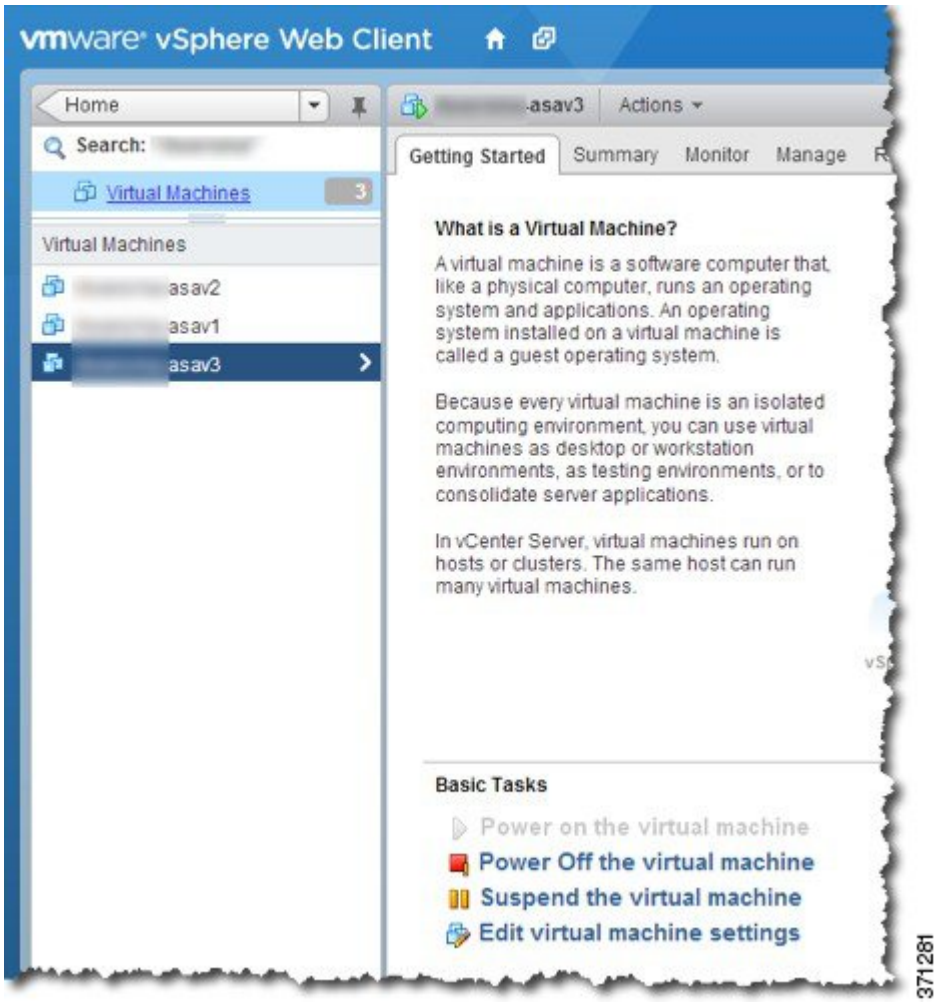
ステップ 9 ウィザードが完了すると、vSphere Web Client は VM を処理します。[グローバル情報（Global Information）] 領域の [最近のタスク（Recent Tasks）] ペインで [OVF展開の初期設定（Initialize OVF deployment）] ステータスを確認できます。



この手順が終了すると、[OVFテンプレートの導入（Deploy OVF Template）] 完了ステータスが表示されます。



その後、ASAv インスタンスがインベントリ内の指定されたデータセンターの下に表示されます。



ステップ 10 ASAv マシンがまだ稼働していない場合は、[仮想マシンの電源をオン（Power on the virtual machine）] をクリックします。

ASDM で接続を試行したりコンソールに接続を試行する前に、ASAv が起動するのを待ちます。ASAv が初めて起動すると、OVF ファイルから提供されたパラメータを読み込み、それらを ASAv システム構成に追加します。その後、起動プロセスが自動的に再開され、稼働を開始します。この二重起動プロセスは、初めて ASAv を導入した場合にのみ発生します。起動メッセージを確認するには、[Console] タブをクリックして、ASAv コンソールにアクセスします。

ステップ 11 フェールオーバー/HA 配置の場合は、この手順を繰り返してセカンダリ装置を追加します。次のガイドラインを参照してください。

- プライマリ装置と同じスループット レベルを設定します。
- プライマリ装置とまったく同じ IP アドレス設定を入力します。両方の装置のブートストラップ設定は、プライマリまたはセカンダリとして装置を識別するパラメータを除いて同一にします。

次のタスク

Cisco Licensing Authority に ASAv を正常に登録するには、ASAv にインターネットアクセスが必要です。インターネットに接続してライセンス登録を完了させるには、導入後に追加の設定が必要になることがあります。

VMware vSphere スタンドアロンクライアントおよび第 0 日用構成を使用した ASAv の導入

ASAv を導入するには、VMware vSphere Client およびオープン仮想化フォーマット（OVF）のテンプレートファイル（vCenter へ導入する場合は asav-vi.ovf、vCenter 以外へ導入する場合は asav-esxi.ovf）を使用します。シスコの ASAv パッケージを導入するには、vSphere Client で [OVFテンプレートの導入（Deploy OVF Template）] ウィザードを使用します。このウィザードでは、ASAv OVA ファイルを解析し、ASAv を実行する仮想マシンを作成し、パッケージをインストールします。

ウィザードの手順のほとんどは、VMware に対し標準のものです。[Deploy OVF Template] ウィザードの詳細については、VMware vSphere クライアントのオンライン ヘルプを参照してください。

始める前に

- ASAv を導入する前に、vSphere（管理用）で少なくとも 1 つのネットワークを設定しておく必要があります。
- ASAv ソフトウェアの解凍と第 0 日用構成ファイルの作成（25 ページ）の手順に従って、第 0 日用構成を作成します。

ステップ 1 VMware vSphere クライアントを起動し、**[File]** > **[Deploy OVF Template]** を選択します。

[Deploy OVF Template] ウィザードが表示されます。

ステップ 2 asav-vi.ovf ファイルを解凍した作業ディレクトリを参照し、それを選択します。

ステップ 3 [OVF Template Details] 画面が表示されます。次の画面に移動します。カスタムの第 0 日用コンフィギュレーション ファイルを使用する場合は、構成を変更する必要はありません。

ステップ 4 最後の画面に導入設定の要約が表示されます。[Finish] をクリックしてVM を導入します。

ステップ 5 ASAv の電源を投入し、VMware コンソールを開いて、2 回目の起動を待機します。

ステップ 6 ASAv に SSH 接続し、必要な構成を完了します。第 0 日用コンフィギュレーション ファイルに必要なすべての構成がされていない場合は、VMware コンソールを開いて、必要な構成を完了します。

これで、ASAv は完全に動作可能な状態です。

OVF ツールおよび第 0 日用構成を使用した ASAv の導入

このセクションでは、第 0 日用構成ファイルを必要とする OVF ツールを使用した ASAv の導入方法について説明します。

始める前に

- OVF ツールを使用して ASAv を導入する場合は、day0.iso ファイルが必要です。ZIP ファイルで提供されるデフォルトの空の day0.iso ファイルを使用するか、または、生成しカスタマイズした第 0 日用コンフィギュレーション ファイルを使用できます。第 0 日用コンフィギュレーション ファイルの作成方法については、ASAv ソフトウェアの解凍と第 0 日用構成ファイルの作成 （25 ページ）を参照してください。
- OVF ツールが Linux または Windows PC にインストールされ、ターゲット ESXi サーバーに接続できることを確認します。

ステップ 1 OVF ツールがインストールされていることを確認します。

例：

```
linuxprompt# which ovftool
```

ステップ 2 必要な導入オプションを指定した .cmd ファイルを作成します。

例：

```
linuxprompt# cat launch.cmd
ovftool \
--name="asav-941-demo" \
--powerOn \
--deploymentOption=4Core8GB \
--diskMode=thin \
--datastore=datastore1 \
--acceptAllEulas \
--net:Management0-0="Portgroup_Mgmt" \
--net:GigabitEthernet0-1="Portgroup_Inside" \
--net:GigabitEthernet0-0="Portgroup_Outside" \
--prop:HARole=Standalone \
asav-esxi.ovf \
vi://root@10.1.2.3/
```

ステップ 3 cmd ファイルを実行します。

例：

```
linuxprompt# ./launch.cmd
```

ASAv の電源を投入し、2 回目の起動を待機します。

ステップ 4 ASAv に SSH 接続し、必要に応じて設定を完了します。さらに設定が必要な場合は、ASAv に対して VMware コンソールを開き、必要な設定を適用します。

これで、ASAv は完全に動作可能な状態です。

ASAv コンソールへのアクセス

ASDM を使用する場合、トラブルシューティングに CLI を使用する必要がある場合があります。デフォルトでは、組み込みの VMware vSphere コンソールにアクセスできます。または、コピー アンド ペーストなどのより優れた機能を持つネットワーク シリアル コンソールを設定できます。

- VMware vSphere コンソールの使用
- ネットワーク シリアル コンソール ポートの設定

（注） 第 0 日用構成ファイルを使用して ASAv を導入する場合、構成ファイルに**コンソールシリアル**の設定を追加して、初回ブート時に仮想 VGA コンソールではなくシリアルポートを使用できます。ASAv ソフトウェアの解凍と第 0 日用構成ファイルの作成（25 ページ）を参照してください。

VMware vSphere コンソールの使用

初期設定またはトラブルシューティングを行うには、VMware vSphere Web Client により提供される仮想コンソールから CLI にアクセスします。後で Telnet または SSH の CLI リモート アクセスを設定できます。

始める前に

vSphere Web Client では、ASAv コンソール アクセスに必要なクライアント統合プラグインをインストールします。

ステップ 1 VMware vSphere Web Client で、インベントリの ASAv インスタンスを右クリックし、[Open Console] を選択します。または、[Summary] タブの [Launch Console] をクリックします。

ステップ 2 コンソールでクリックして Enter を押します。注：Ctrl + Alt を押すと、カーソルが解放されます。

ASAv がまだ起動中の場合は、起動メッセージが表示されます。

ASAv が初めて起動すると、OVF ファイルから提供されたパラメータを読み込み、それらを ASAv システム構成に追加します。その後、起動プロセスが自動的に再開され、稼働を開始します。この二重起動プロセスは、初めて ASAv を導入した場合にのみ発生します。

（注） ライセンスをインストールするまで、スループットは 100 Kbps に制限されるため、予備接続テストを実行できます。ライセンスは、通常の操作に必要です。ライセンスをインストールするまで、次のメッセージがコンソールで繰り返し表示されます。

```
Warning: ASAv platform license state is Unlicensed.
Install ASAv platform license for full functionality.
```

次のプロンプトが表示されます。

```
ciscoasa>
```

このプロンプトは、ユーザー EXEC モードで作業していることを示します。ユーザー EXEC モードでは、基本コマンドのみを使用できます。

ステップ 3 特権 EXEC モードにアクセスします。

例：

```
ciscoasa> enable
```

次のプロンプトが表示されます。

```
Password:
```

ステップ 4 Enter キーを押して、次に進みます。デフォルトでは、パスワードは空白です。以前にイネーブル パスワードを設定した場合は、Enter を押す代わりにこれを入力します。

プロンプトが次のように変化します。

```
ciscoasa#
```

設定以外のすべてのコマンドは、特権 EXEC モードで使用できます。特権 EXEC モードからコンフィギュレーション モードに入ることもできます。

特権モードを終了するには、**disable** コマンド、**exit** コマンド、または **quit** コマンドを入力します。

ステップ 5 グローバル コンフィギュレーション モードにアクセスします。

```
ciscoasa# configure terminal
```

プロンプトが次のように変化します。

```
ciscoasa(config)#
```

グローバル コンフィギュレーション モードから ASAv の設定を開始できます。グローバル コンフィギュレーション モードを終了するには、**exit** コマンド、**quit** コマンド、または **end** コマンドを入力します。

ネットワーク シリアル コンソール ポートの設定

コンソール エクスペリエンスの向上のために、コンソール アクセスについて、ネットワーク シリアル ポートを単独で設定するか、または仮想シリアル ポート コンセントレータ（vSPC）に接続するように設定できます。各方法の詳細については、VMware vSphere のマニュアルを参照してください。ASAv では、仮想コンソールの代わりにシリアル ポートにコンソール出力を送信する必要があります。この手順では、シリアル ポート コンソールを有効にする方法について説明します。

ステップ 1 VMware vSphere でネットワーク シリアル ポートを設定します。VMware vSphere のマニュアルを参照してください。

ステップ 2 ASAv で、「use_ttyS0」という名前のファイルを disk0 のルートディレクトリに作成します。このファイルには内容が含まれている必要はありません。この場所に存在することのみが必要です。

disk0:/use_ttyS0

- ASDM から **[ツール（Tools）]** > **[ファイル管理（File Management）]** ダイアログボックスを使用して、この名前で空のテキストファイルをアップロードできます。
- vSphere コンソールで、ファイル システム内の既存のファイル（任意のファイル）を新しい名前にコピーできます。次に例を示します。

```
ciscoasa(config)# cd coredumpinfo
ciscoasa(config)# copy coredump.cfg disk0:/use_ttyS0
```

ステップ 3 ASAvをリロードします。

- ASDM から **[Tools]** > **[System Reload]** を選択します。
- vSphere コンソールで **reload** を入力します。

  ASAv は vSphere コンソールへの送信を停止し、代わりにシリアル コンソールに送信します。

ステップ 4 シリアル ポートの追加時に指定した vSphere のホスト IP アドレスとポート番号に Telnet 接続するか、または vSPC の IP アドレスとポートに Telnet 接続します。

vCPU またはスループット ライセンスのアップグレード

ASAv は、使用できる vCPU の数に影響するスループット ライセンスを使用します。

ASAv の vCPU の数を増やす（または減らす）場合は、新しいライセンスを要求してその新しいライセンスを適用し、新しい値と一致するように VMware の VM プロパティを変更します。

（注） 割り当てられた vCPU は、ASAv CPU ライセンスまたはスループットライセンスと一致している必要があります。RAM は、vCPU 用に正しくサイズ調整されている必要があります。アップグレードまたはダウングレード時には、この手順に従って、ライセンスと vCPU を迅速に調整するようにします。永続的な不一致がある場合、ASAv は適切に動作しません。

ステップ 1 新しいライセンスを要求します。

ステップ 2 新しいライセンスを適用します。フェールオーバー ペアの場合、両方の装置に新しいライセンスを適用します。

ステップ 3 フェールオーバーを使用するかどうかに応じて、次のいずれかを実行します。

- フェールオーバーあり：vSphere Web Client で、スタンバイ ASAv の電源を切断します。たとえば、ASAv をクリックしてから [仮想マシンの電源をオフ（Power Off the virtual machine）] をクリックするか、または ASAv を右クリックして [ゲストOSをシャットダウン（Shut Down Guest OS）] を選択します。
- フェールオーバーなし：vSphere Web クライアントで、ASAv の電源を切断します。たとえば、ASAv をクリックしてから [仮想マシンの電源をオフ（Power Off the virtual machine）] をクリックするか、または ASAv を右クリックして [ゲストOSをシャットダウン（Shut Down Guest OS）] を選択します。

ステップ 4 ASAv をクリックしてから [仮想マシンの設定の編集（Edit Virtual machine settings）] をクリックします（または ASAv を右クリックして [設定の編集（Edit Settings）] を選択します）。

[Edit Settings] ダイアログボックスが表示されます。

ステップ 5 新しい vCPU ライセンスの正しい値を確認するには、ASAv のライセンス （1 ページ）にある CPU 要件とメモリ要件を参照してください。

ステップ 6 [Virtual Hardware] タブの [CPU] で、ドロップダウン リストから新しい値を選択します。

ステップ 7 [Memory] には、新しい RAM の値を入力します。

ステップ 8 [OK] をクリックします。

ステップ 9 ASAvの電源を入れます。たとえば、[Power On the Virtual Machine] をクリックします。

ステップ 10 フェールオーバー ペアの場合：

1. アクティブ装置へのコンソールを開くか、またはアクティブ装置で ASDM を起動します。
2. スタンバイ装置の起動が終了した後、スタンバイ装置にフェールオーバーします。
   - ASDM：**[Monitoring]** > **[Properties]** > **[Failover]** > **[Status]** を選択し、[Make Standby] をクリックします。
   - CLI： **failover active**
3. アクティブ装置に対して、ステップ 3 ～ 9 を繰り返します。

次のタスク

詳細については、ASAv のライセンス （1 ページ）を参照してください。

VMware での ASAv のパフォーマンス調整

ESXi 構成でのパフォーマンスの向上

ESXi ホストの CPU 構成時の設定を調整することによって、ESXi 環境内の ASAv のパフォーマンスを向上させることができます。[Scheduling Affinity] オプションによって、仮想マシンの CPU をホストの物理コア（およびハイパースレッディングが有効になっている場合のハイパースレッド）にどのように分散させるかを制御できます。この機能を使用すれば、各仮想マシンを、指定したアフィニティ セット内のプロセッサに割り当てることができます。

詳細については、以下の VMware ドキュメントを参照してください。

- 「*Administering CPU Resources*」の章（『vSphere Resource Management』）。
- 『Performance Best Practices for VMware vSphere』
- vSphere Client のオンライン ヘルプ。

NUMA のガイドライン

Non-uniform Memory Access（NUMA）は、マルチプロセッサシステムのプロセッサに対するメインメモリモジュールの配置について記述する共有メモリアーキテクチャです。プロセッサが自身のノード（リモートメモリ）内に存在しないメモリにアクセスする場合は、ローカルメモリにアクセスする場合よりも低速の速度で、NUMA 接続を介してデータを転送する必要があります。

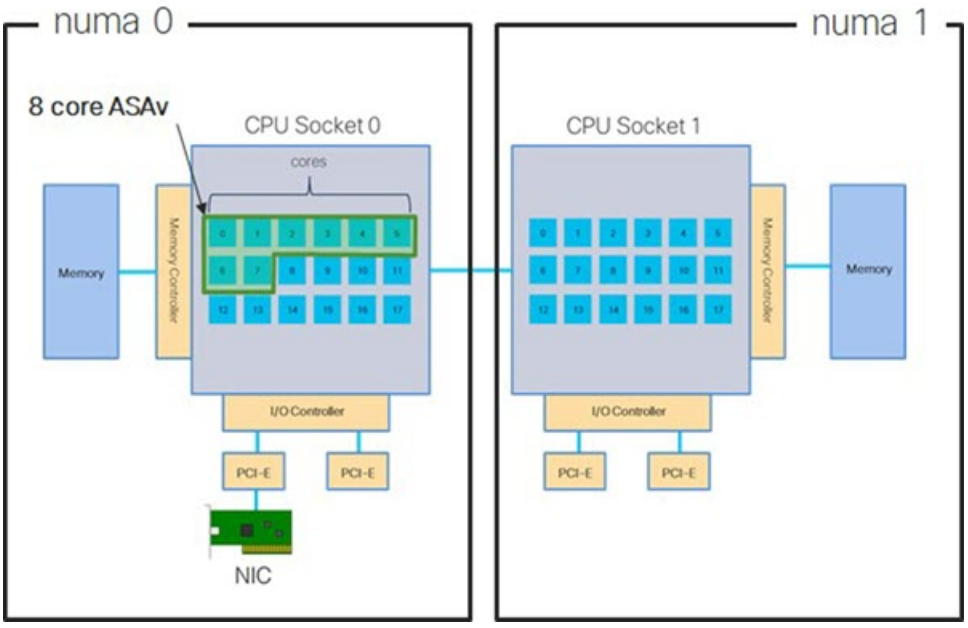
X86 サーバーアーキテクチャは、複数のソケットおよびソケット内の複数のコアで構成されています。各 CPU ソケットとそのメモリおよび I/O が、NUMA ノードと呼ばれます。メモリからパケットを効率的に読み取るには、ゲストアプリケーションおよび関連付けられている周辺機器（NIC など）が同じノード内に存在する必要があります。

最適な ASAv パフォーマンスを実現するには：

- ASAv マシンは、1つの NUMA ノード上で実行する必要があります。1つの ASAv が 2 つのソケットで実行されるように導入されている場合、パフォーマンスは大幅に低下します。
- 8 コア ASAv（図 1 : 8 コア NUMA アーキテクチャの例 （41 ページ））では、ホスト CPU の各ソケットが、それぞれ 8 個以上のコアを備えている必要があります。サーバー上で実行されている他の VM についても考慮する必要があります。
- 16 コア ASAv（図 2 : 16 コア ASAv NUMA アーキテクチャの例 （41 ページ））では、ホスト CPU 上の各ソケットが、それぞれ 16 個以上のコアを備えている必要があります。サーバー上で実行されている他の VM についても考慮する必要があります。
- NIC は、ASAv マシンと同じ NUMA ノード上にある必要があります。

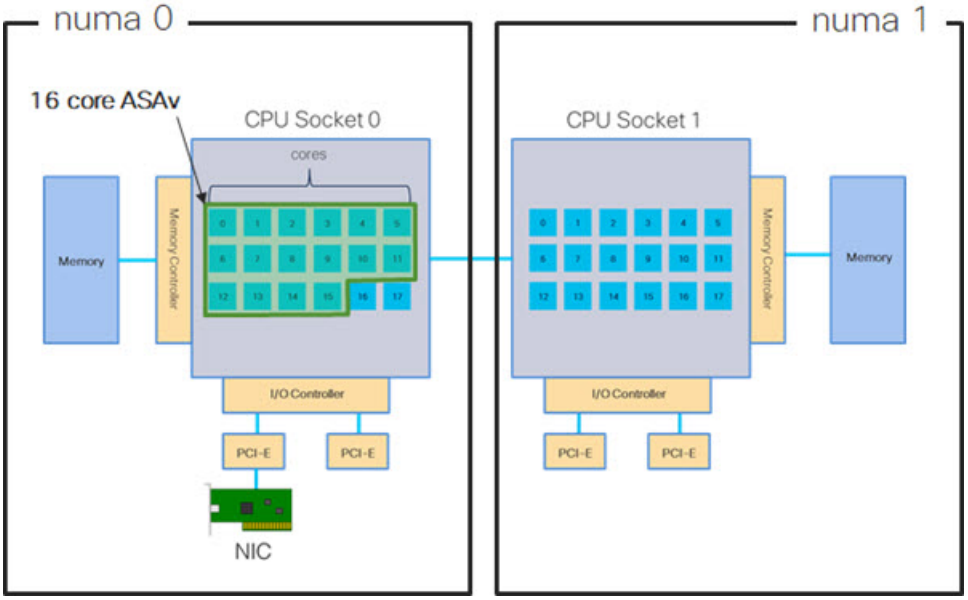
次の図は、2 つの CPU ソケットがあり、各 CPU に 18 個のコアが搭載されているサーバーを示しています。8 コア ASAv では、ホスト CPU の各ソケットに最低 8 個のコアが必要です。

図 *1 : 8* コア *NUMA* アーキテクチャの例



次の図は、2 つの CPU ソケットがあり、各 CPU に 18 個のコアが搭載されているサーバーを示しています。16 コア ASAv では、ホスト CPU の各ソケットに最低 16 個のコアが必要です。

図 *2 : 16* コア *ASAv NUMA* アーキテクチャの例



NUMA システムと ESXi の使用に関する詳細については、VMware ドキュメント『*vSphere Resource Management*』で、お使いの VMware ESXi バージョンを参照してください。このドキュメントおよびその他の関連ドキュメントの最新のエディションを確認するには、http://www.vmware.com/support/pubs を参照してください。

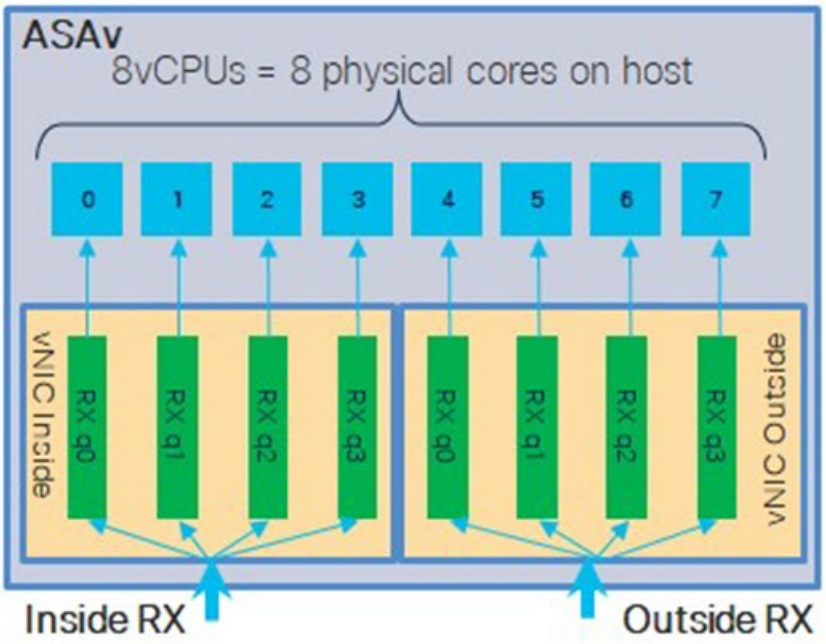
Receive Side Scaling（RSS）用の複数の RX キュー

ASAvは、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ネットワークアダプタによって使用されるテクノロジーである Receive Side Scaling（RSS）をサポートしています。最大スループットを実現するには、各 vCPU（コア）に独自の NIC RX キューが設定されている必要があります。一般的な RA VPN 展開では、1つの内部/外部ペアのインターフェイスを使用する場合があることに注意してください。

重要 複数の RX キューを使用するには、ASAv バージョン 9.13(1) 以降が必要です。

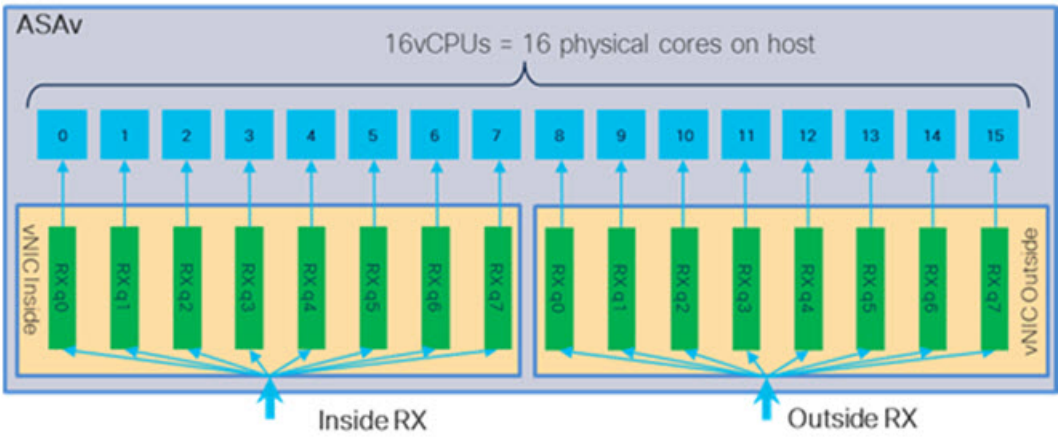
内部/外部ペアのインターフェイスを持つ 8 コア VM の場合、図 3 : 8 コア ASAv RSS RX キュー（42 ページ）に示すように、各インターフェイスには 4 つの RX キューがあります。

図 *3 : 8* コア *ASAv RSS RX* キュー



内部/外部ペアのインターフェイスを持つ 16 コア VM の場合、図 4 : 16 コア ASAv RSS RX キュー （43 ページ）に示すように、各インターフェイスには 8 つの RX キューがあり ます。

図 *4 : 16* コア *ASAv RSS RX* キュー



次の表に、VMware 用の ASAv の vNIC およびサポートされている RX キューの数を示します。サポートされている vNIC の説明については、推奨される vNIC （18 ページ）を参照してください。

表 *11 : VMware* で推奨される *NIC/vNIC*

NIC カード	vNIC ドライバ	ドライバテクノロジー	RX キューの数	パフォーマンス
x710*	i40e	PCI パススルー	最大 8	PCIパススルーは、テストされた NIC の中で最高のパフォーマンスを提供します。パススルーモードでは、NIC は ASAv 専用であり、仮想環境に最適な選択肢ではありません。
	i40evf	SR-IOV	4	X710 NIC を使用した SR-IOV のスループットは PCI パススルーよりも（最大 30%）低下します。VMware の i40evf には、i40evf ごとに最大 4 つの RX キューがあります。16 コア VM で最大スループットを実現するには、8 つの RX キューが必要です。
x520	ixgbe-vf	SR-IOV	2	—
	ixgbe	PCI パススルー	6	ixgbe ドライバ（PCI パススルーモード）には、6 つの RX キューがあります。パフォーマンスは i40evf（SR-IOV）と同等です。

NIC カード	vNIC ドライバ	ドライバテクノロジー	RX キューの数	パフォーマンス
該当なし	VMXNET3	準仮想化	最大 8	ASAv100 には推奨されません。
該当なし	e1000	VMware では推奨されません。		
*ASAv は、x710 NIC の 1.9.5 i40en ホストドライバと互換性がありません。これより古いバージョンまたは新しいバージョンのドライバは動作します。NIC ドライバとファームウェアのバージョンを識別または確認するための ESXCLI コマンドの詳細については、NIC ドライバとファームウェアバージョンの識別 （44 ページ）を参照してください。				

NIC ドライバとファームウェアバージョンの識別

特定のファームウェアおよびドライバのバージョン情報を識別または確認する必要がある場合は、ESXCLI コマンドを使用してそのデータを見つけることができます。

- インストールされている NIC のリストを取得するには、関連するホストに SSH 接続し、`esxcli network nic list` コマンドを実行します。このコマンドから、デバイスおよび一般情報の記録が得られるはずです。
- インストールされている NIC のリストを取得すれば、詳細な設定情報を得ることができます。必要な NIC の名前を指定して、`esxcli network nic get` コマンドを実行します：`esxcli network nic get -n <nic name>`。

（注） 一般的なネットワークアダプタ情報は、VMware vSphere クライアントから確認することもできます。アダプタとドライバは、[Configure] タブ内の [Physical Adapters] の下にあります。

SR-IOV インターフェイスのプロビジョニング

SR-IOV を使用すれば、複数の VM でホスト内部の 1 台の PCIe ネットワーク アダプタを共有することができます。SR-IOV は次の機能を定義しています。

- 物理機能（PF）：PF は、SR-IOV 機能を含むフル PCIe 機能です。これらは、ホスト サーバー上の通常のスタティック NIC として表示されます。
- 仮想機能（VF）：VF は、データ転送を支援する軽量 PCIe 機能です。VF は、PF から抽出され、PF を介して管理されます。

VF は、仮想化されたオペレーティング システム フレームワーク内の ASAv マシンに最大 10 Gbps の接続を提供できます。このセクションでは、KVM 環境で VF を設定する方法について説明します。ASAv 上の SR-IOV サポートについては、ASAv と SR-IOV インターフェイスのプロビジョニング （13 ページ）を参照してください。

注意事項と制約事項

SR-IOV インターフェイスに関するガイドライン

VMware vSphere 5.1 以降のリリースは、特定の設定の環境でしか SR-IOV をサポートしません。vSphere の一部の機能は、SR-IOV が有効になっていると機能しません。

SR-IOV インターフェイスに関するガイドラインと制限事項 （13 ページ） に記載されているASAv と SR-IOV に関するシステム要件に加えて、VMware と SR-IOV に関する要件、サポートされている NIC、機能の可用性、およびアップグレード要件の詳細については、VMware マニュアル内の『Supported Configurations for Using SR-IOV』で確認する必要があります。

このセクションでは、VMware システム上の SR-IOV インターフェイスのプロビジョニングに関するさまざまなセットアップ手順と設定手順を示します。このセクション内の情報は、VMware ESXi 6.0 と vSphere Web Client、Cisco UCS C シリーズ サーバー、および Intel Ethernet Server Adapter X520 - DA2 を使用した特定のラボ環境内のデバイスから作成されたものです。

SR-IOV インターフェイスに関する制限事項

ASAv を起動すると、ESXi で表示される順序とは逆の順序で、SR-IOV インターフェイスが表示される場合があります。これにより、インターフェイス設定エラーが発生し、特定の ASAv マシンへのネットワーク接続が切断する場合があります。

注意 ASAv で SR-IOV ネットワーク インターフェイスの設定を開始する前に、インターフェイスのマッピングを確認することが重要です。これにより、ネットワーク インターフェイスの設定が、VM ホストの正しい物理 MAC アドレスインターフェイスに適用されます。

ASAv が起動したら、MAC アドレスとインターフェイスのマッピングを確認できます。**show interface** コマンドを使用して、インターフェイスの MAC アドレスなど、インターフェイスの詳細情報を確認します。インターフェイス割り当てが正しいことを確認するには、**show kernel ifconfig** コマンドの結果と MAC アドレスを比較します。

ESXi ホスト BIOS の確認

VMware に SR-IOV インターフェイスを備えた ASAv を導入するには、仮想化をサポートして有効にする必要があります。VMware では、SR-IOV サポートに関するオンラインの『Compatibility Guide』だけでなく、仮想化が有効か無効かを検出するダウンロード可能な『CPU Identification Utility』も含めて、仮想化サポートの各種確認手段を提供しています。

また、ESXi ホストにログインすることによって、BIOS 内で仮想化が有効になっているかどうかを判断することもできます。

ステップ 1 次のいずれかの方法を使用して、ESXi シェルにログインします。

- ホストへの直接アクセスがある場合は、Alt+F2 を押して、マシンの物理コンソールのログインページを開きます。

- ホストにリモートで接続している場合は、SSH または別のリモート コンソール接続を使用して、ホスト上のセッションを開始します。

ステップ 2 ホストによって認識されるユーザ名とパスワードを入力します。

ステップ 3 次のコマンドを実行します。

例：

```
esxcfg-info|grep "\----\HV Support"
```

HV Support コマンドの出力は、使用可能なハイパーバイザサポートのタイプを示します。可能性のある値の説明を以下に示します。

0：VT/AMD-V は、サポートがこのハードウェアでは使用できないことを示します。

1：VT/AMD-V は、VT または AMD-V を使用できますが、このハードウェアではサポートされないことを示します。

2：VT/AMD-V は、VT または AMD-V を使用できますが、現在、BIOS 内で有効になっていないことを示します。

3：VT/AMD-V は、VT または AMD-V が BIOS 内で有効になっており、使用できることを示します。

例：

```
~ # esxcfg-info|grep "\----\HV Support"
         |----HV Support.........................3
```

値の 3 は、仮想化がサポートされており、有効になっていることを示します。

次のタスク

- ホスト物理アダプタ上で SR-IOV を有効にします。

ホスト物理アダプタ上での SR-IOV の有効化

vSphere Web Client を使用して、ホストで SR-IOV を有効にし、仮想機能の数を設定します。設定しないと、仮想マシンを仮想機能に接続できません。

始める前に

- SR-IOV 互換ネットワーク インターフェイス カード（NIC）がインストールされていることを確認します。「SR-IOV でサポートされている NIC （15 ページ）」を参照してください。

ステップ 1 vSphere Web Client で、SR-IOV を有効にする ESXi ホストに移動します。

ステップ 2 [Manage] タブで、[Networking] をクリックし、[Physical adapters] を選択します。

SR-IOV プロパティを調査することにより、物理アダプタが SR-IOV をサポートしているかどうかを確認できます。

ステップ 3 物理アダプタを選択し、[Edit adapter settings] をクリックします。

ステップ 4 SR-IOV の下で、[Status] ドロップダウン メニューから [Enabled] を選択します。

ステップ 5 [Number of virtual functions] テキストボックスに、アダプタに設定する仮想機能の数を入力します。

（注） ASAv50 では、インターフェイスあたり 2 つ以上の VF を使用**しない**ことをお勧めします。物理インターフェイスを複数の仮想機能で共有すると、パフォーマンスが低下する可能性があります。

ステップ 6 [OK] をクリックします。

ステップ 7 ESXi ホストを再起動します。

物理アダプタ エントリで表現された NIC ポートで仮想機能がアクティブになります。これらは、ホストの [Settings] タブの [PCI Devices] リストに表示されます。

次のタスク

- SR-IOV 機能と設定を管理するための標準 vSwitch を作成します。

vSphere スイッチの作成

SR-IOV インターフェイスを管理するための vSphere スイッチを作成します。

ステップ 1 vSphere Web Client で、ESXi ホストに移動します。

ステップ 2 [Manage] で、[Networking] を選択してから、[Virtual switches] を選択します。

ステップ 3 プラス（+）記号付きの緑色の地球アイコンである [Add host networking] アイコンをクリックします。

ステップ 4 [標準スイッチ用仮想マシンポートグループ（Virtual Machine Port Group for a Standard Switch）] 接続タイプを選択して、[次へ（Next）] をクリックします。

ステップ 5 [New standard switch] を選択して、[Next] をクリックします。

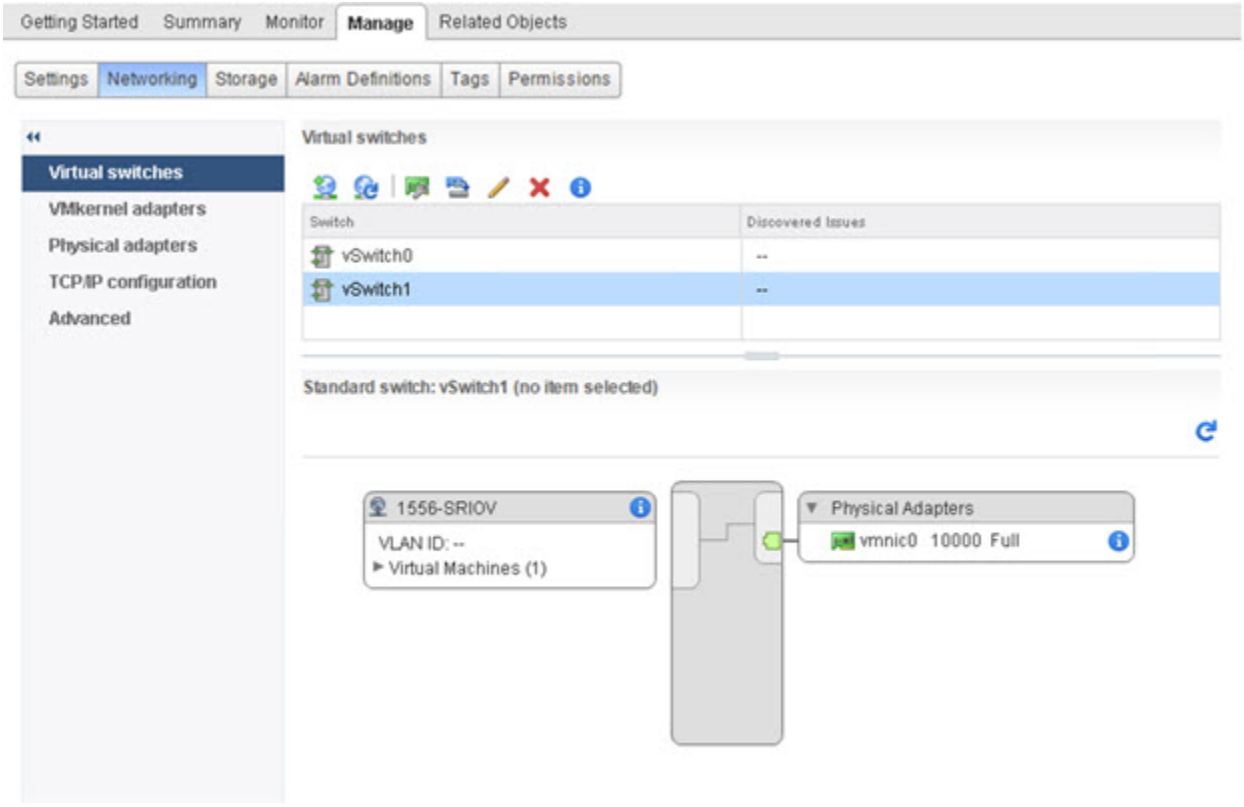
ステップ 6 物理ネットワーク アダプタを新しい標準スイッチに追加します。

a) 割り当てられたアダプタの下で、緑色のプラス（+）記号をクリックしてアダプタを追加します。
b) リストから SR-IOV に対応するネットワーク インターフェイスを選択します。たとえば、Intel(R) 82599 10 Gigabit Dual Port Network Connection を選択します。
c) [Failover order group] ドロップダウン メニューで、[Active adapters] から選択します。
d) [OK] をクリックします。

ステップ 7 SR-IOV vSwitch の [Network label] を入力して、[Next] をクリックします。

ステップ 8 [Ready to complete] ページで選択を確認してから、[Finish] をクリックします。

図 *5 : SR-IOV* インターフェイスがアタッチされた新しい *vSwitch*



次のタスク

- 仮想マシンの互換性レベルを確認します。

仮想マシンの互換性レベルのアップグレード

互換性レベルは、ホストマシンで使用可能な物理ハードウェアに対応する仮想マシンで使用可能な仮想ハードウェアを決定します。ASAvマシンは、ハードウェアレベルを10以上にする必要があります。これにより、SR-IOV のパススルー機能が ASAv に公開されます。この手順では、ASAv を短時間で最新のサポートされている仮想ハードウェアバージョンにアップグレードします。

仮想マシンのハードウェアバージョンと互換性については、vSphere 仮想マシン管理マニュアルを参照してください。

ステップ 1 vSphere Web Client から vCenter Server にログインします。

ステップ 2 変更する ASAv マシンを特定します。

a) データセンター、フォルダ、クラスタ、リソース プール、またはホストを選択して、[Related Objects] タブをクリックします。

b) [仮想マシン（Virtual Machines）] をクリックして、リストから ASAv マシンを選択します。

ステップ 3 選択した仮想マシンの電源をオフにします。

ステップ 4 ASAv を右クリックして、**[アクション（Actions）]**>**[すべてのvCenterアクション（All vCenter Actions）]**>**[互換性（Compatibility）]** > **[VMアップグレードの互換性（Upgrade VM Compatibility）]** を選択します。

ステップ 5 [はい（Yes）] をクリックして、アップグレードを確認します。

ステップ 6 仮想マシンの互換性で [ESXi 5.5以降（ESXi 5.5 and later）] オプションを選択します。

ステップ 7 （オプション）[通常のゲストOSのシャットダウン後にのみアップグレード（Only upgrade after normal guest OS shutdown）] を選択します。

選択された仮想マシンが、選択された [Compatibility] 設定の対応するハードウェア バージョンにアップグレードされ、仮想マシンの [Summary] タブで新しいハードウェア バージョンが更新されます。

次のタスク

- SR-IOV パススルー ネットワーク アダプタを介して ASAv と仮想機能を関連付けます。

ASAv への SR-IOV NIC の割り当て

ASAv マシンと物理 NIC がデータを交換可能なことを保証するには、ASAv を SR-IOV パススルー ネットワーク アダプタとして 1 つ以上の仮想機能に関連付ける必要があります。次の手順では、vSphere Web Client を使用して、SR-IOV NIC を ASAv マシンに割り当てる方法について説明します。

ステップ 1 vSphere Web Client から vCenter Server にログインします。

ステップ 2 変更する ASAv マシンを特定します。

a) データセンター、フォルダ、クラスタ、リソース プール、またはホストを選択して、[Related Objects] タブをクリックします。

b) [仮想マシン（Virtual Machines）] をクリックして、リストから ASAv マシンを選択します。

ステップ 3 仮想マシンの [Manage] タブで、**[Settings]** > **[VM Hardware]** を選択します。

ステップ 4 [Edit] をクリックして、[Virtual Hardware] タブを選択します。

ステップ 5 [New device] ドロップダウン メニューで、[Network] を選択して、[Add] をクリックします。

[New Network] インターフェイスが表示されます。

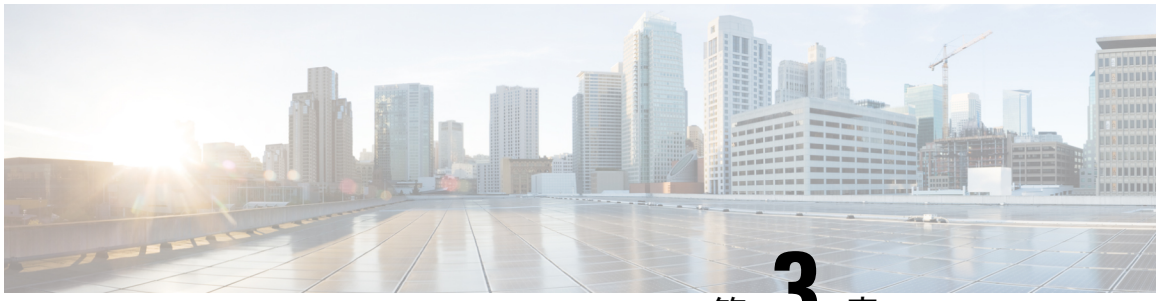
ステップ 6 [New Network] セクションを展開して、使用可能な SRIOV オプションを選択します。

ステップ 7 [Adapter Type] ドロップダウン メニューで、[SR-IOV passthrough] を選択します。

ステップ 8 [Physical function] ドロップダウン メニューで、パススルー仮想マシン アダプタに対応する物理アダプタを選択します。

ステップ 9 仮想マシンの電源をオンにします。

仮想マシンの電源をオンにすると、ESXi ホストが物理アダプタから空いている仮想機能を選択して、それを SR-IOV パススルーアダプタにマップします。ホストが仮想マシンアダプタと基礎となる仮想機能のすべてのプロパティを確認します。



第 3 章

KVM を使用した ASAv の導入

カーネルベースの仮想マシン（KVM）を実行できる任意のサーバークラスの x86 CPU デバイスに ASAv を導入できます。

重要 ASAv の最小メモリ要件は 2GB です。現在の ASAv が 2GB 未満のメモリで動作している場合、ASAv マシンのメモリを増やさないと、以前のバージョンから 9.13(1) 以降にアップグレードできません。また、最新バージョンを使用して新しい ASAv マシンを再導入できます。

KVM での ASAv のガイドラインで制限事項

ASAv の導入に使用される特定のハードウェアは、導入されるインスタンスの数や使用要件によって異なります。作成する各仮想アプライアンスには、ホストマシン上での最小リソース割り当て（メモリ、CPU 数、およびディスク容量）が必要です。

重要 ASAv は、8GB のディスクストレージサイズで導入されます。ディスク容量のリソース割り当てを変更することはできません。

ASAv を導入する前に、次のガイドラインと制限事項を確認します。

KVM での ASAv のシステム要件

最適なパフォーマンスを確保するために、以下の仕様に準拠していることを確認してください。ASAv には、次の要件があります。

- ホスト CPU は、仮想化拡張機能を備えたサーバークラスの x86 ベースの Intel または AMD CPU である必要があります。

  たとえば、ASAv パフォーマンステストラボでは、2.6GHz で動作する Intel® Xeon® CPU E5-2690v4 プロセッサを搭載した Cisco Unified Computing System™（Cisco UCS®）C シリーズ M4 サーバーを最低限使用しています。

推奨される vNIC

最適なパフォーマンスを得るためには、次の vNIC を推奨します。

- PCI パススルーでの i40e：サーバーの物理 NIC を VM に関連付け、DMA（ダイレクトメモリアクセス）を介して NIC と VM の間でパケットデータを転送します。パケットの移動に CPU サイクルは必要ありません。
- i40evf/ixgbe-vf：実質的に上記と同じですが（NIC と VM 間の DMA パケット）、NIC を複数の VM 間で共有できます。SR-IOV は、導入の柔軟性が高いため、一般的に推奨されます。参照先
- virtio：10Gbps の動作をサポートしますが、CPU サイクルも必要な準仮想化ネットワークドライバです。

（注） KVM システムで実行されている ASAv インスタンスでは、vNIC ドライバ i40e バージョン 2.11.25 を使用する SR-IOV インターフェイスでデータ接続の問題が発生する場合があります。この問題の回避策として、この vNIC バージョンを他のバージョンにアップグレードすることを推奨します。

パフォーマンスの最適化

ASAv 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、KVM での ASAv のパフォーマンス調整 （60 ページ）を参照してください。

- **NUMA**：ゲスト VM の CPU リソースを単一の Non-Uniform Memory Access（NUMA）ノードに分離することで、ASAv のパフォーマンスを向上できます。詳細については、NUMA のガイドライン （61 ページ）を参照してください。
- **Receive Side Scaling**：ASAv 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。詳細については、Receive Side Scaling（RSS）用の複数の RX キュー （64 ページ）を参照してください。

- **VPN の最適化**：ASAv で VPN パフォーマンスを最適化するための追加の考慮事項については、VPN の最適化 （66 ページ）を参照してください。

クラスタリング

バージョン 9.17 以降、クラスタリングは KVM で展開された ASA 仮想インスタンスでサポートされます。詳細については、「ASA Cluster for the ASAv」を参照してください。

CPU ピニング

KVM 環境で ASAv を機能させるには、CPU ピニングが必要です。CPU ピンニングの有効化（60 ページ）を参照してください。

ハイ アベイラビリティ ガイドラインのためのフェールオーバー

フェールオーバー配置の場合は、スタンバイ装置が同じライセンス権限付与を備えていることを確認してください（たとえば、両方の装置が 2Gbps の権限付与であることなど）。

重要 ASAv を使用して高可用性ペアを作成する場合は、データインターフェイスを各 ASAv に同じ順序で追加する必要があります。完全に同じインターフェイスが異なる順序で各 ASAv に追加されると、ASAv コンソールにエラーが表示されることがあります。また、フェールオーバー機能にも影響が出ることがあります。

Proxmox VE 上の ASAv

Proxmox Virtual Environment（VE）は、KVM 仮想マシンを管理できるオープンソースのサーバー仮想化プラットフォームです。Proxmox VE は、Web ベースの管理インターフェイスも提供します。

Proxmox VE に ASAv を導入する場合は、エミュレートされたシリアルポートを持つように VM を設定する必要があります。シリアルポートがないと、ブートアッププロセス中に ASAv がループ状態になります。すべての管理タスクは、Proxmox VE Web ベース管理インターフェイスを使用して実行できます。

（注） Unix シェルまたは Windows Powershell に慣れている上級ユーザー向けに、Proxmox VE は仮想環境のすべてのコンポーネントを管理するコマンド ライン インターフェイスを提供します。このコマンド ライン インターフェイスには、インテリジェントなタブ補完機能と UNIX の man ページ形式の完全なドキュメントがあります。

ASAv を正しく起動するには、VM にシリアルデバイスを設定する必要があります。

1. メイン Management Center の左側のナビゲーションツリーで ASAv マシンを選択します。
2. 仮想マシンの電源をオフにします。

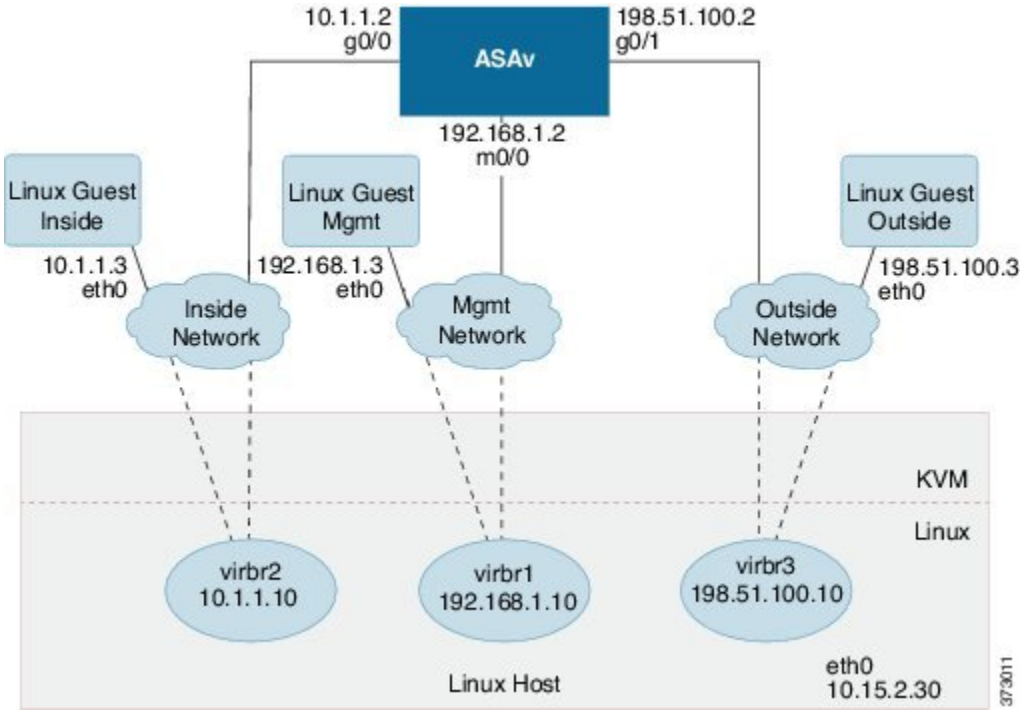
3. **Hardware** > **Add** > **Network Device** を選択して、シリアルポートを追加します。
4. 仮想マシンの電源をオンにします。
5. Xterm.js を使用して ASAv マシンにアクセスします。

ゲスト/サーバーで端末をセットアップしてアクティブ化する方法については、Proxmox シリアル端末のページを参照してください。

KVM を使用した ASAv の導入について

次の図は、ASAv と KVM のネットワークトポロジの例を示します。この章で説明している手順は、このトポロジの例に基づいています。ASAv は、内部ネットワークと外部ネットワークの間のファイアウォールとして動作します。また、別個の管理ネットワークが設定されます。

図 *6 : KVM* を使用した *ASAv* の導入例



ASAv と KVM の前提条件

- Cisco.com から ASAv qcow2 ファイルをダウンロードし、Linux ホストに格納します。

  http://www.cisco.com/go/asa-software

（注） Cisco.com のログインおよびシスコ サービス契約が必要です。

- このマニュアルの導入例では、ユーザーが Ubuntu 18.04 LTS を使用していることを前提としています。Ubuntu 18.04 LTS ホストの最上部に次のパッケージをインストールします。
  - qemu-kvm
  - libvirt bin
  - bridge-utils
  - Virt-Manager
  - virtinst
  - virsh tools
  - genisoimage
- パフォーマンスはホストとその設定の影響を受けます。ホストを調整することで、KVM での ASAv のスループットを最大化できます。一般的なホスト調整の概念については、『NFV Delivers Packet Processing Performance with Intel』を参照してください。
- Ubuntu 18.04 の便利な最適化には、次のものが含まれます。
  - macvtap：高性能の Linux ブリッジ。Linux ブリッジの代わりに macvtap を使用できます。ただし、Linux ブリッジの代わりに macvtap を使用する場合は、特定の設定を行う必要があります。
  - Transparent Huge Pages：メモリページサイズを増加させます。Ubuntu 18.04 では、デフォルトでオンになっています。

    Hyperthread disabled：2 つの vCPUを 1 つのシングル コアに削減します。
  - txqueuelength：デフォルトの txqueuelength を 4000 パケットに増加させ、ドロップ レートを低減します。
  - pinning：qemu および vhost プロセスを特定のCPU コア にピン接続します。特定の条件下では、ピン接続によってパフォーマンスが大幅に向上します。
- RHEL ベースのディストリビューションの最適化については、『Red Hat Enterprise Linux 7 Virtualization Tuning and Optimization Guide』を参照してください。
- ASA ソフトウェアおよび ASAv ハイパーバイザの互換性については、Cisco ASA の互換性 [英語] を参照してください。

第 0 日のコンフィギュレーション ファイルの準備

ASAv を起動する前に、第 0 日用のコンフィギュレーション ファイルを準備できます。このファイルは、ASAv の起動時に適用される ASAv の設定を含むテキストファイルです。この初期設定は、「day0-config」というテキスト ファイルとして指定の作業ディレクトリに格納され、さらに day0.iso ファイルへと処理されます。この day0.iso ファイルが最初の起動時にマウントされて読み取られます。第 0 日用コンフィギュレーション ファイルには、少なくとも、管理インターフェイスをアクティブ化するコマンドと、公開キー認証用 SSH サーバーを設定するコマンドを含める必要がありますが、すべての ASA 設定を含めることもできます。

day0.iso ファイル（カスタム day0.iso またはデフォルト day0.iso）は、最初の起動中に使用できる必要があります。

- 初期導入時に自動的に ASAv にライセンスを付与するには、Cisco Smart Software Manager からダウンロードした Smart Licensing Identity（ID）トークンを「idtoken」というテキスト ファイルに格納し、第 0 日用構成ファイルと同じディレクトリに保存します。
- 仮想 VGA コンソールではなく、ハイパーバイザの**シリアルポート**から ASAv にアクセスし、設定する場合は、第 0 日用構成ファイルにコンソールシリアルの設定を追加して初回ブート時にシリアルポートを使用する必要があります。
- トランスペアレントモードで ASAv を導入する場合は、トランスペアレントモードで実行される既知の ASA 構成ファイルを、第 0 日用構成ファイルとして使用する必要があります。これは、ルーテッド ファイアウォールの第 0 日用コンフィギュレーション ファイルには該当しません。

（注） この例では Linux が使用されていますが、Windows の場合にも同様のユーティリティがあります。

ステップ 1 「day0-config」というテキストファイルに ASAv の CLI 設定を記入します。3 つのインターフェイスの設定とその他の必要な設定を追加します。

最初の行は ASA のバージョンで始める必要があります。day0-config は、有効な ASA 構成である必要があります。day0-config を生成する最適な方法は、既存の ASA または ASAv から実行コンフィギュレーションの関連部分をコピーする方法です。day0-config 内の行の順序は重要で、既存の **show running-config** コマンド出力の順序と一致している必要があります。

例：

```
ASA Version 9.4.1
!
console serial
interface management0/0
nameif management
security-level 100
ip address 192.168.1.2 255.255.255.0
no shutdown
```

```
interface gigabitethernet0/0
nameif inside
security-level 100
ip address 10.1.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/1
nameif outside
security-level 0
ip address 198.51.100.2 255.255.255.0
no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
```

ステップ 2 （任意）ASAv の初期導入時に自動的にライセンスを許諾する場合は、day0-config ファイルに次の情報が含まれていることを確認してください。

- 管理インターフェイスの IP アドレス
- （任意）SSmart Licensing で使用する HTTP プロキシ
- HTTP プロキシ（指定した場合）または tools.cisco.com への接続を有効にする **route** コマンド
- tools.cisco.com を IP アドレスに解決する DNS サーバー
- 要求する ASAv ライセンスを指定するための Smart Licensing の設定
- （任意）CSSM での ASAv の検索を容易にするための一意のホスト名

ステップ 3 （任意）Cisco Smart Software Manager によって発行された Smart License ID トークンファイルをコンピュータにダウンロードし、ダウンロードファイルから ID トークンをコピーし、ID トークンのみを含む「idtoken」というテキストファイルを作成します。

ステップ 4 テキスト ファイルを ISO ファイルに変換して仮想CD-ROM を生成します。

例：

```
stack@user-ubuntu:-/KvmAsa$ sudo genisoimage -r -o day0.iso day0-config idtoken
I: input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 252
Total directory bytes: 0
Path table size (byptes): 10
Max brk space used 0
176 extents written (0 MB)
stack@user-ubuntu:-/KvmAsa$
```

この ID トークンによって、Smart Licensing サーバーに ASAv が自動的に登録されます。

ステップ 5 ステップ 1 から 5 を繰り返し、導入する ASAv ごとに、適切な IP アドレスを含むデフォルトの構成ファイルを作成します。

仮想ブリッジ XML ファイルの準備

ASAv ゲストを KVM ホストに接続し、ゲストを相互接続する仮想ネットワークを設定する必要があります。

（注） この手順では、KVM ホストから外部への接続は確立されません。

KVM ホスト上に仮想ブリッジ XML ファイルを準備します。第 0 日のコンフィギュレーションファイルの準備 （56 ページ） に記載されている仮想ネットワーク トポロジの例では、3 つの仮想ブリッジ ファイル（virbr1.xml、virbr2.xml、virbr3.xml）が必要です（これらの 3 つのファイル名を使用する必要があります。たとえば、virbr0 はすでに存在しているため使用できません）。各ファイルには、仮想ブリッジの設定に必要な情報が含まれています。仮想ブリッジに対して名前と一意の MAC アドレスを指定する必要があります。IP アドレスの指定は任意です。

ステップ 1 3 つの仮想ネットワーク ブリッジ XML ファイルを作成します。次の例では、virbr1.xml、virbr2.xml、および virbr3.xml です。

例：

```
<network>
<name>virbr1</name>
<bridge name='virbr1' stp='on' delay='0' />
<mac address='52:54:00:05:6e:00' />
<ip address='192.168.1.10' netmask='255.255.255.0' />
</network>
```

例：

```
<network>
<name>virbr2</name>
<bridge name='virbr2' stp='on' delay='0' />
<mac address='52:54:00:05:6e:01' />
<ip address='10.1.1.10' netmask='255.255.255.0' />
</network>
```

例：

```
<network>
<name>virbr3</name>
<bridge name='virbr3' stp='on' delay='0' />
<mac address='52:54:00:05:6e:02' />
<ip address='198.51.100.10' netmask='255.255.255.0' />
</network>
```

ステップ 2 以下を含むスクリプトを作成します（この例では、スクリプトに virt_network_setup.sh という名前を付けます）。

```
virsh net-create virbr1.xml
virsh net-create virbr2.xml
virsh net-create virbr3.xml
```

ステップ 3 このスクリプトを実行して、仮想ネットワークを設定します。このスクリプトは、仮想ネットワークを稼働状態にします。ネットワークは、KVM ホストが動作している限り稼動します。

```
stack@user-ubuntu:-/KvmAsa$ virt_network_setup.sh
```

（注） Linux ホストをリロードする場合は、virt_network_setup.sh スクリプトを再実行する必要があります。スクリプトはリブート後に継続されません。

ステップ 4 仮想ネットワークが作成されたことを確認します。

```
stack@user-ubuntu:-/KvmAsa$ brctl show
bridge name bridge id STP enabled Interfaces
virbr0 8000.0000000000000 yes
virbr1 8000.5254000056eed yes virb1-nic
virbr2 8000.5254000056eee yes virb2-nic
virbr3 8000.5254000056eec yes virb3-nic
stack@user-ubuntu:-/KvmAsa$
```

ステップ 5 virbr1 ブリッジに割り当てられている IP アドレスを表示します。これは、XML ファイルで割り当てた IP アドレスです。

```
stack@user-ubuntu:-/KvmAsa$ ip address show virbr1
S: virbr1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
link/ether 52:54:00:05:6e:00 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.10/24 brd 192.168.1.255 scope global virbr1
valid_lft forever preferred_lft forever
```

ASAv の起動

virt-install ベースの導入スクリプトを使用して ASAv を起動できます。

ステップ 1 「virt_install_asav.sh」という virt-install スクリプトを作成します。

ASAv マシンの名前は、この KVM ホスト上の他の全 VM で一意である必要があります。

ASAv では最大 10 のネットワークがサポートされます。この例では 3 つのネットワークが使用されています。ネットワーク ブリッジの句の順序は重要です。リストの最初の句は常に ASAv の管理インターフェイス（Management 0/0）、2 番目の句は ASAv の GigabitEthernet 0/0、3 番目の句は ASAv の GigabitEthernet 0/1 に該当し、GigabitEthernet 0/8 まで同様に続きます。仮想 NIC は Virtio でなければなりません。

例：

```
virt-install \
--connect=qemu:///system \
--network network=default,model=virtio \
--network network=default,model=virtio \
--network network=default,model=virtio \
--name=asav \
```

```
--cpu host \
--arch=x86_64 \
--machine=pc-1.0 \
--vcpus=1 \
--ram=2048 \
--os-type=linux \
--virt-type=kvm \
--import \
--disk path=/home/kvmperf/Images/desmo.qcow2,format=qcow2,device=disk,bus=virtio,cache=none \
--disk path=/home/kvmperf/asav_day0.iso,format=iso,device=cdrom \
--console pty,target_type=virtio \
--serial tcp,host=127.0.0.1:4554,mode=bind,protocol=telnet
```

ステップ 2 virt_install スクリプトを実行します。

例：

```
stack@user-ubuntu:-/KvmAsa$ ./virt_install_asav.sh

Starting install...
Creating domain...
```

ウィンドウが開き、VMのコンソールが表示されます。VMが起動中であることを確認できます。VMが起動するまでに数分かかります。VM が起動したら、コンソール画面から CLI コマンドを実行できます。

KVM での ASAv のパフォーマンス調整

KVM 構成でのパフォーマンスの向上

KVMホストの設定を変更することによって、KVM環境内のASAvのパフォーマンスを向上させることができます。これらの設定は、ホストサーバー上の構成時の設定とは無関係です。このオプションは、Red Hat Enterprise Linux 7.0 KVM で使用できます。

CPU ピニングを有効にすると、KVM 構成でのパフォーマンスを向上できます。

CPU ピンニングの有効化

ASAv では、KVM 環境での ASAv のパフォーマンスを向上させるために KVM CPU アフィニティオプションを使用する必要があります。プロセッサアフィニティ（CPU ピニング）により、プロセスまたはスレッドと中央処理装置（CPU）や幅広い CPU 間のバインドとバインド解除が可能になり、任意の CPU ではなく、指定された CPU でのみプロセスまたはスレッドが実行されるようになります。

ピン接続されていないインスタンスでピン接続されているインスタンスのリソース要件が使用されないようにするために、CPU ピニングを使用しないインスタンスとは別のホストに CPU ピニングを使用するインスタンスを展開するようにホスト集約を設定します。

注目 NUMA トポロジを持たないインスタンスと同じホストにNUMA トポロジを持つインスタンスを展開しないでください。

このオプションを使用する場合は、KVM ホストで CPU ピンニングを構成します。

ステップ 1 KVM ホスト環境で、ピンニングに使用できる vCPU の数を調べるために、ホストのトポロジを確認します。

例：

```
virsh nodeinfo
```

ステップ 2 使用可能な vCPU の数を確認します。

例：

```
virsh capabilities
```

ステップ 3 vCPU をプロセッサ コアのセットにピンニングします。

例：

```
virsh vcpupin <vm-name> <vcpu-number> <host-core-number>
```

virsh vcpupin コマンドは、ASAv 上の vCPU ごとに実行する必要があります。次の例は、vCPU が 4 個の ASAv 構成を使用し、ホストに 8 個のコアが搭載されている場合に必要になる KVM コマンドを示しています。

```
virsh vcpupin asav 0 2
virsh vcpupin asav 1 3
virsh vcpupin asav 2 4
virsh vcpupin asav 3 5
```

ホストのコア番号は、0 ～ 7 のどの番号でもかまいません。詳細については、KVM のドキュメンテーションを参照してください。

（注） CPU ピンニングを構成する場合は、ホスト サーバーの CPU トポロジを慎重に検討してください。複数のコアで構成されたサーバーを使用している場合は、複数のソケットにまたがる CPU ピンニングを設定しないでください。

KVM 構成でのパフォーマンスの向上には、専用のシステム リソースが必要になるという短所もあります。

NUMA のガイドライン

Non-uniform Memory Access（NUMA）は、マルチプロセッサシステムのプロセッサに対するメインメモリモジュールの配置について記述する共有メモリアーキテクチャです。プロセッサが自身のノード（リモートメモリ）内に存在しないメモリにアクセスする場合は、ローカルメモリにアクセスする場合よりも低速の速度で、NUMA 接続を介してデータを転送する必要があります。

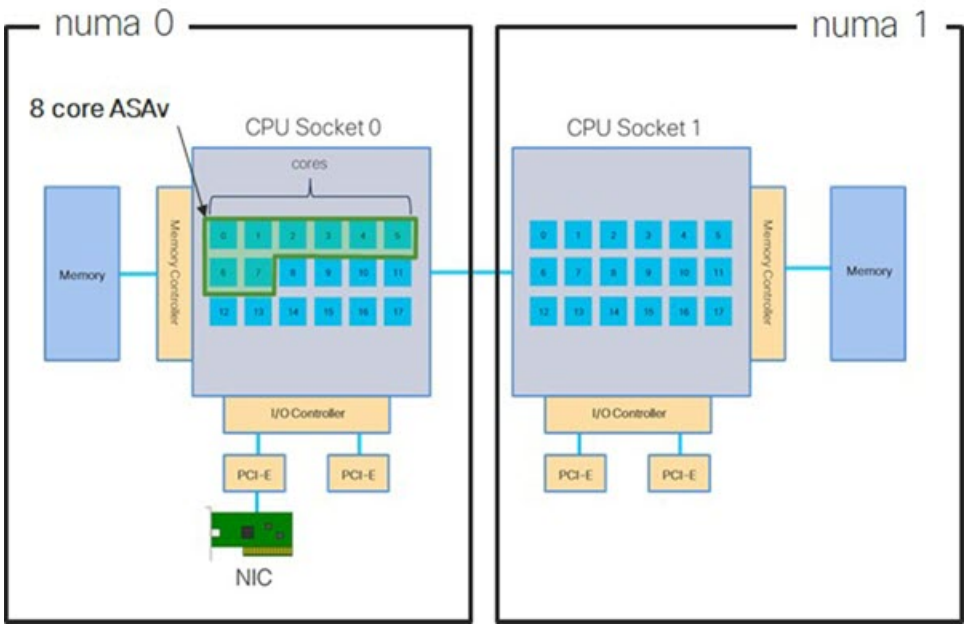
X86 サーバーアーキテクチャは、複数のソケットおよびソケット内の複数のコアで構成されています。各 CPU ソケットとそのメモリおよび I/O が、NUMA ノードと呼ばれます。メモリからパケットを効率的に読み取るには、ゲストアプリケーションおよび関連付けられている周辺機器（NIC など）が同じノード内に存在する必要があります。

最適な ASAv パフォーマンスを実現するには：

- ASAv マシンは、1つの NUMA ノード上で実行する必要があります。1つの ASAv が 2 つのソケットで実行されるように導入されている場合、パフォーマンスは大幅に低下します。
- 8 コア ASAv（図 7 : 8 コア ASAv NUMA アーキテクチャの例 （62 ページ））では、ホスト CPU の各ソケットが、それぞれ 8 個以上のコアを備えている必要があります。サーバー上で実行されている他の VM についても考慮する必要があります。
- 16 コア ASAv（図 8 : 16 コア ASAv NUMA アーキテクチャの例 （63 ページ））では、ホスト CPU 上の各ソケットが、それぞれ 16 個以上のコアを備えている必要があります。サーバー上で実行されている他の VM についても考慮する必要があります。
- NIC は、ASAv マシンと同じ NUMA ノード上にある必要があります。

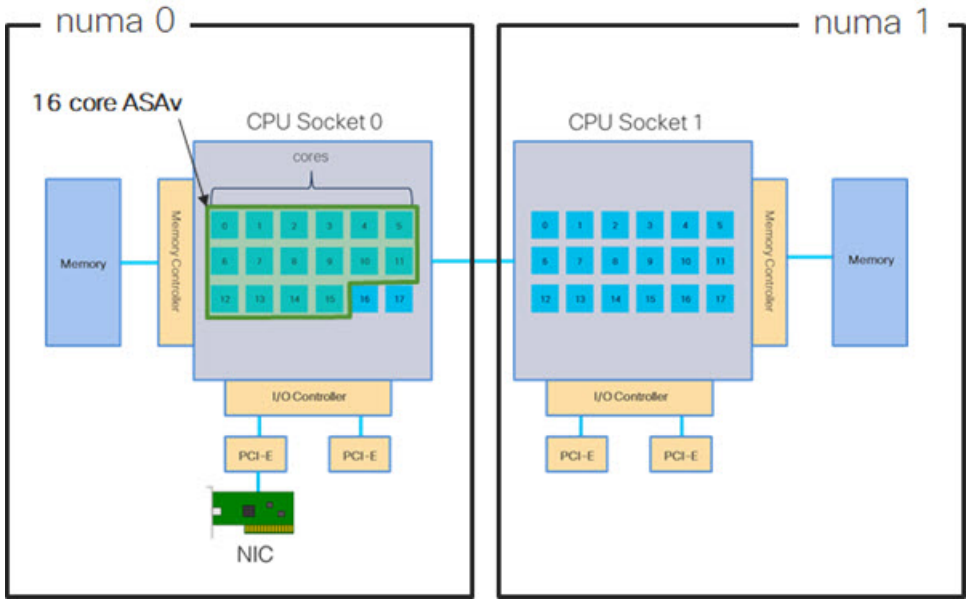
次の図は、2 つの CPU ソケットがあり、各 CPU に 18 個のコアが搭載されているサーバーを示しています。8 コア ASAv では、ホスト CPU の各ソケットに最低 8 個のコアが必要です。

図 *7 : 8* コア *ASAv NUMA* アーキテクチャの例



次の図は、2 つの CPU ソケットがあり、各 CPU に 18 個のコアが搭載されているサーバーを示しています。16 コア ASAv では、ホスト CPU の各ソケットに最低 16 個のコアが必要です。

図 *8 : 16* コア *ASAv NUMA* アーキテクチャの例



NUMA の最適化

理想的には、ASAv マシンは、NIC が動作しているノードと同じ NUMA ノード上で実行する必要があります。手順は次のとおりです。

1. 「lstopo」を使用して NIC がオンになっているノードを判別し、ノードの図を表示します。NIC を見つけて、どのノードが接続されているかをメモします。
2. KVM ホストで、`virsh list` を使用して ASAv を検出します。
3. `virsh edit <VM Number>` を使用して VM を編集します。
4. 選択したノードに ASAv を配置します。次の例では、18 コアノードを想定しています。

   ノード 0 への配置：

   ```
   <vcpu placement='static' cpuset='0-17'>16</vcpu>
   <numatune>
       <memory mode='strict' nodeset='0'/>
   </numatune>
   ```

   ノード 1 への配置：

   ```
   <vcpu placement='static' cpuset='18-35'>16</vcpu>
   <numatune>
       <memory mode='strict' nodeset='1'/>
   </numatune>
   ```

5. .xml の変更を保存し、ASAv マシンの電源を再投入します。
6. VM が目的のノードで実行されていることを確認するには、`ps aux | grep <name of your ASAv VM>` を実行して、プロセス ID を取得します。

7. `sudo numastat -c <ASAv VM Process ID>` を実行して、ASAv マシンが適切に配置されているか確認します。

KVM での NUMA 調整の使用に関する詳細については、RedHat のドキュメント『9.3. libvirt NUMA Tuning』を参照してください。

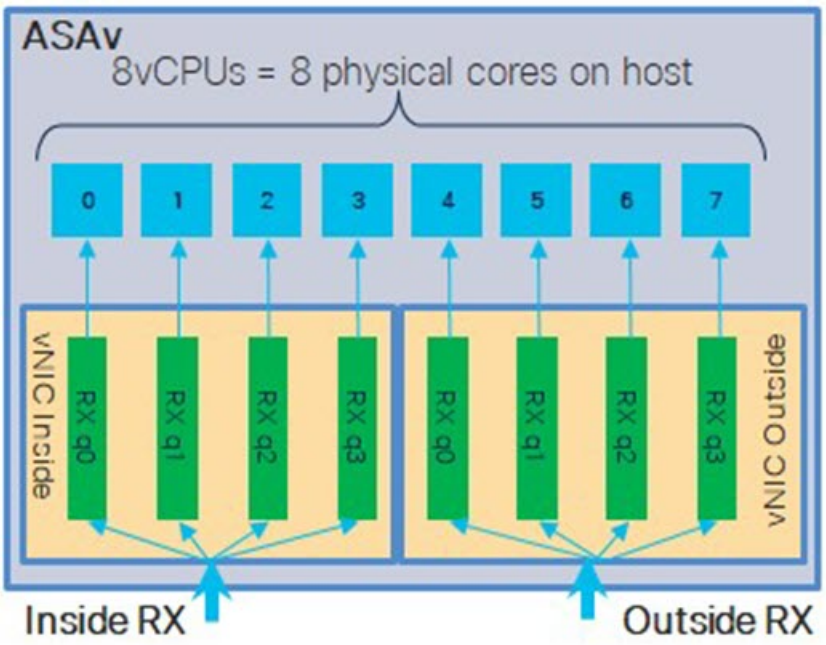
Receive Side Scaling（RSS）用の複数の RX キュー

ASAvは、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ネットワークアダプタによって使用されるテクノロジーである Receive Side Scaling（RSS）をサポートしています。最大スループットを実現するには、各 vCPU（コア）に独自の NIC RX キューが設定されている必要があります。一般的な RA VPN 展開では、1つの内部/外部ペアのインターフェイスを使用する場合があることに注意してください。

重要 複数の RX キューを使用するには、ASAv バージョン 9.13(1) 以降が必要です。KVM の場合、*libvirt* のバージョンは 1.0.6 以降である必要があります。

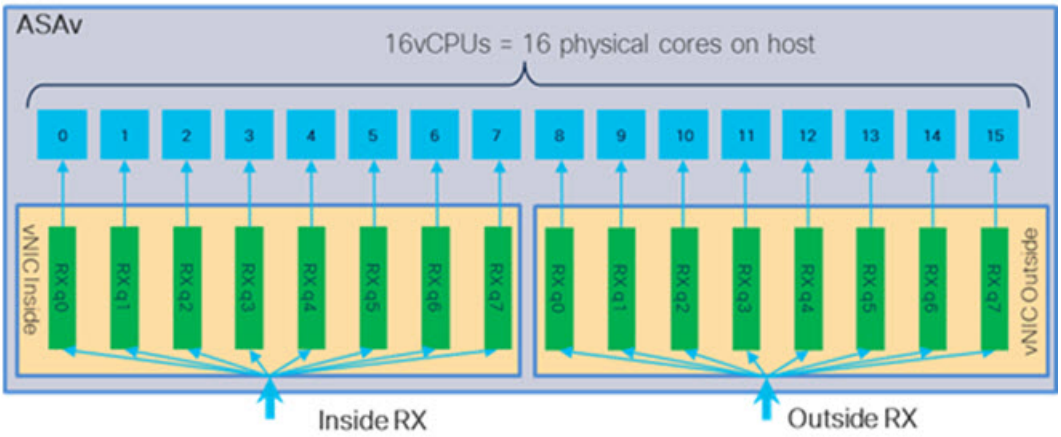
内部/外部ペアのインターフェイスを持つ 8 コア VM の場合、図 9 : 8 コア ASAv RSS RX キュー（64 ページ）に示すように、各インターフェイスには 4 つの RX キューがあります。

図 *9 : 8* コア *ASAv RSS RX* キュー



内部/外部ペアのインターフェイスを持つ 16 コア VM の場合、図 10 : 16 コア ASAv RSS RX キュー （65 ページ）に示すように、各インターフェイスには 8 つの RX キューがあり ます。

図 *10 : 16* コア *ASAv RSS RX* キュー



次の表に、KVM 用の ASAv の vNIC およびサポートされている RX キューの数を示します。サポートされている vNIC の説明については、推奨される vNIC （52 ページ）を参照してください。

表 *12 : KVM* で推奨される *NIC/vNIC*

NIC カード	vNIC ドライバ	ドライバテクノロジー	RX キューの数	パフォーマンス
x710	i40e	PCI パススルー	8（最大）	x710 の PCI パススルーおよび SR-IOV モードは、最適なパフォーマンスを提供します。通常、仮想展開では、複数の VM 間で NIC を共有できるため、SR-IOV が推奨されます。
	i40evf	SR-IOV	8	
x520	ixgbe	PCI パススルー	6	x520 NIC は、x710 よりも 10 ～ 30% パフォーマンスが低くなります。X520 の PCI パススルーおよび SR-IOV モードは、同様のパフォーマンスを提供します。通常、仮想展開では、複数の VM 間で NIC を共有できるため、SR-IOV が推奨されます。
	ixgbe-vf	SR-IOV	2	
該当なし	virtio	準仮想化	8（最大）	ASAv100 には推奨されません。 その他の展開については、KVM での Virtio のマルチキューサポートの有効化 （66 ページ） を参照してください。

KVM での Virtio のマルチキューサポートの有効化

次の例は、libvirt xml を編集するために、Virtio NIC RX キューの数を 4 に設定する方法を示しています。

```
<interface type='bridge'>
 <mac address='52:54:00:43:6e:3f'/>
 <source bridge='clients'/>
 <model type='virtio'/>
 <driver name='vhost' queues='4'/>
 <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
</interface>
```

重要 複数の RX キューをサポートするには、*libvirt* のバージョンが 1.0.6 以降である必要があります。

VPN の最適化

ASAv で VPN パフォーマンスを最適化するための追加の考慮事項は、次のとおりです。

- IPSec のスループットは DTLS よりも高くなります。
- GCM 暗号には、CBC の約 2 倍のスループットがあります。

SR-IOV インターフェイスのプロビジョニング

SR-IOV を使用すれば、複数の VM でホスト内部の 1 台の PCIe ネットワーク アダプタを共有することができます。SR-IOV は次の機能を定義しています。

- 物理機能（PF）：PF は、SR-IOV 機能を含むフル PCIe 機能です。これらは、ホスト サーバー上の通常のスタティック NIC として表示されます。
- 仮想機能（VF）：VF は、データ転送を支援する軽量 PCIe 機能です。VF は、PF から抽出され、PF を介して管理されます。

VF は、仮想化されたオペレーティング システム フレームワーク内の ASAv マシンに最大 10 Gbps の接続を提供できます。このセクションでは、KVM 環境で VF を設定する方法について説明します。ASAv 上の SR-IOV サポートについては、ASAv と SR-IOV インターフェイスのプロビジョニング （13 ページ）を参照してください。

SR-IOV インターフェイスのプロビジョニングに関する要件

SR-IOV をサポートする物理 NIC がある場合、SR-IOV 対応 VF または仮想 NIC（vNIC）を ASAv インスタンスにアタッチできます。SR-IOV は、BIOS だけでなく、ハードウェア上で実行しているオペレーティング システム インスタンスまたはハイパーバイザでのサポートも必

要です。KVM 環境で実行中の ASAv 用の SR-IOV インターフェイスのプロビジョニングに関する一般的なガイドラインのリストを以下に示します。

- ホスト サーバーには SR-IOV 対応物理 NIC が必要です。SR-IOV インターフェイスに関するガイドラインと制限事項 （13 ページ）を参照してください。
- ホスト サーバーの BIOS で仮想化が有効になっている必要があります。詳細については、ベンダーのマニュアルを参照してください。
- ホスト サーバーの BIOS で IOMMU グローバル サポートが SR-IOV に対して有効になっている必要があります。詳細については、ハードウェア ベンダーのマニュアルを参照してください。

KVM ホスト BIOS とホスト OS の変更

このセクションでは、KVM システム上の SR-IOV インターフェイスのプロビジョニングに関するさまざまなセットアップ手順と設定手順を示します。このセクション内の情報は、Intel Ethernet Server Adapter X520 - DA2 を使用した Cisco UCS C シリーズ サーバー上の Ubuntu 14.04 を使用して、特定のラボ環境内のデバイスから作成されたものです。

始める前に

- SR-IOV 互換ネットワーク インターフェイス カード（NIC）が取り付けられていることを確認します。
- Intel 仮想化テクノロジー（VT-x）機能と VT-d 機能が有効になっていることを確認します。

（注） システム メーカーによっては、これらの拡張機能がデフォルトで無効になっている場合があります。システムごとに BIOS 設定にアクセスして変更する方法が異なるため、ベンダーのマニュアルでプロセスを確認することをお勧めします。

- オペレーティング システムのインストール中に、Linux KVM モジュール、ライブラリ、ユーザ ツール、およびユーティリティのすべてがインストールされていることを確認します。ASAv と KVM の前提条件 （54 ページ）を参照してください。
- 物理インターフェイスが稼働状態であることを確認します。ifconfig <*ethname*> を使用して確認します。

ステップ 1 "root" ユーザー アカウントとパスワードを使用してシステムにログインします。

ステップ 2 Intel VT-d が有効になっていることを確認します。

例：

```
kvmuser@kvm-host:/$ dmesg | grep -e DMAR -e IOMMU
[    0.000000] ACPI: DMAR 0x000000006F9A4C68 000140 (v01 Cisco0 CiscoUCS 00000001 INTL 20091013)
[    0.000000] DMAR: IOMMU enabled
```

最後の行は、VT-d が有効になっていることを示しています。

ステップ 3 */etc/default/grub* 設定ファイル内の GRUB_CMDLINE_LINUX エントリに *intel_iommu=on* パラメータを付加することによって、カーネル内の Intel VT-d をアクティブにします。

例：

```
# vi /etc/default/grub
...
GRUB_CMDLINE_LINUX="nofb splash=quiet console=tty0 ... intel_iommu=on"
...
```

（注） AMD プロセッサを使用している場合は、代わりに、*amd_iommu=on* をブート パラメータに付加します。

ステップ 4 iommu の変更を有効にするためにサーバーをリブートします。

例：

```
> shutdown -r now
```

ステップ 5 次の形式を使用して *sysfs* インターフェイス経由で *sriov_numvfs* パラメータに適切な値を書き込むことによって、VF を作成します。

```
#echo n > /sys/class/net/device name/device/sriov_numvfs
```

サーバーの電源を入れ直すたびに必要な数の VF が作成されるようにするには、*/etc/rc.d/* ディレクトリに配置されている *rc.local* ファイルに上記コマンドを付加します。Linux OS は、ブートプロセスの最後で *rc.local* スクリプトを実行します。

たとえば、ポートあたり 1 つの VF を作成するケースを以下に示します。お使いのセットアップではインターフェイスが異なる可能性があります。

例：

```
echo '1' > /sys/class/net/eth4/device/sriov_numvfs
echo '1' > /sys/class/net/eth5/device/sriov_numvfs
echo '1' > /sys/class/net/eth6/device/sriov_numvfs
echo '1' > /sys/class/net/eth7/device/sriov_numvfs
```

ステップ 6 サーバーをリブートします。

例：

```
> shutdown -r now
```

ステップ 7 *lspci* を使用して、VF が作成されたことを確認します。

例：

```
 > lspci | grep -i "Virtual Function"
kvmuser@kvm-racetrack:~$ lspci | grep -i "Virtual Function"
0a:10.0 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.1 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.2 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.3 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
```

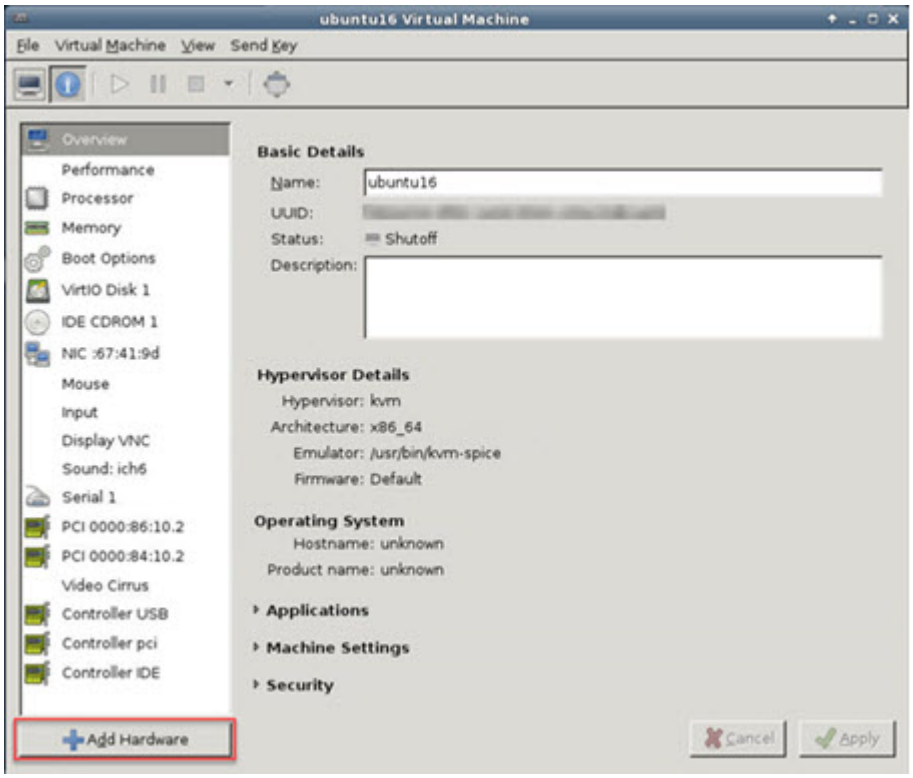
（注） **ifconfig** コマンドを使用して、新しいインターフェイスを表示します。

ASAv への PCI デバイスの割り当て

VF を作成したら、PCI デバイスを追加するのと同様に、VF を ASAv に追加できます。次の例では、グラフィカル **virt-manager** ツールを使用して、イーサネット VF コントローラを ASAv に追加する方法について説明します。

ステップ 1 ASAv を開いて、[Add Hardware] ボタンをクリックし、新しいデバイスを仮想マシンに追加します。

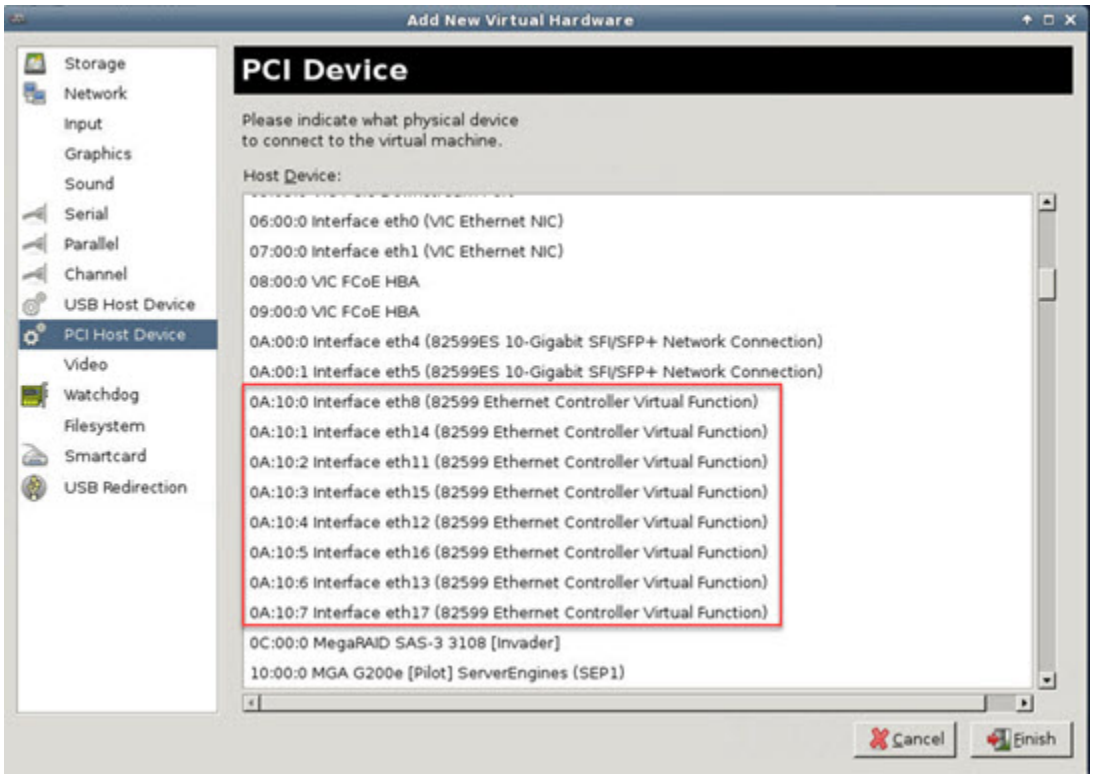
図 *11* : ハードウェアの追加



ステップ 2 左ペインの [Hardware] リストで [PCI Host Device] をクリックします。

VF を含む PCI デバイスのリストが中央ペインに表示されます。

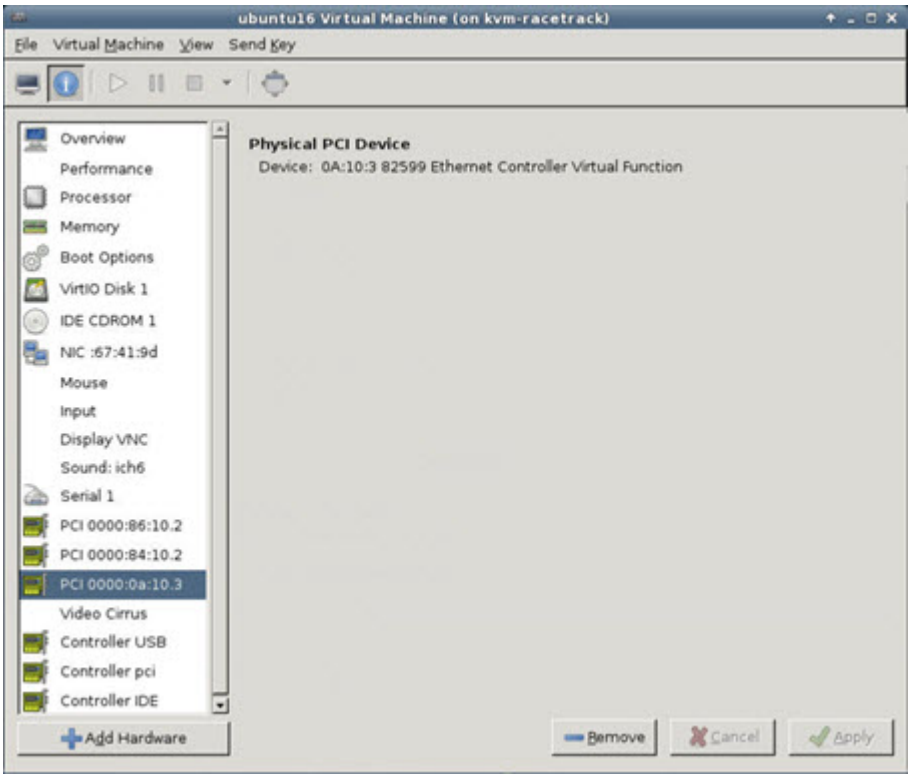
図 *12 :* 仮想機能のリスト



ステップ 3 使用可能な仮想機能のいずれかを選択して、[Finish] をクリックします。

PCI デバイスがハードウェア リストに表示されます。デバイスの記述が Ethernet Controller Virtual Function になっていることに注意してください。

図 *13* : 追加された仮想機能



次のタスク

- ASAv コマンドラインから、**show interface** コマンドを使用して、新しく設定したインターフェイスを確認します。
- ASAv でインターフェイス コンフィギュレーション モードを使用して、トラフィックの送受信インターフェイスを設定して有効化します。詳細については、『Cisco ASA シリーズ CLI コンフィギュレーション ガイド（一般的な操作）』の「*Basic Interface Configuration*」の章を参照してください。

CPU 使用率とレポート

CPU使用率レポートには、指定された時間内に使用されたCPUの割合の要約が表示されます。通常、コアはピーク時以外には合計 CPU 容量の約 30 ～ 40% で動作し、ピーク時は約 60 ～ 70% の容量で動作します。

重要 9.13(1) 以降では、サポートされているすべての ASA Virtual vCPU/メモリ構成ですべての ASA Virtual ライセンスを使用できるようになり、ASA Virtual 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ます。

ASA Virtual の vCPU 使用率

ASA Virtual の vCPU 使用率には、データパス、制御ポイント、および外部プロセスで使用されている vCPU の量が表示されます。

vSphere で報告される vCPU の使用率には、ASA Virtual の使用率に加えて、次のものが含まれます。

- ASA Virtual アイドル時間
- ASA Virtual マシンに使用された %SYS オーバーヘッド
- vSwitch、vNIC および pNIC の間を移動するパケットのオーバーヘッド。このオーバーヘッドは非常に大きくなる場合があります。

CPU 使用率の例

CPU 使用率の統計情報を表示するには、**show cpu usage** コマンドを使用します。

例

```
Ciscoasa#show cpu usage

CPU □□□□5□□□ 1%□1 □□□ 2%□5 □□□ 1%
```

報告された vCPU の使用率が大幅に異なる例を次に示します。

- ASA Virtual レポート：40%
- DP：35%
- 外部プロセス：5%
- ASA（ASA Virtual レポート）：40%
- ASA アイドル ポーリング：10%
- オーバーヘッド：45%

オーバーヘッドは、ハイパーバイザ機能の実行、および vSwitch を使用した NIC と vNIC の間のパケット転送に使用されています。

KVM CPU 使用率レポート

値は、

```
virsh cpu-stats domain --total start count
```

コマンドを実行すると、指定されたゲスト仮想マシンの CPU 統計情報が表示されます。デフォルトでは、すべての CPU の統計と合計が表示されます。`--total` オプションを指定すると、合計統計のみ表示されます。`--count` オプションを指定すると、count 個の CPU の統計のみ表示されます。

OProfile、top などのツールを実行すると、ハイパーバイザと VM の両方の CPU 使用率を含む、特定の KVM VM の合計 CPU 使用率が表示されます。同様に、Xen VMM に固有の XenMon などのツールの場合、Xen ハイパーバイザ、つまり Dom 0 の合計 CPU 使用率が表示されますが、VM ごとのハイパーバイザ使用率には分割されません。

これらのツールとは別に、OpenNebula などのクラウド コンピューティング フレームワークには、VM によって使用される仮想 CPU の割合の大まかな情報のみを提供する特定のツールが存在します。

ASA Virtual と KVM のグラフ

ASA Virtual と KVM の間には CPU % の数値に違いがあります。

- KVM グラフの数値は ASA Virtual の数値よりも常に大きくなります。
- KVM ではこの値は「%CPU usage」と呼ばれ、ASA Virtual ではこの値は「%CPU utilization」と呼ばれます。

用語「%CPU utilization」と「%CPU usage」は別のものを意味しています。

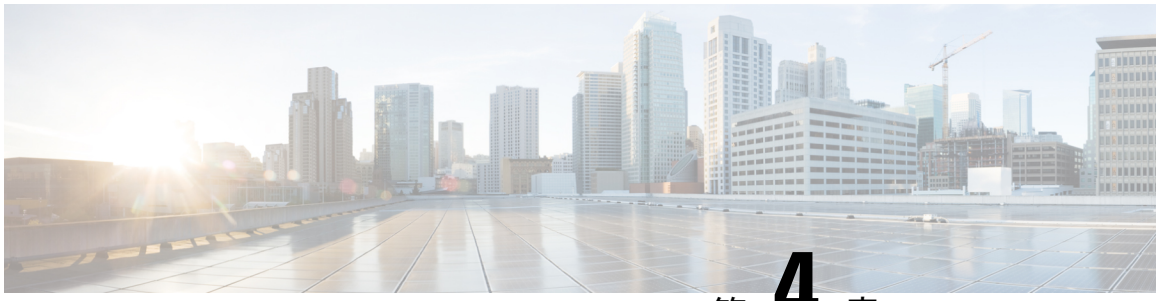
- CPU utilization は、物理 CPU の統計情報を提供します。
- CPU usage は CPU のハイパースレッディングに基づいた論理 CPU の統計情報を提供します。しかし、1 つの vCPU のみが使用されるため、ハイパースレッディングは動作しません。

KVM では「%CPU usage」は次のように計算されます。

アクティブに使用された仮想 CPU の量。使用可能な CPU の合計に対する割合として指定されます。

この計算は、ホストから見た CPU 使用率であり、ゲスト オペレーティング システムから見た CPU 使用率ではありません。また、これは仮想マシンで使用可能なすべての仮想 CPU の平均 CPU 使用率になります。

たとえば、1 個の仮想 CPU を搭載した 1 つの仮想マシンが、4 個の物理 CPU を搭載した 1 台のホストで実行されており、その CPU 使用率が 100% の場合、仮想マシンは、1 個の物理 CPU をすべて使用しています。仮想 CPU の使用率は、「MHz 単位の使用率 / 仮想 CPU の数 x コア周波数」として計算されます。



第 4 章

AWS クラウドへの ASAv の導入

Amazon Web Services（AWS）クラウドに ASAv 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 ASAv vCPU/メモリ構成ですべての ASAv ライセンスを使用できるようになりました。これにより、ASAv 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。

AWS クラウドへの ASAv の導入について

ASAv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。ASAv 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。

ASAv は、次の AWS インスタンスタイプをサポートしています。

表 *13 : 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
	vCPU	メモリ（GB）	
c5.xlarge	4	8	4

インスタンス	属性		インターフェイス
	vCPU	メモリ（GB）	
c5.2xlarge	8	16	4
c4.large	2	3.75	3
c4.xlarge	4	7.5	4
c4.2xlarge	8	15	4
c3.large	2	3.75	3
c3.xlarge	4	7.5	4
c3.2xlarge	8	15	4
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	5.25	3
c5n.xlarge	4	10.5	4
c5n.2xlarge	8	21	4
c5n.4xlarge	16	54	8
g4ad.4xlarge	16	64	3
g4dn.xlarge	4	16	3
g4dn.2xlarge	8	32	3
g4dn.4xlarge	16	64	3
i3en.large	2	16	3
i3en.xlarge	4	32	4

インスタンス	属性		インターフェイス
	vCPU	メモリ（GB）	
i3en.2xlarge	8	64	4
i3en.3xlarge	12	96	4
inf1.xlarge	4	8	4
inf1.2xlarge	8	16	4
m4.large	2	4	3
m4.xlarge	4	16	4
m4.2xlarge	8	32	4
m5.large	2	8	3
m5.xlarge	4	16	4
m5.2xlarge	8	32	4
m5.4xlarge	16	64	8
m5a.large	2	8	3
m5a.xlarge	4	16	4
m5a.2xlarge	8	32	4
m5a.4xlarge	16	64	8
m5ad.large	2	8	3
m5ad.xlarge	4	16	4
m5ad.2xlarge	8	32	4
m5ad.4xlarge	16	64	8
m5d.large	2	8	3
m5d.xlarge	4	16	4
m5d.2xlarge	8	32	4
m5d.4xlarge	16	64	8
m5dn.large	2	8	3
m5dn.xlarge	4	16	4
m5dn.2xlarge	8	32	4
m5dn.4xlarge	16	64	8
m5n.large	2	8	3
m5n.xlarge	4	16	4
m5n.2xlarge	8	32	4

インスタンス	属性		インターフェイス
	vCPU	メモリ（GB）	
m5n.4xlarge	16	64	8
m5zn.large	2	8	3
m5zn.xlarge	4	16	4
m5zn.2xlarge	8	32	4
m5zn.3xlarge	12	48	8
r5.large	2	16	3
r5.xlarge	4	32	4
r5.2xlarge	8	64	4
r5.4xlarge	16	128	8
r5a.large	2	16	3
r5a.xlarge	4	32	4
r5a.2xlarge	8	64	4
r5a.4xlarge	16	128	8
r5ad.large	2	16	3
r5ad.xlarge	4	32	4
r5ad.2xlarge	8	64	4
r5ad.4xlarge	16	128	8
r5b.large	2	16	3
r5b.xlarge	4	32	4
r5b.2xlarge	8	64	4
r5b.4xlarge	16	128	8
r5d.large	2	16	3
r5d.xlarge	4	32	4
r5d.2xlarge	8	64	4
r5d.4xlarge	16	128	8
r5dn.large	2	16	3
r5dn.xlarge	4	32	4
r5dn.2xlarge	8	64	4
r5dn.4xlarge	16	128	8
r5n.large	2	16	3
r5n.xlarge	4	32	4

インスタンス	属性		インターフェイス
	vCPU	メモリ（GB）	
r5n.2xlarge	8	64	4
r5n.4xlarge	16	128	8
z1d.large	2	16	3
z1d.xlarge	4	32	4
z1d.2xlarge	8	64	4
z1d.3xlarge	12	96	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 C5 または M5 インスタンスタイプに移行することを推奨します。

AWS にアカウントを作成し、AWS ウィザードを使用して ASAv 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。

ASAv と AWS の前提条件

- aws.amazon.com でアカウントを作成します。
- ASAv へのライセンス付与。ASAv 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。「ASAv のライセンス（1 ページ）」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
  - 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 ASAv 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 ASAv 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）：ASAv 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 ASAv を接続するために使用されます。

- ASAv システム要件については、Cisco ASA の互換性 [英語] を参照してください。

ASAv および AWS のガイドラインと制限事項

サポートされる機能

AWS 上の ASAv 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）への展開
- 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展開
- L3 ネットワークのユーザー展開
- ルーテッド モード（デフォルト）
- Amazon CloudWatch

サポートされない機能

AWS 上の ASAv 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
- クラスタ
- ASAv 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。

- IPv6

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 ASAv 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。

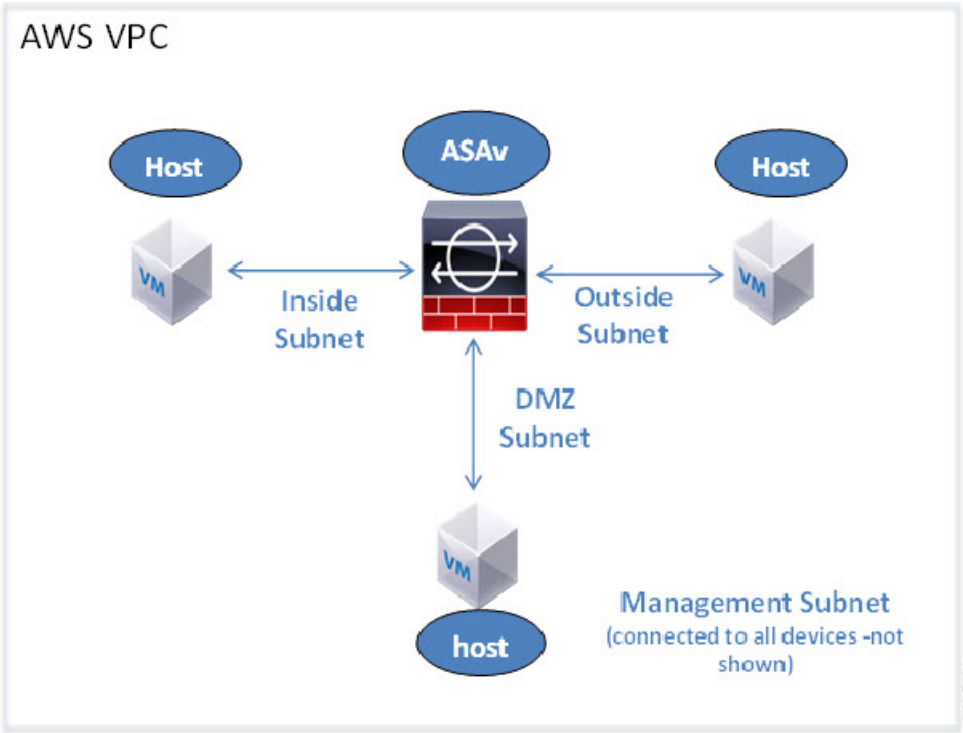
アップグレード後は、 **username** コマンドに対する **password** または **nopassword** キーワードの指定は任意となり、ユーザーがパスワードを入力できなくするよう指定できるようになります。よって、公開キー認証のみを強制的に使用する場合は、**username** コマンドを入力しなおします。

```
username admin privilege 15
```

AWS 上の ASAv のネットワークトポロジの例

次の図は、ASAv 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 ASAv の推奨トポロジを示しています。

図 *14 : AWS* への *ASAv* の導入例



AWS での ASAv の展開

次の手順は、ASAv で AWS をセットアップする手順の概略です。設定の詳細な手順については、『Getting Started with AWS』を参照してください。

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、画面の右上隅に表示されます。ある地域内のリソースは、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 **[My Account]** > **[AWS Management Console]** をクリックし、[Networking] で **[VPC]** > **[Start VPC Wizard]** をクリックして、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できます）。

- 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：インターネット経由の直接接続を有効にします（インターネット ゲートウェイの名前を入力します）。
- 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）。

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（たとえば、Ubuntu Server 14.04 LTS）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。
- ASAv 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、ASAv の起動時に適用される ASAv 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 ASAv を導入すると、ASAv はデフォルトの ASAv 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。
- **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

no shutdown
!
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

no shutdown
!
interface G0/1
nameif inside
ip address dhcp

no shutdown
!
```

- ストレージ（デフォルトを受け入れます）。
- 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。タグを容易に見つけるために使用できる名前を付けます。
- 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。ASAv 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。
- 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、ASAv 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 ASAv 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。

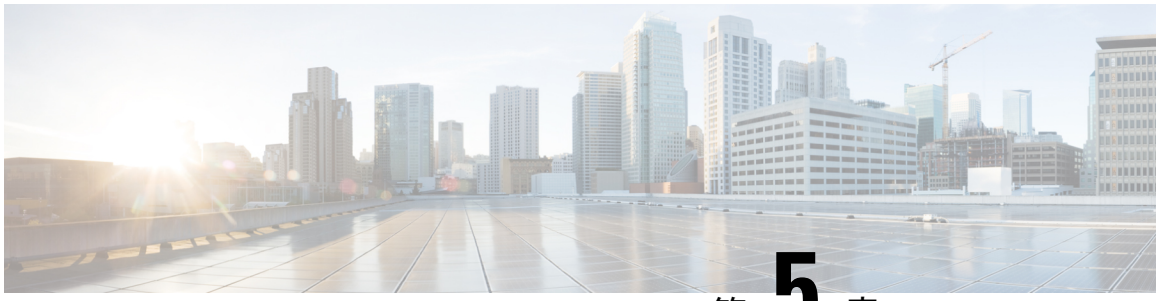
AWS のデフォルト設定では、インスタンスはその IP アドレス（IPv4）のトラフィックのみを受信でき、インスタンスは独自の IP アドレス（IPv4）からのみトラフィックを送信できます。ASAv のルーテッドホップとしての動作を有効にするには、ASAv の各トラフィックインターフェイス（内部、外部、および DMZ）の [送信元または宛先の確認（Source/Destination Check）] を無効にする必要があります。

AWS での ASAv のパフォーマンス調整

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）



第 5 章

AWS への ASAv Auto Scale ソリューションの導入

AWS での FTDv ASAv 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 ASAv 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、ASAv 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 ASAv Auto Scale は、AWS 環境の ASAv 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、FMC によって管理されるでサポートされます。

ASAv 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 ASAv 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。

Auto Scale の導入例

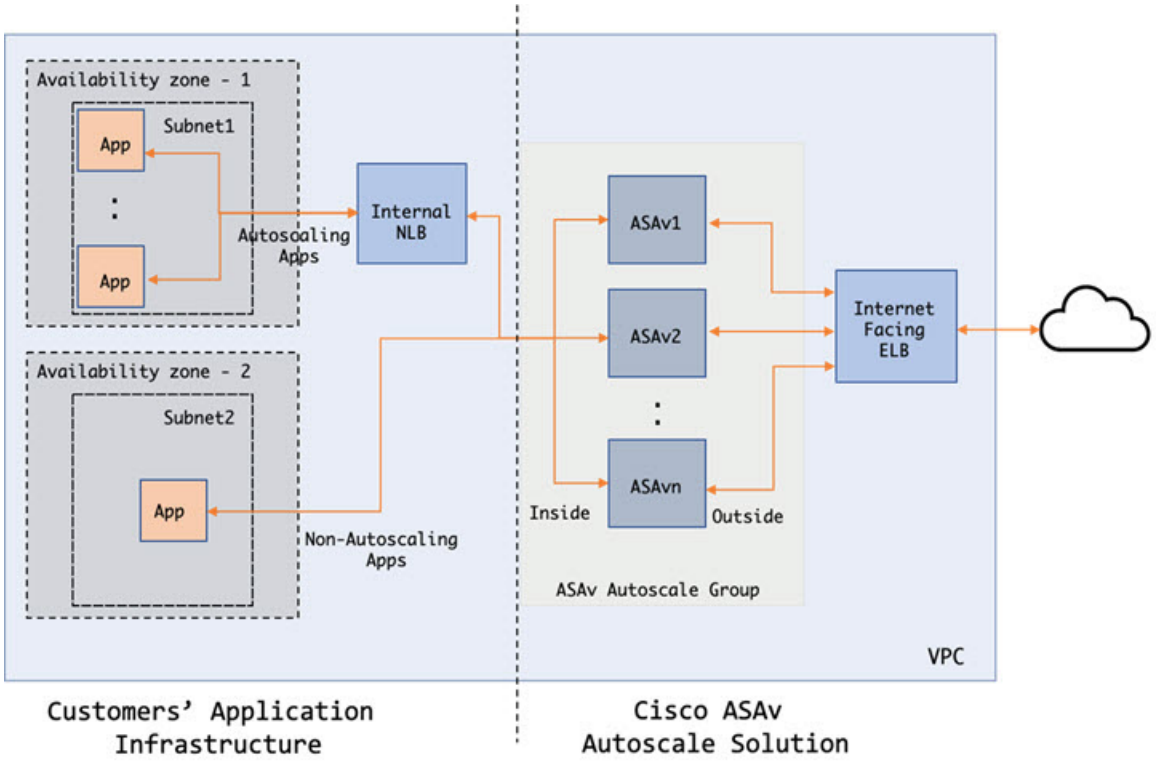
この ASAv AWS Auto Scale ソリューションの導入例は、導入例の図に示されています。AWS ロードバランサはインバウンドで開始された接続のみを許可するため、外部で生成されたトラフィックのみが ASAv ファイアウォール経由で内部を通過できます。

（注） 前提条件のSSL サーバー証明書 （93 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 ASAv 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 ASAv を通過しません。

図 *15 : ASAv 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。

Auto Scale ソリューションの仕組み

ASAv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 ASAv インスタンスの追加または削除を指示し、 ASAv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 ASAv 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 ASAv 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、ASAv 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 ASAv 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

ASAv 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、ASAv 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、ASAv 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。ASAv 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 ASAv 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、ASAv 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 *14 :* 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、ASAv 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 *15 : 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（ASAv-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 *ASAv-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-ASAv-1
NotifyEmailID	文字列	Auto Scale イベントはこの電子メールアドレスに送信されます。サブスクリプション電子メール要求を受け入れる必要があります。 例：admin@company.com
VpcId	文字列	デバイスを展開する必要がある VPC ID。これは、AWS の要件に従って設定する必要があります。 タイプ：AWS::EC2::VPC::Id 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
LambdaSubnets	リスト	Lambda 関数が展開されるサブネット。 タイプ：List<AWS::EC2::Subnet::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。
LambdaSG	リスト	Lambda 機能のセキュリティグループ。 タイプ：List<AWS::EC2::SecurityGroup::Id> 「*infrastructure.yaml*」ファイルを使用してインフラストラクチャを展開すると、スタックの出力セクションにこの値が設定されます。その値を使用してください。

パラメータ	使用できる値/タイプ	説明
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
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
SSLcertificate	文字列	セキュアポート接続の SSL 証明書の ARN。指定しない場合、ロードバランサで開かれるポートは TCP/HTTP になります。指定した場合、ロードバランサで開かれるポートは TLS/HTTPS になります。

パラメータ	使用できる値/タイプ	説明
TgHealthPort	整数	このポートは、正常性プローブのターゲットグループによって使用されます。ASAv 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 ASAv 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、ASAv は Unhealthy インスタンスのしきい値アラームにより、非正常としてマークされ、削除されます。 例：8080
AssignPublicIP	ブール値	「true」を選択すると、パブリック IP が割り当てられます。BYOL タイプの ASAv 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。 ASAv 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	ASAv 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	ASAv AMI ID（有効な Cisco ASAv 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	ASAv 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	ASAv 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、ASAv 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、ASAv 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 ASAv 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

ASAv 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。

ASAv 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

ASAv 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 （94 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 ASAv 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 ASAv 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 ASAv 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 ASAv VM に属する異常な IP がある場合、ASAv 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 ASAv 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、ASAv 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

ASAv 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 ASAv 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、ASAv 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（104 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 （104 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 （104 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。

ログイン情報と登録 ID 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。

ASAv の管理者パスワードを変更します。

ASAv 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 ASAv デバイスをオンボードする場合、ASAv 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

ASAv 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 ASAv 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。

ASAv が設定に失敗

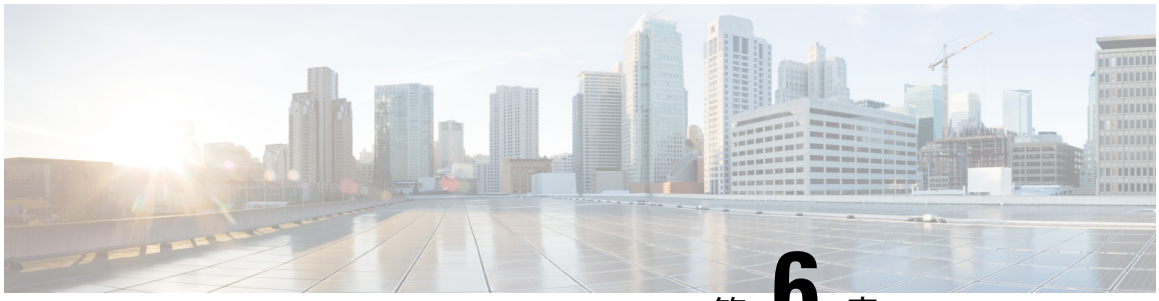
ASAv の設定に失敗した場合は、Amazon S3 の静的な HTTP Web サーバーのホスティング構成への接続を確認してください。詳細については、「Amazon S3 での静的な Web サイトのホスティング（Hosting a static website on Amazon S3）」https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.htmlを参照してください。

ASAv でライセンス交付に失敗

ASAv がライセンスに失敗した場合は、CSSM サーバーへの接続、ASAv 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。

ASAv に SSH 接続できない

ASAv 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 ASAv に渡されたかどうかを確認します。



第 6 章

Microsoft Azure クラウドへの ASAv の導入

Microsoft Azure クラウドに ASAv 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 ASAv vCPU/メモリ構成ですべての ASAv ライセンスを使用できるようになりました。これにより、ASAv 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 Azure インスタンスタイプの数も増えます。

Microsoft Azure クラウドへの ASAv 導入について

ASAv のニーズに合わせて Azure 仮想マシンの階層とサイズを選択します。すべての ASAv ライセンスを、サポートされているすべての ASAv vCPU/メモリ構成で使用できます。そのため、さまざまな Azure インスタンスタイプで ASAv を実行できます。

表 *16 : Azure* 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
	vCPU	メモリ（GB）	
D3、D3_v2、DS3、DS3_v2	4	14	4
D4、D4_v2、DS4、DS4_v2	8	36	8
D5、D5_v2、DS5、DS5_v2	16	72	8
D8_v3	8	32	4
D16_v3	16	64	4
D8s_v3	8	32	4
D16s_v3	16	64	8
F4、F4s	4	8	4
F8、F8s	8	16	8
F16、F16s	16	32	8
F8s_v2	8	16	4
F16s_v2	16	32	8

次の方法で Microsoft Azure に ASAv を導入できます。

- 標準的な Azure パブリック クラウドおよび Azure Government 環境で、Azure Resource Manager を使用してスタンドアロン ファイアウォールとして導入
- Azure Security Center を使用して統合パートナー ソリューションとして導入
- 標準的な Azure パブリック クラウドおよび Azure Government 環境で、Azure Resource Manager を使用してハイ アベイラビリティ（HA）ペアとして導入

「Azure Resource Manager からの ASAv の導入 （119 ページ）」を参照してください。標準的な Azure パブリッククラウドおよび Azure Government 環境で ASAv HA 構成を導入できます。

ASAv および Azure の前提条件およびシステム要件

- Azure.com でアカウントを作成します。

Microsoft Azure でアカウントを作成したら、ログインして、Microsoft Azure Marketplace 内で ASAv を選択し、ASAv を導入できます。

- ASAv へのライセンス付与。

  ASAv 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。「Smart Software Licensing for the ASAv」を参照してください。

（注） Azure に導入する場合、ASAv にはデフォルトで 2Gbps の権限が付与されています。100Mbps および 1Gbps の権限付与を使用できます。ただし、100Mbps または 1Gbps の権限付与を使用できるように、スループットレベルを明示的に設定する必要があります。

- インターフェイスの要件：

  4 つのネットワーク上の 4 つのインターフェイスとともに ASAv を導入する必要があります。任意のインターフェイスにパブリック IP アドレスを割り当てることができます。パブリック IP アドレスの作成、変更、削除など、パブリック IP に関する Azure のガイドラインについては、パブリック IP アドレス [英語] を参照してください。

- 管理インターフェイス：

  Azure では、最初に定義されたインターフェイスが常に管理インターフェイスです。

- 通信パス：

  - 管理インターフェイス：SSH アクセス、および ASAv を ASDM に接続するために使用されます。

（注） Azure Accelerated Networking は、管理インターフェイスではサポートされていません。

  - 内部インターフェイス（必須）：内部ホストに ASAv 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）：ASAv 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）：Standard_D3 インターフェイスを使用する場合に、ASAv を DMZ ネットワークに接続するために使用されます。

- ASAv ハイパーバイザおよび仮想プラットフォームのサポート情報については、Cisco 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

- Microsoft Azure クラウドからの導入
- Azure Accelerated Networking（AN）
- 選択したインスタンスタイプに基づく最大 16 個の vCPU

（注） Azure では L2 vSwitch 機能は設定できません。

- インターフェイスのパブリック IP アドレス

  任意のインターフェイスにパブリック IP アドレスを割り当てることができます。パブリック IP アドレスの作成、変更、削除など、パブリック IP に関する Azure のガイドラインについては、パブリック IP アドレス [英語] を参照してください。

- ルーテッド ファイアウォール モード（デフォルト）

（注） ルーテッド ファイアウォール モードでは、ASAv はネットワーク内の従来のレイヤ 3 境界となります。このモードには、各インターフェイスの IP アドレスが必要です。Azure は VLAN タグ付きインターフェイスをサポートしていないため、IP アドレスはタグなしのトランク以外のインターフェイスで設定する必要があります。

既知の問題

アイドル タイムアウト

Azure 上の ASAv には、VM で設定可能なアイドルタイムアウトがあります。最小設定値は 4 分、最大設定値は 30 分です。ただし、SSH セッションでは最小設定値は 5 分、最大設定値は 60 分です。

（注） ASAv のアイドルタイムアウトにより、SSH タイムアウトは常に上書きされ、セッションが切断されることに注意してください。セッションがどちらの側からもタイムアウトしないように、VM のアイドルタイムアウトを SSH タイムアウトに合わせることができます。

プライマリ ASAv からスタンバイ ASAv へのフェールオーバー

Azure での ASAv HA 導入で Azure のアップグレードが発生すると、プライマリ ASAv からスタンバイ ASAv へのフェールオーバーが発生する場合があります。Azure のアップグレードにより、プライマリ ASAv が一時停止状態になります。プライマリ ASAv が一時停止している場合、スタンバイ ASAv は hello パケットを受信しません。スタンバイ ASAv がフェールオーバーホールド時間を経過しても hello パケットを受信しない場合、スタンバイ ASAv へのフェールオーバーが発生します。

また、フェールオーバーホールド時間を経過していなくてもフェールオーバーが発生する可能性があります。プライマリ ASAv が一時停止状態に入ってから 19 秒後に再開するシナリオを考えてみましょう。フェールオーバーホールド時間は 30 秒ですが、クロックは約 2 分ごとに同期されるため、スタンバイ ASAv は正しいタイムスタンプの hello パケットを受信しません。その結果、プライマリ ASAv からスタンバイ ASAv へのフェールオーバーが発生します。

（注） この機能は IPv4 のみをサポートし、ASA Virtual HA は IPv6 設定ではサポートされません。

サポートされない機能

- 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）
- ユーザー インスタンス インターフェイスの VLAN タギング
- ジャンボ フレーム
- Azure の観点からの、デバイスが所有していない IP アドレスのプロキシ ARP
- 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）

（注） Azure ポリシーでは、インターフェイスは無差別モードでは動作できないため、ASAv はトランスペアレント ファイアウォール モードでは動作しません。

- マルチ コンテキスト モード
- クラスタ
- ASAv ネイティブ HA
- VM のインポート/エクスポート
- デフォルトでは、Azureクラウド内で稼働する ASAv の FIPS モードは無効になっています。

（注） FIPS モードを有効にする場合は、**ssh key-exchange group dh-group14-sha1** コマンドを使用して、Diffie-Helman キー交換グループをより強力なキーに変更する必要があります。Diffie-Helman グループを変更しないと、ASAv に SSH 接続できなくなるため、グループの変更が、最初に ASAv を管理する唯一の方法です。

- IPv6

Azure DDoS Protection 機能

Microsoft Azure の Azure DDoS Protection は、ASAv の最前線に実装された追加機能です。仮想ネットワークでこの機能を有効にすると、ネットワークで予想されるトラフィックの 1 秒あたりのパケット数に応じて、一般的なネットワーク層攻撃からアプリケーションを保護するのに役立ちます。この機能は、ネットワーク トラフィック パターンに基づいてカスタマイズできます。

Azure DDoS Protection 機能の詳細については、『Azure DDoS Protection Standard overview』[英語] を参照してください。

導入時に作成されるリソース

Azure に ASAv を展開すると、次のリソースが作成されます。

- ASAv マシン
- リソース グループ（既存のリソース グループを選択していない場合）

  ASAv リソースグループは、仮想ネットワークとストレージアカウントで使用するリソースグループと同じである必要があります。

- vm name-Nic0、vm name-Nic1、vm name-Nic2、vm name-Nic3 という名前の 4 つの NIC

  これらの NIC は、それぞれ ASAv インターフェイスの Management 0/0、GigabitEthernet 0/0、GigabitEthernet 0/1、および GigabitEthernet 0/2 にマッピングされます。

（注） 要件に基づいて、IPv4 のみで VNet を作成できます。

- VM 名-SSH-SecurityGroup という名前のセキュリティ グループ

  セキュリティグループは、ASAv Management 0/0 にマッピングされる VM の Nic0 にアタッチされます。

  セキュリティ グループには、VPN 目的で SSH、UDP ポート 500、および UDP 4500 を許可するルールが含まれます。導入後に、これらの値を変更できます。

- パブリック IP アドレス（展開時に選択した値に従って命名）。

  パブリック IP アドレス（IPv4 のみ）。

  任意のインターフェイスにパブリック IP アドレスを割り当てることができます。パブリック IP アドレスの作成、変更、削除など、パブリック IP に関する Azure のガイドラインについては、「パブリック IP アドレス」を参照してください。

- 4 つのサブネットを備えた仮想ネットワーク（既存のネットワークを選択していない場合）
- サブネットごとのルーティング テーブル（既存の場合は最新のもの）

  このテーブルの名前は、サブネット名-ASAv-RouteTable です。

  各ルーティングテーブルには、ASAv IP アドレスを持つ他の 3 つのサブネットへのルートがネクストホップとして含まれています。トラフィックを他のサブネットまたはインターネットに到達させる必要がある場合は、デフォルトルートを追加することもできます。

- 選択したストレージアカウントの起動時診断ファイル

  起動時診断ファイルは、ブロブ（サイズの大きいバイナリオブジェクト）内に配置されます。

- 選択したストレージ アカウントのブロブおよびコンテナ VHD にある 2 つのファイル（名前は、*vm name*-disk.vhd および *vm name*-<uuid>.status）
- ストレージアカウント（既存のストレージアカウントが選択されていない場合）

（注） VM を削除すると、保持を希望する任意のリソースを除き、これらの各リソースを個別に削除する必要があります。

Azure ルーティング

Azure 仮想ネットワークでのルーティングは、仮想ネットワークの有効なルーティング テーブルによって決まります。有効なルーティング テーブルは、既存のシステム ルーティング テーブルとユーザー定義のルーティング テーブルの組み合わせです。

（注） ASAv では、Azure クラウドルーティングの特性により、EIGRP や OSPF などのダイナミックな内部ルーティングプロトコルを使用できません。有効なルーティング テーブルは、仮想クライアントにスタティック/ダイナミック ルートが設定されているかどうかに関係なく、ネクスト ホップを決定します。

現在、有効なルーティング テーブルまたはシステム ルーティング テーブルはどちらも表示できません。

ユーザー定義のルーティング テーブルは表示および編集できます。システム テーブルとユーザー定義のテーブルを組み合わせて有効なルーティング テーブルを形成した場合、最も限定的なルート（同位のものを含め）がユーザー定義のルーティング テーブルに含められます。システム ルーティング テーブルには、Azure の仮想ネットワーク インターネット ゲートウェイを指すデフォルトルート（0.0.0.0/0）が含まれます。また、システム ルーティング テーブルには、Azureの仮想ネットワーク インフラストラクチャ ゲートウェイを指すネクストホップとともに、他の定義済みのサブネットへの固有ルートが含まれます。

ASAvを介してトラフィックをルーティングするために、ASAv導入プロセスで、ASAvをネクストホップとして使用する他の3つのサブネットへのルートが各サブネットに追加されます。サブネット上の ASAv インターフェイスを指すデフォルトルート（0.0.0.0/0）を追加することもできます。これで、サブネットからのトラフィックはすべて ASAv を介して送信されますが、場合によっては、トラフィックを処理する前に、ASAv ポリシーを設定する必要があります（通常は NAT/PAT を使用）。

システムルーティングテーブル内の既存の限定的なルートのために、ユーザー定義のルーティングテーブルに、ネクストホップとしてASAvを指す限定的なルートを追加する必要があります。追加しないと、ユーザー定義のテーブル内のデフォルトルートではなく、システム ルーティングテーブル内のより限定的なルートが選択され、トラフィックはASAvをバイパスします。

仮想ネットワーク内の VM のルーティング設定

Azure 仮想ネットワーク内のルーティングは、クライアントの特定なゲートウェイ設定ではなく、有効なルーティングテーブルに依存します。仮想ネットワーク内で稼働するクライアントは、DHCPによって、それぞれのサブネット上の 1 アドレスとなるルートを指定されることがあります。これはプレースホルダで、仮想ネットワークのインフラストラクチャ仮想ゲートウェイにパケットを送信するためにだけ使用されます。パケットは、VMから送信されると、有効なルーティングテーブル（ユーザー定義のテーブルによって変更された）に従ってルーティングされます。有効なルーティング テーブルは、クライアントでゲートウェイが 1 として、またはASAvアドレスとして設定されているかどうかに関係なく、ネクストホップを決定します。

Azure VM ARP テーブルには、すべての既知のホストに対して同じ MAC アドレス（1234.5678.9abc）が表示されます。これによって、Azure VMからのすべてのパケットが、有効なルーティングテーブルを使用してパケットのパスを決定するAzureゲートウェイに到達するように保証されます。

（注） ASAvでは、Azureクラウドルーティングの特性により、EIGRPやOSPFなどのダイナミックな内部ルーティングプロトコルを使用できません。有効なルーティング テーブルは、仮想クライアントにスタティック/ダイナミック ルートが設定されているかどうかに関係なく、ネクスト ホップを決定します。

IP アドレス

次の情報は Azure の IP アドレスに適用されます。

- ASAv インターフェイスの IP アドレスを設定するには、DHCP を使用する必要があります。

  Azure インフラストラクチャは、Azure に設定された IP アドレスが確実に ASAv インターフェイスに割り当てられるように動作します。

- Management 0/0 には、それが接続されているサブネット内のプライベート IP アドレスが割り当てられます。

  パブリック IP アドレスは、プライベート IP アドレスに関連付けられる場合があり、Azure インターネット ゲートウェイは NAT 変換を処理します。

- 任意のインターフェイスにパブリック IP アドレスを割り当てることができます。
- ダイナミック パブリック IP アドレスは Azure の停止/開始サイクル中に変更される場合があります。ただし、Azure の再起動時および ASAv のリロード時には、パブリック IP アドレスは保持されます。
- スタティック パブリック IP アドレスは Azure 内でそれらを変更するまで変わりません。

DNS

すべての Azure 仮想ネットワークが、次のように使用できる 168.63.129.16 で、組み込みの DNS サーバーにアクセスできます。

```
configure terminal
dns domain-lookup management
dns server-group DefaultDNS
 name-server 168.63.129.16
end
```

この構成は、Smart Licensing を設定し、専用の DNS サーバーをセットアップしていない場合に使用できます。

Accelerated Networking（AN）

Azure の Accelerated Networking（AN）機能により、VM に対するシングルルート I/O 仮想化（SR-IOV）が可能になります。これにより、VM NIC がハイパーバイザをバイパスしてその下の PCIe カードに直接アクセスできるようになり、ネットワークが高速化します。AN は VM のスループットパフォーマンスを大幅に向上させ、コアの追加（つまり VM の拡大）にも対応します。

AN はデフォルトではディセーブルになっています。Azure は、事前プロビジョニングされた仮想マシンでの AN の有効化をサポートしています。Azure で VM を停止し、ネットワークカー

ドのプロパティを更新して *enableAcceleratedNetworking* パラメータを true に設定するだけです。Microsoft ドキュメントの「既存の VM で高速ネットワークを有効にする」を参照してください。その後、VM を再起動します。

Mellanox ハードウェアのサポート

Microsoft Azure クラウドには、AN 機能をサポートする Mellanox 4（MLX4）と Mellanox 5（MLX5）の 2 種類のハードウェアがあります。ASAv は、リリース 9.15 以降の次のインスタンスに対する Mellanox ハードウェアの AN をサポートしています。

- D3、D3_v2、DS3、DS3_v2
- D4、D4_v2、DS4、DS4_v2
- D5、D5_v2、DS5、DS5_v2
- D8_v3、D8s_v3
- D16_v3、D16s_v3
- F4、F4s
- F8、F8s、F8s_v2
- F16、F16s、F16s_v2

（注） MLX4（Mellanox 4）は connectx3 = cx3 とも呼ばれ、MLX5（Mellanox 5）は connectx4 = cx4 とも呼ばれます。

VM の導入に Azure が使用する NIC（MLX4 または MLX5）は指定できません。シスコでは、高速ネットワーキング機能を使用するために、ASAv 9.15 バージョン以降にアップグレードすることを推奨しています。

Microsoft Azure への ASAv の導入

Microsoft Azure に ASAv を導入できます。

- 標準的な Azure パブリッククラウドおよび Azure Government 環境で、Azure Resource Manager を使用してスタンドアロンファイアウォールとして ASAv を導入します。「Azure Resource Manager からの ASAv の導入」を参照してください。
- Azure Security Center を使用して、Azure 内の統合パートナーソリューションとして ASAv を導入します。セキュリティを重視するお客様には、Azure ワークロードを保護するためのファイアウォールオプションとして ASAv が提供されます。セキュリティイベントとヘルスイベントが単一の統合ダッシュボードからモニターされます。「Azure Security Center からの ASAv の導入」を参照してください。

- Azure Resource Manager を使用して ASAv 高可用性ペアを導入します。冗長性を確保するために、ASAv をアクティブ/バックアップ高可用性（HA）設定で導入できます。パブリッククラウドでの HA では、アクティブな ASAv の障害時に、バックアップ ASAv へのシステムの自動フェールオーバーをトリガーできるステートレスなアクティブ/バックアップソリューションが実装されます。「Azure Resource Manager からの ASAv for High Availability の導入 （123 ページ）」を参照してください。
- VHD（cisco.com から入手可能）から管理対象イメージを使用し、カスタムテンプレートで ASAv または ASAv 高可用性ペアを導入します。シスコでは、圧縮仮想ハードディスク（VHD）を提供しています。この VHD を Azure にアップロードすることで、ASAv の導入プロセスを簡素化できます。管理対象イメージと 2 つの JSON ファイル（テンプレートファイルおよびパラメータファイル）を使用して、単一の協調操作で ASAv のすべてのリソースを導入およびプロビジョニングできます。カスタムテンプレートを使用するには、「VHD およびリソーステンプレートを使用した Azure からの ASAv の導入 （124 ページ）」を参照してください。

Azure Resource Manager からの ASAv の導入

次の手順は、ASAv で Microsoft Azure をセットアップする手順の概略を示しています。Azure の設定の詳細な手順については、『Azure を使ってみる』を参照してください。

Azure に ASAv を導入すると、リソース、パブリック IP アドレス、ルートテーブルなどのさまざまな設定が自動的に生成されます。導入後に、これらの設定をさらに管理できます。たとえば、アイドルタイムアウト値を、デフォルトの短いタイムアウトから変更することができます。

ステップ 1 Azure Resource Manager（ARM）ポータルにログインします。

Azure ポータルは、データセンターの場所に関係なく、現在のアカウントとサブスクリプションに関連付けられた仮想要素を表示します。

ステップ 2 Cisco ASAv のマーケットプレイスを検索し、導入する ASAv をクリックします。

ステップ 3 基本的な設定を行います。

a) 仮想マシンの名前を入力します。この名前は Azure サブスクリプション内で一意である必要があります。

重要 名前が一意でなく、既存の名前を再使用すると、導入に失敗します。

b) ユーザー名を入力します。

c) 認証タイプとして、[パスワード（Password）] または [SSH公開キー（SSH public key）] を選択します。

[パスワード（Password）] を選択した場合は、パスワードを入力して確定します。

d) サブスクリプション タイプを選択します。

e) [Resource group] を選択します。

リソース グループは、仮想ネットワークのリソース グループと同じである必要があります。

f) 場所を選択します。

場所は、ネットワークおよびリソース グループと同じである必要があります。

g) [OK] をクリックします。

ステップ 4 ASAv の設定項目を設定します。

a) 仮想マシンのサイズを選択します。

b) ストレージアカウントを選択します。

既存のストレージアカウントを使用するほか、新規に作成することもできます。ストレージ アカウントの場所はネットワークおよび仮想マシンと同じである必要があります。

c) [Name] フィールドに IP アドレスのラベルを入力し、[OK] をクリックして、パブリック IP アドレスを要求します。

Azure は、VM を停止して再起動すると変更される可能性のある、ダイナミック パブリック IP をデフォルトでは作成します。固定 IP アドレスを優先する場合は、ポータルのパブリック IP を開き、ダイナミック アドレスからスタティック アドレスに変更します。

d) 必要に応じて、DNS のラベルを追加します。

完全修飾ドメイン名は、DNS ラベルと Azure URL の組み合わせで、*<dnslabel>.<location>.cloupapp.azure.com* の形式になります。

e) 既存の仮想ネットワークを選択するか、新しい仮想ネットワークを作成します。

f) ASAv を導入する 4 つのサブネットを設定し、[OK] をクリックします。

重要 各インターフェイスを一意のサブネットにアタッチする必要があります。

g) [OK] をクリックします。

ステップ 5 構成サマリを確認し、[OK] をクリックします。

ステップ 6 利用条件を確認し、[作成（Create）] をクリックします。

次のタスク

- SSH を介して入力できる CLI コマンドを使用するか、または ASDM を使用して、設定を続行します。ASDM にアクセスする手順については、「ASDM の起動」を参照してください。

Azure Security Center からの ASAv の導入

Microsoft Azure Security Center は、お客様がクラウド導入に対するセキュリティ リスクを防御、検出、および軽減できるようにする Azure 向けのセキュリティ ソリューションです。Security Center のダッシュボードから、セキュリティ ポリシーを設定したり、セキュリティ設定をモニターしたり、セキュリティ アラートを表示したりできます。

Security Center は、Azure リソースのセキュリティ状態を分析して、潜在的なセキュリティの脆弱性を特定します。推奨事項のリストに従い、必要なコントロールを設定するプロセスを実行します。対象には、Azure のお客様に対するファイアウォール ソリューションとしての ASAv の導入を含めることができます。

Security Center の統合ソリューションのように、数クリックで ASAv をすばやく導入し、単一のダッシュボードからセキュリティイベントと正常性イベントをモニターできます。次の手順は、Security Center から ASAv を導入する手順の概要です。詳細については、『Azure Security Center』を参照してください。

ステップ 1 Azure ポータルにログインします。

Azure ポータルは、データセンターの場所に関係なく、現在のアカウントとサブスクリプションに関連付けられた仮想要素を表示します。

ステップ 2 Microsoft Azure メニューから、[Security Center] を選択します。

初めて Security Center にアクセスする場合は、[Welcome] ブレードが開きます。**[Yes! I want to Launch Azure Security Center]** を選択して、[Security Center] ブレードを開き、データ収集を有効にします。

ステップ 3 [Security Center] ブレードで、[Policy] タイルを選択します。

ステップ 4 [Security policy] ブレードで、[Prevention policy] を選択します。

ステップ 5 [Prevention policy] ブレードで、セキュリティ ポリシーの一部として表示する推奨事項をオンにします。

a) [Next generation firewall] を [On] に設定します。これで、ASAv が Security Center 内の推奨ソリューションとなります。

b) 必要に応じて、他の推奨事項を設定します。

ステップ 6 [Security Center] ブレードに戻って、[Recommendations] タイルを選択します。

Security Center は、Azure リソースのセキュリティ状態を定期的に分析します。Security Center が潜在的なセキュリティの脆弱性を特定すると、[Recommendations] ブレードに推奨事項が表示されます。

ステップ 7 [Recommendations] ブレードで [Add a Next Generation Firewall] 推奨事項を選択して、詳細を表示したり、問題を解決するためのアクションを実行したりします。

ステップ 8 [新規作成（Create New）] または [既存のソリューションを使用（Use existing solution）] を選択してから、導入する ASAv をクリックします。

ステップ 9 基本的な設定を行います。

a) 仮想マシンの名前を入力します。この名前は Azure サブスクリプション内で一意である必要があります。

重要 名前が一意でなく、既存の名前を再使用すると、導入に失敗します。

b) ユーザー名を入力します。

c) 認証のタイプとして、パスワードまたは SSH キーのいずれかを選択します。

パスワードを選択した場合は、パスワードを入力して確定します。

d) サブスクリプション タイプを選択します。

e) リソース グループを選択します。

リソース グループは、仮想ネットワークのリソース グループと同じである必要があります。

f) 場所を選択します。

場所は、ネットワークおよびリソース グループと同じである必要があります。

g) [OK] をクリックします。

ステップ 10 ASAv の設定項目を設定します。

a) 仮想マシンのサイズを選択します。

ASAv では、Standard D3 および Standard D3_v2 がサポートされます。

b) ストレージアカウントを選択します。

既存のストレージアカウントを使用するほか、新規に作成することもできます。ストレージ アカウントの場所はネットワークおよび仮想マシンと同じである必要があります。

c) [Name] フィールドに IP アドレスのラベルを入力し、[OK] をクリックして、パブリック IP アドレスを要求します。

Azure は、VM を停止して再起動すると変更される可能性のある、ダイナミック パブリック IP をデフォルトでは作成します。固定 IP アドレスを優先する場合は、ポータルのパブリック IP を開き、ダイナミック アドレスからスタティック アドレスに変更します。

d) 必要に応じて、DNS のラベルを追加します。

完全修飾ドメイン名は、DNS ラベルと Azure URL の組み合わせで、*<dnslabel>.<location>.cloupapp.azure.com* の形式になります。

e) 既存の仮想ネットワークを選択するか、新しい仮想ネットワークを作成します。

f) ASAv を導入する 4 つのサブネットを設定し、[OK] をクリックします。

重要 各インターフェイスを一意のサブネットにアタッチする必要があります。

g) [OK] をクリックします。

ステップ 11 構成サマリを確認し、[OK] をクリックします。

ステップ 12 利用条件を確認し、[作成（Create）] をクリックします。

次のタスク

- SSH を介して入力できる CLI コマンドを使用するか、または ASDM を使用して、設定を続行します。ASDM にアクセスする手順については、「ASDM の起動」を参照してください。
- Security Center 内の推奨事項がどのように Azure リソースの保護に役立つかの詳細については、Security Center から入手可能なマニュアルを参照してください。

Azure Resource Manager からの ASAv for High Availability の導入

次の手順は、Microsoft Azure で高可用性（HA）ASAv ペアを設定する手順の概略を示しています。Azure の設定の詳細な手順については、『Azure を使ってみる』を参照してください。

Azure の ASAv HA では、2 つの ASAv を可用性セットに導入し、リソース、パブリック IP アドレス、ルートテーブルなどの各種設定を自動的に生成します。導入後に、これらの設定をさらに管理できます。

ステップ 1 Azure ポータルにログインします。

Azure ポータルは、データセンターの場所に関係なく、現在のアカウントとサブスクリプションに関連付けられた仮想要素を表示します。

ステップ 2 マーケットプレイスで [Cisco ASAv] を検索し、[ASAv 4 NIC HA] をクリックして、フェールオーバー ASAv 構成を導入します。

ステップ 3 [Basics] 設定を構成します。

a) ASAv マシン名のプレフィックスを入力します。ASAv の名前は「プレフィックス」-A と「プレフィックス」-B になります。

重要 既存のプレフィックスを使用していないことを確認します。使用すると、導入は失敗します。

b) ユーザー名を入力します。

これは両方の仮想マシンの管理ユーザー名です。

重要 Azure では、admin というユーザー名は使用できません。

c) 両方の仮想マシンに認証タイプとして、[Password] または [SSH public key] のいずれかを選択します。

[パスワード（Password）] を選択した場合は、パスワードを入力して確定します。

d) サブスクリプション タイプを選択します。

e) [Resource group] を選択します。

[Create new] を選択して新しいリソース グループを作成するか、[Use existing] で既存のリソース グループを選択します。既存のリソース グループを使用する場合は、空である必要があります。そうでない場合は、新しいリソース グループを作成する必要があります。

f) [Location] を選択します。

場所は、ネットワークおよびリソース グループと同じである必要があります。

g) [OK] をクリックします。

ステップ 4 [Cisco ASAv settings] を設定します。

a) 仮想マシンのサイズを選択します。

b) [Managed] または [Unmanaged OS disk] ストレージを選択します。

重要 ASA HA モードでは常に [Managed] を使用します。

ステップ 5 [ASAv-A] 設定を構成します。

a) （オプション）[Create new] を選択して、[Name] フィールドに IP アドレスのラベルを入力し、[OK] をクリックしてパブリック IP アドレスを要求します。パブリック IP アドレスが必要ない場合は、[None] を選択します。

（注） Azure は、VM を停止して再起動すると変更される可能性のある、ダイナミック パブリック IP をデフォルトでは作成します。固定 IP アドレスを優先する場合は、ポータルのパブリック IP を開き、ダイナミック アドレスからスタティック アドレスに変更します。

b) 必要に応じて、DNS のラベルを追加します。

完全修飾ドメイン名は、DNS ラベルと Azure URL の組み合わせで、*<dnslabel>.<location>.cloupapp.azure.com* の形式になります。

c) ASAv-A 起動時診断のストレージ アカウントに必要な設定を構成します。

ステップ 6 [ASAv-B] 設定についても、この手順を繰り返します。

ステップ 7 既存の仮想ネットワークを選択するか、新しい仮想ネットワークを作成します。

a) ASAv を導入する 4 つのサブネットを設定し、[OK] をクリックします。

重要 各インターフェイスを一意のサブネットにアタッチする必要があります。

b) [OK] をクリックします。

ステップ 8 構成の [Summary] を確認し、[OK] をクリックします。

ステップ 9 利用条件を確認し、[作成（Create）] をクリックします。

次のタスク

- SSH を介して入力できる CLI コマンドを使用するか、または ASDM を使用して、設定を続行します。ASDM にアクセスする手順については、「ASDM の起動」を参照してください。
- Azure の ASAv HA 構成の詳細については、ASA コンフィギュレーション ガイド [英語] の「Failover for High Availability in the Public Cloud」の章を参照してください。

VHD およびリソーステンプレートを使用した Azure からの ASAv の導入

シスコが提供する圧縮 VHD イメージを使用して、独自のカスタム ASAv イメージを作成できます。VHD イメージを使用して展開するには、Azure ストレージアカウントに VHD イメージをアップロードする必要があります。次に、アップロードしたディスクイメージおよび Azure Resource Manager テンプレートを使用して、管理対象イメージを作成できます。Azure テンプレートは、リソースの説明とパラメータの定義が含まれている JSON ファイルです。

始める前に

- ASAvテンプレートの展開には、JSONテンプレートおよび対応するJSONパラメータファイルが必要です。テンプレートファイルは、次の GitHub リポジトリからダウンロードできます。

  https://github.com/CiscoDevNet/cisco-asav/tree/master/deployment-templates/azure

- テンプレートとパラメータファイルを構築する手順については、付録：Azure リソーステンプレートの例 （128 ページ）を参照してください。
- この手順では、Azure に Linux VM が存在している必要があります。一時的な Linux VM（Ubuntu 16.04 など）を使用して、Azure に圧縮 VHD イメージをアップロードすることをお勧めします。このイメージを解凍するには、約 50 GB のストレージが必要です。また、Azure の Linux VM から Azure ストレージへのアップロード時間が短くなります。

  VM を作成する必要がある場合は、次のいずれかの方法を使用します。

  - Azure CLI による Linux 仮想マシンの作成
  - Azure ポータルでの Linux 仮想マシンの作成

- Azure サブスクリプションには、ASAv を展開する場所で使用可能なストレージアカウントが必要です。

ステップ 1 https://software.cisco.com/download/home ページから、ASAv 圧縮 VHD イメージをダウンロードします。

a) [Products] > [Security] > [Firewalls] > [Adaptive Security Appliances (ASA)] > [Adaptive Security Appliance (ASA) Software] に移動します。

b) [Adaptive Security Virtual Appliance (ASAv)] をクリックします。

手順に従ってイメージをダウンロードしてください。

たとえば、asav9-14-1.vhd.bz2

ステップ 2 Azure の Linux VM に圧縮 VHD イメージをコピーします。

Azure との間でファイルをやり取りするために使用できるオプションが数多くあります。この例では、SCP（セキュアコピー）を示します。

```
# scp /username@remotehost.com/dir/asav9-14-1.vhd.bz2 <linux-ip>
```

ステップ 3 Azure の Linux VM にログインし、圧縮 VHD イメージをコピーしたディレクトリに移動します。

ステップ 4 ASAv VHD イメージを解凍します。

ファイルを解凍または圧縮解除するために使用できるオプションが数多くあります。この例では Bzip2 ユーティリティを示しますが、Windows ベースのユーティリティも正常に機能します。

```
# bunzip2 asav9-14-1.vhd.bz2
```

ステップ 5 Azure ストレージアカウントのコンテナに VHD をアップロードします。既存のストレージアカウントを使用するほか、新規に作成することもできます。ストレージアカウント名には、小文字と数字のみを使用できます。

ストレージアカウントに VHD をアップロードするために使用できるオプションが数多くあります。AzCopy、Azure Storage Copy Blob API、Azure Storage Explorer、Azure CLI、Azure ポータルなどです。ASAv と同等の大きさのファイルには、Azure ポータルを使用しないことを推奨します。

次の例は、Azure CLI を使用した構文を示しています。

```
azure storage blob upload \
       --file <unzipped vhd> \
       --account-name <azure storage account> \
       --account-key yX7txxxxxxxx1dnQ== \
       --container <container> \
       --blob <desired vhd name in azure> \
       --blobtype page
```

ステップ 6 VHD から管理対象イメージを作成します。

a) Azure ポータルで、[イメージ（Images）] を選択します。

b) [追加（Add）] をクリックして、新しいイメージを作成します。

c) 次の情報を入力します。

- [名前（Name）]：管理対象イメージのユーザー定義の名前を入力します。
- [サブスクリプション（Subscription）]：ドロップダウンリストからサブスクリプションを選択します。
- [リソースグループ（Resource group）]：既存のリソースグループを選択するか、新しいリソースグループを作成します。
- [OSディスク（OS disk）]：OS タイプとして Linux を選択します。
- [ストレージブロブ（Storage blob）]：ストレージアカウントを参照して、アップロードした VHD を選択します。
- [アカウントタイプ（Account type）]：ドロップダウンリストから [標準（HDD）（Standard (HDD)）] を選択します。
- [ホストキャッシング（Host caching）]：ドロップダウンリストから [読み取り/書き込み（Read/write）] を選択します。
- [データディスク（Data disks）]：デフォルトのままにしておきます。データディスクを追加しないでください。

d) [作成（Create）] をクリックします。

「イメージが正常に作成されました（Successfully created image）」というメッセージが [通知（Notifications）] タブの下に表示されるまで待ちます。

（注） 管理対象イメージが作成されたら、アップロードした VHD とアップロード ストレージ アカウントを削除できます。

ステップ 7 新規に作成した管理対象イメージのリソース ID を取得します。

Azure の内部では、あらゆるリソースがリソース ID に関連付けられています。リソース ID は、この管理対象イメージから新しい ASAv ファイアウォールを展開するときに必要になります。

a) Azure ポータルで、[イメージ（Images）] を選択します。
b) 前のステップで作成した管理対象イメージを選択します。
c) [概要（Overview）] をクリックして、イメージのプロパティを表示します。
d) クリップボードに**リソース ID** をコピーします。

リソース ID は、次の形式を取ります。

/subscriptions/<subscription-id>/resourceGroups/<resourceGroup>
/providers/Microsoft.Compute/<container>/ <vhdname>

ステップ 8 管理対象イメージおよびリソーステンプレートを使用して、ASAv ファイアウォールを構築します。

a) [新規（New）] を選択し、オプションから選択できるようになるまで [テンプレート展開（Template Deployment）] を検索します。
b) [作成（Create）] を選択します。
c) [エディタで独自のテンプレートを構築する（Build your own template in the editor）] を選択します。

カスタマイズできる空白のテンプレートが作成されます。テンプレートを作成する方法の例については、リソース テンプレートの作成 （129 ページ）を参照してください。

d) カスタマイズした JSON テンプレートコードをウィンドウに貼り付け、[保存（Save）] をクリックします。
e) ドロップダウンリストから [サブスクリプション（Subscription）] を選択します。
f) 既存の [リソースグループ（Resource group）] を選択するか、新しいリソースグループを作成します。
g) ドロップダウンリストから [ロケーション（Location）] を選択します。
h) 前ステップからの管理対象イメージの [リソースID（Resource ID）] を [VM管理対象イメージID（Vm Managed Image Id）] フィールドに貼り付けます。

ステップ 9 [カスタム展開（Custom deployment）] ページの最上部にある [パラメータの編集（Edit parameters）] をクリックします。カスタマイズできるパラメータテンプレートが作成されます。

a) [ファイルのロード（Load file）] をクリックし、カスタマイズした ASAv パラメータファイルを参照します。パラメータテンプレートを作成する例については、「パラメータ ファイルの作成 （138 ページ）」を参照してください。
b) カスタマイズした JSON パラメータコードをウィンドウに貼り付け、[保存（Save）] をクリックします。

ステップ 10 カスタム展開の詳細を確認します。[基本（Basics）] と [設定（Settings）] の情報（[リソースID（Resource ID）] など）が、想定した展開設定に一致することを確認します。

ステップ 11 利用規約を確認し、[上記の利用規約に同意します（I agree to the terms and conditions stated above）] チェックボックスをオンにします。

ステップ 12 [購入（Purchase）] をクリックし、管理対象イメージおよびカスタムテンプレートを使用して ASAv ファイアウォールを導入します。

テンプレートファイルとパラメータファイルに競合がなければ、展開が正常に完了しているはずです。

管理対象イメージは、同じサブスクリプションおよび地域内の複数の展開に使用できます。

次のタスク

- SSH を介して入力できる CLI コマンドを使用するか、または ASDM を使用して、設定を続行します。ASDM にアクセスする手順については、「ASDM の起動」を参照してください。

付録：Azure リソース テンプレートの例

この項では、ASAv を展開するために使用できる Azure Resource Manager テンプレートの構造について説明します。Azure リソーステンプレートは JSON ファイルです。必須の全リソースの展開を簡素化するため、この例には 2 つの JSON ファイルが含まれています。

- **テンプレートファイル**：これは、リソースグループ内のすべてのコンポーネントを展開するメインリソースファイルです。
- **パラメータ ファイル**：このファイルには、ASAv を正常に展開するために必要なパラメータが含まれています。このファイルには、サブネット情報、仮想マシンの階層とサイズ、ASAv のユーザー名とパスワード、ストレージコンテナの名前など、詳細な情報が含まれています。このファイルは Azure Stack Hub 展開環境用にカスタマイズできます。

テンプレート ファイルの形式

この項では、Azure Resource Manager テンプレートの構造について説明します。次の例は、テンプレートファイルを縮小表示したもので、テンプレートのさまざまな部分を示しています。

Azure Resource Manager JSON テンプレート ファイル

```
{
    "$schema":
"http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "",
    "parameters": { },
    "variables": { },
    "resources": [ ],
    "outputs": { }
}
```

テンプレートは、ASAv 展開の値を作成するために使用できる JSON および式で構成されています。その最も単純な構造では、テンプレートに次の要素が含まれています。

表 *17*：定義済みの *Azure Resource Manager JSON* テンプレートファイル要素

要素	必須	説明
$schema	はい	テンプレート言語のバージョンを説明する JSON スキーマファイルの場所。前の図に示した URL を使用します。

要素	必須	説明
contentVersion	はい	テンプレートのバージョン（1.0.0.0 など）。この要素には任意の値を指定できます。テンプレートを使用してリソースを展開するときは、この値を使用して、適切なテンプレートが使用されていることを確認できます。
parameters	いいえ	展開を実行してリソース展開をカスタマイズするときに指定する値。パラメータにより、展開時に値を入力できます。絶対に必要というわけではありませんが、指定しないと、JSONテンプレートは毎回同じパラメータでリソースを展開します。
variables	いいえ	テンプレート言語式を簡素化するためにテンプレートで JSON フラグメントとして使用される値。
resources	はい	リソースグループで展開または更新されるリソースタイプ。
outputs	いいえ	展開後に返される値。

JSONテンプレートは、展開するリソース タイプを宣言するためだけでなく、その関連する設定パラメータを宣言するためにも利用できます。次の例は、新しい ASAv を展開するテンプレートを示しています。

リソース テンプレートの作成

以下の例を使用して、テキストエディタを使用した独自の導入テンプレートを作成できます。

ステップ 1 次の例に示したテキストをコピーします。

例：

```
{
    "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "vmName": {
            "type": "string",
            "defaultValue": "ngfw",
            "metadata": {
                "description": "Name of the NGFW VM"
            }
        },
        "vmManagedImageId": {
            "type": "string",
            "defaultValue":
"/subscriptions/{subscription-id}/resourceGroups/myresourcegroup1/providers/Microsoft.Compute/images/myImage",

            "metadata": {
```

```
                "description": "The ID of the managed image used for deployment.
/subscriptions/{subscription-id}/resourceGroups/myresourcegroup1/providers/Microsoft.Compute/images/myImage"

            }
        },
        "adminUsername": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "Username for the Virtual Machine. admin, Administrator among other
 values are disallowed - see Azure docs"
            }
        },
        "adminPassword": {
            "type": "securestring",
            "defaultValue" : "",
            "metadata": {
                "description": "Password for the Virtual Machine. Passwords must be 12 to 72 chars
 and have at least 3 of the following: Lowercase, uppercase, numbers, special chars"
            }
        },
        "vmStorageAccount": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "A storage account name (boot diags require a storage account).
Between 3 and 24 characters. Lowercase letters and numbers only"
            }
        },
        "virtualNetworkResourceGroup": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "Name of the virtual network's Resource Group"
            }
        },
        "virtualNetworkName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "Name of the virtual network"
            }
        },
        "mgmtSubnetName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The FTDv management interface will attach to this subnet"
            }
        },
        "mgmtSubnetIP": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "NGFW IP on the mgmt interface (example: 192.168.0.10)"
            }
        },
        "diagSubnetName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The FTDv diagnostic0/0 interface will attach to this subnet"
            }
        },
```

```
            "diagSubnetIP": {
                "type": "string",
                "defaultValue": "",
                "metadata": {
                    "description": "NGFW IP on the diag interface (example: 192.168.1.10)"
                }
            },
            "gig00SubnetName": {
                "type": "string",
                "defaultValue": "",
                "metadata": {
                    "description": "The FTDv Gigabit 0/0 interface will attach to this subnet"
                }
            },
            "gig00SubnetIP": {
                "type": "string",
                "defaultValue": "",
                "metadata": {
                    "description": "The IP on the Gigabit 0/0 interface (example: 192.168.2.10)"
                }
            },
            "gig01SubnetName": {
                "type": "string",
                "defaultValue": "",
                "metadata": {
                    "description": "The FTDv Gigabit 0/1 interface will attach to this subnet"
                }
            },
            "gig01SubnetIP": {
                "type": "string",
                "defaultValue": "",
                "metadata": {
                    "description": "The IP on the Gigabit 0/1 interface (example: 192.168.3.5)"
                }
            },
            "VmSize": {
                "type": "string",
                "defaultValue": "Standard_D3_v2",
                "allowedValues": [ "Standard_D3_v2" , "Standard_D3" ],
                "metadata": {
                    "description": "NGFW VM Size (Standard_D3_v2 or Standard_D3)"
                }
            }
        },
        "variables": {

            "virtualNetworkID":
"[resourceId(parameters('virtualNetworkResourceGroup'),'Microsoft.Network/virtualNetworks',
parameters('virtualNetworkName'))]",

            "vmNic0Name":"[concat(parameters('vmName'),'-nic0')]",
            "vmNic1Name":"[concat(parameters('vmName'),'-nic1')]",
            "vmNic2Name":"[concat(parameters('vmName'),'-nic2')]",
            "vmNic3Name":"[concat(parameters('vmName'),'-nic3')]",

            "vmNic0NsgName":"[concat(variables('vmNic0Name'),'-NSG')]",

            "vmMgmtPublicIPAddressName": "[concat(parameters('vmName'),'nic0-ip')]",
            "vmMgmtPublicIPAddressType": "Static",
            "vmMgmtPublicIPAddressDnsName": "[variables('vmMgmtPublicIPAddressName')]"
        },
        "resources": [
            {
                "apiVersion": "2017-03-01",
```

```
            "type": "Microsoft.Network/publicIPAddresses",
            "name": "[variables('vmMgmtPublicIPAddressName')]",
            "location": "[resourceGroup().location]",
            "properties": {
              "publicIPAllocationMethod": "[variables('vmMgmtPublicIpAddressType')]",
              "dnsSettings": {
                "domainNameLabel": "[variables('vmMgmtPublicIPAddressDnsName')]"
              }
            }
        },
        {
            "apiVersion": "2015-06-15",
            "type": "Microsoft.Network/networkSecurityGroups",
            "name": "[variables('vmNic0NsgName')]",
            "location": "[resourceGroup().location]",
            "properties": {
                "securityRules": [
                    {
                        "name": "SSH-Rule",
                        "properties": {
                            "description": "Allow SSH",
                            "protocol": "Tcp",
                            "sourcePortRange": "*",
                            "destinationPortRange": "22",
                            "sourceAddressPrefix": "Internet",
                            "destinationAddressPrefix": "*",
                            "access": "Allow",
                            "priority": 100,
                            "direction": "Inbound"
                        }
                    },
                    {
                        "name": "SFtunnel-Rule",
                        "properties": {
                            "description": "Allow tcp 8305",
                            "protocol": "Tcp",
                            "sourcePortRange": "*",
                            "destinationPortRange": "8305",
                            "sourceAddressPrefix": "Internet",
                            "destinationAddressPrefix": "*",
                            "access": "Allow",
                            "priority": 101,
                            "direction": "Inbound"
                        }
                    }
                ]
            }
        },
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('vmNic0Name')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
                "[concat('Microsoft.Network/networkSecurityGroups/',variables('vmNic0NsgName'))]",
                "[concat('Microsoft.Network/publicIPAddresses/',
variables('vmMgmtPublicIPAddressName'))]"
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Static",
```

```
                            "privateIPAddress" : "[parameters('mgmtSubnetIP')]",
                            "subnet": {
                                "id": "[concat(variables('virtualNetworkID'),'/subnets/',
parameters('mgmtSubnetName'))]"
                            },
                            "publicIPAddress":{
                                "id": "[resourceId('Microsoft.Network/publicIPAddresses/',
variables('vmMgmtPublicIPAddressName'))]"
                            }
                        }
                    }
                ],
                "networkSecurityGroup": {
                    "id": "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('vmNic0NsgName'))]"
                },
                "enableIPForwarding": true
            }
        },
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('vmNic1Name')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Static",
                            "privateIPAddress" : "[parameters('diagSubnetIP')]",
                            "subnet": {
                                "id": "[concat(variables('virtualNetworkID'),'/subnets/',
parameters('diagSubnetName'))]"
                            }                        }
                    }
                ],
                "enableIPForwarding": true
            }
        },
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('vmNic2Name')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Static",
                            "privateIPAddress" : "[parameters('gig00SubnetIP')]",
                            "subnet": {
                                "id": "[concat(variables('virtualNetworkID'),'/subnets/',
parameters('gig00SubnetName'))]"
                            }                        }
                    }
                ],
                "enableIPForwarding": true
            }
```

```
        },
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('vmNic3Name')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Static",
                            "privateIPAddress" : "[parameters('gig01SubnetIP')]",
                            "subnet": {
                                "id": "[concat(variables('virtualNetworkID'),'/subnets/',
parameters('gig01SubnetName'))]"

                            }                       }
                    }
                ],
                "enableIPForwarding": true
            }
        },
        {
            "type": "Microsoft.Storage/storageAccounts",
            "name": "[concat(parameters('vmStorageAccount'))]",
            "apiVersion": "2015-06-15",
            "location": "[resourceGroup().location]",
            "properties": {
              "accountType": "Standard_LRS"
            }
        },
        {
            "apiVersion": "2017-12-01",
            "type": "Microsoft.Compute/virtualMachines",
            "name": "[parameters('vmName')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
                "[concat('Microsoft.Storage/storageAccounts/', parameters('vmStorageAccount'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic0Name'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic1Name'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic2Name'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic3Name'))]"
            ],
            "properties": {
                "hardwareProfile": {
                    "vmSize": "[parameters('vmSize')]"
                },
                "osProfile": {
                    "computername": "[parameters('vmName')]",
                    "adminUsername": "[parameters('AdminUsername')]",
                    "adminPassword": "[parameters('AdminPassword')]"
                },
                "storageProfile": {
                    "imageReference": {
                        "id": "[parameters('vmManagedImageId')]"
                    },
                    "osDisk": {
                        "osType": "Linux",
                        "caching": "ReadWrite",
                        "createOption": "FromImage"
                    }
```

```
                },
                "networkProfile": {
                    "networkInterfaces": [
                        {
                            "properties": {
                                "primary": true
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic0Name'))]"
                        },
                        {
                            "properties": {
                                "primary": false
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic1Name'))]"
                        },
                        {
                            "properties": {
                                "primary": false
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic2Name'))]"
                        },
                        {
                            "properties": {
                                "primary": false
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic3Name'))]"
                        }
                    ]
                },
                "diagnosticsProfile": {
                    "bootDiagnostics": {
                        "enabled": true,
                        "storageUri":
"[concat('http://',parameters('vmStorageAccount'),'.blob.core.windows.net')]"
                    }
                }
            }
        }
    ],
    "outputs": {  }
}
```

ステップ 2 このファイルを、たとえば **azureDeploy.json** というように、JSON ファイルとしてローカルに保存します。

ステップ 3 ファイルを編集し、導入パラメータに合うテンプレートを作成します。

ステップ 4 VHD およびリソーステンプレートを使用した Azure からの ASAv の導入 （124 ページ）で説明しているように、このテンプレートを使用して ASAv を展開します。

パラメータファイルの形式

新しい展開を開始するときには、リソーステンプレートにパラメータが定義されています。展開を開始するには、これらを入力しておく必要があります。リソーステンプレートに定義したパラメータを手動で入力することも、パラメータをテンプレートパラメータ JSON ファイルに配置しておくこともできます。

パラメータ ファイルには、パラメータ ファイルの作成（138ページ）のパラメータの例に表示される各パラメータの値が含まれています。これらの値は、展開時にテンプレートに自動的に渡されます。さまざまな展開シナリオに合わせて複数のパラメータ ファイルを作成できます。

この例のASAvテンプレートの場合、パラメータ ファイルに次のパラメータを定義する必要があります。

表 *18 : ASAv* パラメータの定義

フィールド	説明	例
vmName	Azure における ASAv マシンの名前。	cisco-asav
vmManagedImageId	展開に使用される管理対象イメージの ID。Azure の内部では、あらゆるリソースがリソース ID に関連付けられています。	/subscriptions/73d2537e-ca44-46aa-beb2-74ff1dd61b41/resourceGroups/ewManagedImages-rg/providers/Microsoft.Compute/images/ASAv910-Managed-Image
adminUsername	ASAv にログインするためのユーザー名。予約名の「admin」にすることはできません。	jdoe
adminPassword	管理者アカウントのパスワード。これは、12 ～ 72 文字の長さで、1 つの小文字、1 つの大文字、1 つの数字、1 つの特殊文字のうち 3 つを含める必要があります。	Pw0987654321
vmStorageAccount	Azure ストレージアカウント。既存のストレージアカウントを使用するほか、新規に作成することもできます。ストレージアカウント名は、3 ～ 24 文字で、小文字と数字のみ含めることができます。	ciscoasavstorage
virtualNetworkResourceGroup	仮想ネットワークのリソースグループの名前。ASAv は常に新しいリソースグループに配置されます。	ew-west8-rg
virtualNetworkName	仮想ネットワークの名前。	ew-west8-vnet

フィールド	説明	例
mgmtSubnetName	管理インターフェイスは、このサブネットに接続されます。これは、Nic0（最初のサブネット）にマップされます。既存のネットワークに参加する場合、これは既存のサブネット名に一致する必要があります。	mgmt
mgmtSubnetIP	管理インターフェイス IP アドレス。	10.8.0.55
gig00SubnetName	GigabitEthernet 0/0 インターフェイスは、このサブネットに接続されます。これは、Nic1（2番目のサブネット）にマップされます。既存のネットワークに参加する場合、これは既存のサブネット名に一致する必要があります。	inside
gig00SubnetIP	GigabitEthernet 0/0 インターフェイス IP アドレス。これは、ASAv の最初のデータインターフェイス用です。	10.8.2.55
gig01SubnetName	GigabitEthernet 0/1 インターフェイスは、このサブネットに接続されます。これは、Nic2（3番目のサブネット）にマップされます。既存のネットワークに参加する場合、これは既存のサブネット名に一致する必要があります。	outside
gig01SubnetIP	GigabitEthernet 0/1 インターフェイス IP アドレス。これは、ASAv の2番目のデータインターフェイス用です。	10.8.3.55

フィールド	説明	例
gig02SubnetName	GigabitEthernet 0/2 インターフェイスは、このサブネットに接続されます。これは、Nic3（4番目のサブネット）にマップされます。既存のネットワークに参加する場合、これは既存のサブネット名に一致する必要があります。	dmz
gig02SubnetIP	GigabitEthernet 0/2 インターフェイスの IP アドレス。これは、ASAv の 3 番目のデータインターフェイス用です。	10.8.4.55
vmSize	ASAv VM に使用する VM のサイズ。Standard_D3_V2 と Standard_D3 がサポートされています。Standard_D3_V2 がデフォルトです。	Standard_D3_V2 または Standard_D3

パラメータ ファイルの作成

以下の例を使用して、テキスト エディタを使用した独自のパラメータ ファイルを作成できます。

（注） 次の例は、IPV4 専用です。

ステップ 1 次の例に示したテキストをコピーします。

例：

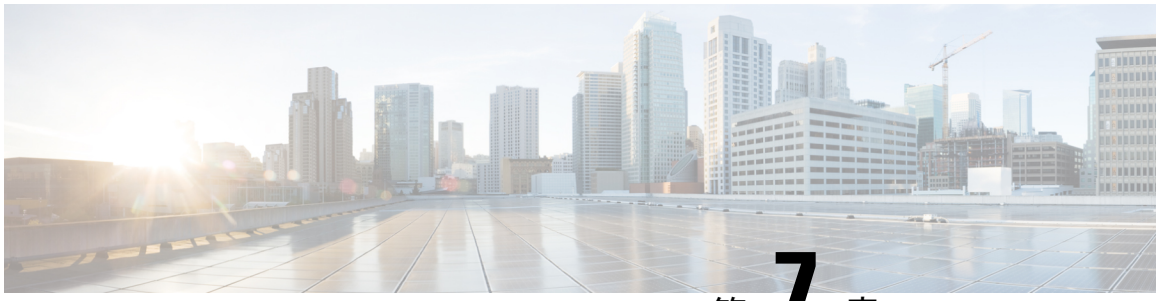
```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmName": {
      "value": "cisco-asav1"
    },
    "vmManagedImageId": {
      "value":
"/subscriptions/33d2517e-ca88-46aa-beb2-74ff1dd61b41/resourceGroups/ewManagedImages-rg/providers/Microsoft.Compute/images/ASAv-9.10.1-81-Managed-Image"

    },
    "adminUsername": {
      "value": "jdoe"
```

```
    },
    "adminPassword": {
      "value": "Pw0987654321"
    },
    "vmStorageAccount": {
      "value": "ciscoasavstorage"
    },
    "virtualNetworkResourceGroup": {
      "value": "ew-west8-rg"
    },
    "virtualNetworkName": {
      "value": "ew-west8-vn"
    },
    "mgmtSubnetName": {
      "value": "mgmt"
    },
    "mgmtSubnetIP": {
      "value": "10.8.3.77"
    },
    "gig00SubnetName": {
      "value": "inside"
    },
    "gig00SubnetIP": {
      "value": "10.8.2.77"
    },
    "gig01SubnetName": {
      "value": "outside"
    },
    "gig01SubnetIP": {
      "value": "10.8.1.77"
    },
    "gig02SubnetName": {
      "value": "dmz"
    },
    "gig02SubnetIP": {
      "value": "10.8.0.77"
    },
    "VmSize": {
      "value": "Standard_D3_v2"
    }
  }
}
```

ステップ 2 このファイルを、たとえば **azureParameters.json** というように、JSON ファイルとしてローカルに保存します。

ステップ 3 ファイルを編集し、導入パラメータに合うテンプレートを作成します。

ステップ 4 VHD およびリソーステンプレートを使用した Azure からの ASAv の導入 （124 ページ）で説明しているように、このパラメータ テンプレートを使用して ASAv を展開します。



第 7 章

Microsoft Azure への ASAv Auto Scale ソリューションの導入

Azure での ASAv の Auto Scale ソリューション

Auto Scale ソリューションについて

ASAv Auto Scale for Azure は、Azure が提供するサーバーレス インフラストラクチャ（Logic App、Azure 関数、ロードバランサ、セキュリティグループ、仮想マシンスケールセットなど）を使用する完全なサーバーレス導入です。

ASAv Auto Scale for Azure 導入の主な特徴は次のとおりです。

- Azure Resource Manager（ARM）テンプレートベースの展開。
- CPU および に基づくスケーリングメトリックのサポート：

（注） 詳細については、「Auto Scale ロジック （168 ページ）」を参照してください。

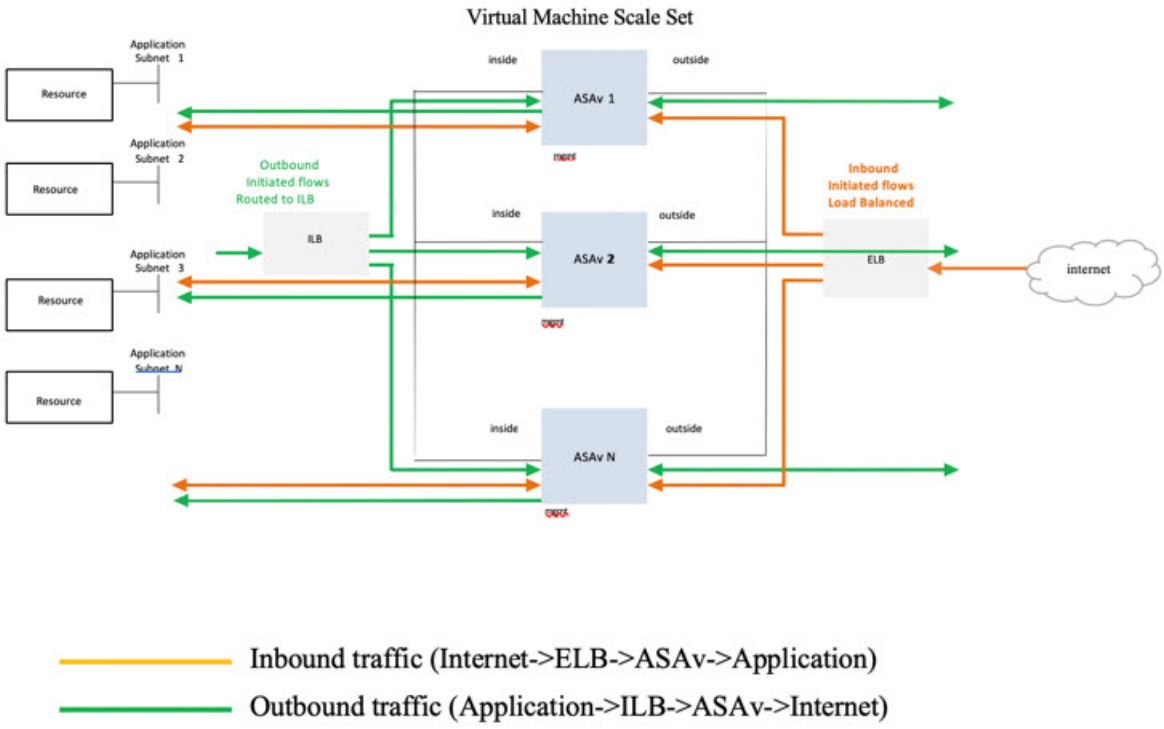
- ASAv 展開とマルチ可用性ゾーンのサポート。
- スケールアウトされた ASAv インスタンスに完全に自動化された構成を自動適用。
- ロードバランサとマルチ可用性ゾーンのサポート。
- Auto Scale 機能の有効化と無効化をサポート。
- シスコでは、導入を容易にするために、Auto Scale for Azure 導入パッケージを提供しています。

Auto Scale の導入例

ASAv Auto Scale for Azure は、ASAv スケールセットを Azure の内部ロードバランサ（ILB）と Azure の外部ロードバランサ（ELB）の間に配置する自動水平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スケールセット内の ASAv インスタンスに分散させます。その後、ファイアウォール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アウトバウンド インターネット トラフィックをスケールセット内の ASAv インスタンスに分散させます。その後、ファイアウォール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 ASAv 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 *16 : ASAv Auto Scale* の導入例



スコープ

このドキュメントでは、ASAv Auto Scale for Azure ソリューションと、 のサーバーレスコンポーネントを展開する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

ASAv Auto Scale for Azure ソリューションは、Azure が提供するサーバーレス インフラストラクチャ（Logic App、Azure 関数、ロードバランサ、仮想マシンスケールセットなど）を使用する Azure Resource Manager（ARM）テンプレートベースの展開です。

ASAv Auto Scale for Azure 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。

ASM_Function.zip パッケージの作成方法については、「ソースコードからの Azure 関数の構築 （171 ページ）」を参照してください。

Auto Scale ソリューションのコンポーネント

ASAv Auto Scale for Azure 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。

Azure 関数（Function App）

Function App とは一連の Azure 関数です。基本的な機能は次のとおりです。

- Azure メトリックを定期的に通信またはプローブします。
- ASAv の負荷をモニターし、スケールイン/スケールアウト操作をトリガーします。

関数は、圧縮された Zip パッケージの形式で提供されます（「Azure Function App パッケージの構築 （148 ページ）」を参照）。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。

Orchestrator（Logic App）

Auto Scale Logic App は、ワークフロー、つまり一連のステップの集合です。Azure 関数は独立したエンティティであり、相互に通信できません。この Orchestrator は、関数の実行を順序付けし、関数間で情報を交換します。

- Logic App は、Auto Scale Azure 関数間で情報をオーケストレーションおよび受け渡すために使用されます。
- 各ステップは、Auto Scale Azure 関数または組み込みの標準ロジックを表します。
- Logic App は JSON ファイルとして提供されます。
- Logic App は、GUI または JSON ファイルを使用してカスタマイズできます。

仮想マシンスケールセット（VMSS）

VMSS は、ASAv デバイスなどの同種の仮想マシンの集合です。

- VMSS では、新しい同一の VM をセットに追加できます。
- VMSS に追加された新しい VM は、ロードバランサ、セキュリティグループ、およびネットワーク インターフェイスに自動的に接続されます。
- VMSS には組み込みの Auto Scale 機能があり、ASAv for Azure では無効になっています。
- VMSS で ASAv インスタンスを手動で追加したり、削除したりしないでください。

Azure Resource Manager（ARM）テンプレート

ARM テンプレートは、ASAv Auto Scale for Azure 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。

ARM テンプレートは、以下を含む Auto Scale Manager コンポーネントへの入力情報を提供します。

- Azure Function App
- Azure Logic App
- 仮想マシンスケールセット（VMSS）
- 内部および外部ロードバランサ。
- 展開に必要なセキュリティグループおよびその他のコンポーネント。

重要 ユーザー入力の検証に関しては、ARM テンプレートには限界があるため、展開時に入力を検証する必要があります。

Auto Scale ソリューションの前提条件

Azure のリソース

リソース グループ

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リソースグループが必要です。

（注） 後で使用するために、リソースグループ名、リソースグループが作成されたリージョン、および Azure サブスクリプション ID を記録します。

ネットワーキング

仮想ネットワークが使用可能または作成済みであ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。

ASAv には 3 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 3 つのサブネットが必要です。

1. 管理トラフィック
2. 内部トラフィック
3. 外部トラフィック

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 ASAv 間の正常性プローブに必要です。

  サーバーレス機能と ASAv 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）。

（注） 仮想ネットワーク名、仮想ネットワーク CIDR、3 つすべてのサブネットの名前、および外部と内部のサブネットのゲートウェイ IP アドレスを記録します。

ASA 構成ファイルの準備

ASAv 構成ファイルを準備し、ASAv 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 ASAv により、このファイルがダウンロードされて構成が更新されます。

ASA 構成ファイルでは、（少なくとも）次のことが必要になります。

- すべてのインターフェイスに DHCP IP 割り当てを設定します。
- GigabitEthernet0/1 は「内部」インターフェイスである必要があります。
- GigabitEthernet0/0 は「外部」インターフェイスである必要があります。
- ゲートウェイを内部インターフェイスと外部インターフェイスに設定します。
- 内部インターフェイスと外部インターフェイスで Azure ユーティリティ IP からの SSH を有効にします（ヘルスプローブ用）。
- 外部インターフェイスから内部インターフェイスにトラフィックを転送するための NAT 構成を作成します。

- 目的のトラフィックを許可するアクセスポリシーを作成します。
- 構成のライセンスを取得します。PAYG 課金はサポートされていません。

（注） 管理インターフェイスを特別に設定する必要はありません。

以下は、 の ASA 構成ファイルのサンプルです。

```
ASA Version 9.13(1)
!
interface GigabitEthernet0/1
nameif inside
security-level 100
ip address dhcp setroute
!
interface GigabitEthernet0/0
nameif outside
security-level 0
ip address dhcp setroute
!
route outside 0.0.0.0 0.0.0.0 10.12.3.1 2
!
route inside 0.0.0.0 0.0.0.0 10.12.2.1 3
!
ssh 168.63.129.0 255.255.255.0 outside
!
ssh 168.63.129.0 255.255.255.0 inside
!
object network webserver
host 10.12.2.5
object service myport
service tcp source range 1 65535 destination range 1 65535
access-list outowebaccess extended permit object myport any any log disable
access-group outowebaccess in interface outside
object service app
service tcp source eq www
nat (inside,outside) source static webserver interface destination static interface any
 service app app
object network obj-any
subnet 0.0.0.0 0.0.0.0
nat (inside,outside) source dynamic obj-any interface destination static obj-any obj-any
configure terminal
dns domain-lookup management
policy-map global_policy
class inspection_default
inspect icmp
call-home
profile License
destination transport-method http
destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService
license smart
feature tier standard
throughput level 2G
license smart register idtoken <TOKEN>
: end
```

Azure Function App パッケージの構築

ASAv Auto Scale ソリューションでは、*ASM_Function.zip* アーカイブファイルを作成する必要があります。このファイルから、圧縮された ZIP パッケージの形式で一連の個別の Azure 関数が提供されます。

ASM_Function.zip パッケージの作成方法については、「ソースコードからの Azure 関数の構築（171 ページ）」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、Azure サブスクリプションに ARM テンプレートを展開するときに、各パラメータを使用して ASAv デバイスを作成できます。「Auto Scale ARM テンプレートの展開（153 ページ）」を参照してください。

表 *19*：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	リソースの作成タイプ
resourceNamePrefix	文字列*（3 ～ 10 文字）	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。 注：小文字のみを使用してください。 例：asav	新規作成
virtualNetworkRg	文字列	仮想ネットワークのリソースグループの名前。 例：cisco-virtualnet-rg	既存
virtualNetworkName	文字列	仮想ネットワーク名（作成済み） 例：cisco-virtualnet	既存
mgmtSubnet	文字列	管理サブネット名（作成済み） 例：cisco-mgmt-subnet	既存

パラメータ名	使用できる値/タイプ	説明	リソースの作成タイプ
insideSubnet	文字列	内部サブネット名（作成済み） 例：cisco-inside-subnet	既存
internalLbIp	文字列	内部サブネットの内部ロードバランサの IP アドレス（作成済み）。 例：1.2.3.4	既存
outsideSubnet	文字列	外部サブネット名（作成済み） 例：cisco-outside-subnet	既存
softwareVersion	文字列	ASAvバージョン（展開時にドロップダウンから選択） デフォルト：914.1.0許可：914.1.0, 913.1.0	既存
vmSize	文字列	ASAv インスタンスのサイズ（展開時にドロップダウンから選択）	該当なし
asaAdminUserName	文字列 *	ASAv「admin」ユーザーのユーザー名。 パスワードの長さは 12 ー 72 文字で、小文字、大文字、数字、特殊文字を使用する必要があります。また、文字の繰り返しは 2 回までにする必要があります。 これは「admin」には**できません**。VM 管理者ユーザー名のガイドラインについては、「Azure」を参照してください。 （注） テンプレートには、このパラメータのコンプライアンスチェック機能はありません。	新規作成

パラメータ名	使用できる値/タイプ	説明	リソースの作成タイプ
asaAdminUserPassword	文字列 *	ASAv 管理者ユーザのパスワード。 パスワードの長さは 12 ー 72 文字で、小文字、大文字、数字、特殊文字を使用する必要があります。また、文字の繰り返しは 2 回までにする必要があります。 （注） テンプレートには、このパラメータのコンプライアンスチェック機能はありません。	新規作成
scalingPolicy	POLICY-1/POLICY-2	**POLICY-1**：設定された期間に、いずれかの ASAv の平均負荷がスケールアウトしきい値を超えるとスケールアウトがトリガーされます。 **POLICY-2**：設定された期間に、Auto Scale グループ内のすべての ASAv デバイスの平均負荷がスケールアウトしきい値を超えるとスケールアウトがトリガーされます。 どちらの場合も、スケールインロジックは同じままです。設定された期間に、すべての ASAv デバイスの平均負荷がスケールインしきい値を下回るとスケールインがトリガーされます。	該当なし
scalingMetricsList	文字列	スケーリングの決定に使用されるメトリック。 許可：CPU デフォルト：CPU	該当なし

パラメータ名	使用できる値/タイプ	説明	リソースの作成タイプ
scaleInThreshold	文字列	スケールインしきい値（パーセント単位）。 デフォルト：10 ASAvメトリック（CPU 使用率）がこの値を下回ると、スケールインがトリガーされます。 「Auto Scale ロジック （168 ページ）」を参照してください。	該当なし
scaleOutThreshold	文字列	スケールアウトしきい値（パーセント単位）。 デフォルト：80 ASAvメトリック（CPU 使用率）がこの値を上回ると、スケールアウトがトリガーされます。 「scaleOutThreshold」は、常に「scaleInThreshold」より**大きく**する必要があります。 「Auto Scale ロジック （168 ページ）」を参照してください。	該当なし
minAsaCount	整数	任意の時点でスケールセットで使用可能な最小 ASAv インスタンス数。 例：2。	該当なし
maxAsaCount	整数	スケールセットで許可される最大 ASAv インスタンス数。 例：10 （注） Auto Scale ロジックではこの変数の範囲はチェックされないため、慎重に入力してください。	該当なし

パラメータ名	使用できる値/タイプ	説明	リソースの作成タイプ
metricsAverageDuration	整数	ドロップダウンから選択します。 この数値は、メトリックが平均化される時間（分単位）を表します。 この変数の値が5（5分）の場合、Auto Scale Managerがスケジュールされると、メトリックの過去5分間の平均がチェックされ、その結果に基づいてスケーリングの判断が行われます。 （注） Azureの制限により、有効な数値は1、5、15、および30だけです。	該当なし
initDeploymentMode	BULK/STEP	主に最初の展開、またはスケールセットにASAvインスタンスが含まれていない場合に適用されます。 BULK：Auto Scale Managerは、「minAsaCount」個のASAvインスタンスを同時に展開しようとします。 STEP：Auto Scale Managerは、スケジュールされた間隔ごとに「minAsaCount」個のASAvデバイスを1つずつ展開します。	
configurationFile	文字列	ASAv構成ファイルのファイルパス。 例： https://myserver/asavconfig/asaconfig.txt	該当なし
*Azureには、新しいリソースの命名規則に関する制限があります。制限を確認するか、またはすべて小文字を使用してください**スペースやその他の特殊文字は使用しないでください**。			

Auto Scale の展開

Auto Scale ARM テンプレートの展開

：ARM テンプレートを使用して、Azure 用 ASAv Auto Scale に必要なリソースを展開します。特定のリソースグループ内では、ARM テンプレートを展開することで次の内容が作成されます。

- 仮想マシンスケールセット（VMSS）
- 外部ロードバランサ
- 内部ロードバランサ
- Azure Function App
- Logic App
- セキュリティグループ（データインターフェイスおよび管理インターフェイス用）

始める前に

- GitHub リポジトリ（https://github.com/CiscoDevNet/cisco-asav）から、ARM テンプレート *azure_asav_autoscale.json* をダウンロードします。

ステップ 1 複数の Azure ゾーンに ASAv インスタンスを展開する必要がある場合は、展開リージョンで使用可能なゾーンに基づいて、ARM テンプレートを編集します。

例：

```
"zones": [
    "1",
    "2",
    "3"
],
```

この例は、3 つのゾーンを持つ「Central US」リージョンを示しています。

ステップ 2 外部ロードバランサで必要なトラフィックルールを編集します。この「json」配列を拡張することで、任意の数のルールを追加できます。

例：

```
{
 "type": "Microsoft.Network/loadBalancers",
 "name": "[variables('elbName')]",
 "location": "[resourceGroup().location]",
 "apiVersion": "2018-06-01",
 "sku": {
   "name": "Standard"
```

```
        },
        "dependsOn": [
          "[concat('Microsoft.Network/publicIPAddresses/', variables('elbPublicIpName'))]"

        ],
        "properties": {
          "frontendIPConfigurations": [
            {
              "name": "LoadBalancerFrontEnd",
                "properties": {
                  "publicIPAddress": {
                    "id": "[resourceId('Microsoft.Network/publicIPAddresses/',
variables('elbPublicIpName'))]"
                  }
                }
            }
          ],
          "backendAddressPools": [
            {
              "name": "backendPool"
            }
          ],
          "loadBalancingRules": [
            {
              "properties": {
                "frontendIPConfiguration": {
                "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/frontendIpConfigurations/LoadBalancerFrontend')]"
                },
                "backendAddressPool": {
                "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/backendAddressPools/BackendPool')]"
                },
                "probe": {
                "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/probes/lbprobe')]"
                },
                "protocol": "TCP",
                "frontendPort": "80",
                "backendPort": "80",
                "idleTimeoutInMinutes": "[variables('idleTimeoutInMinutes')]"
              },
              "Name": "lbrule"
            }
          ],
```

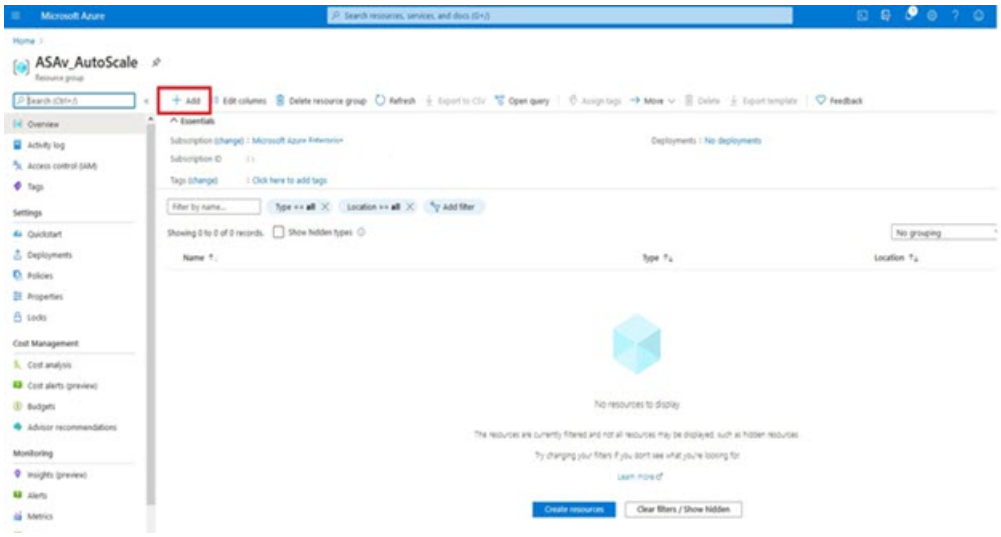
（注） このファイルを編集しない場合は、導入後に Azure ポータルから編集することもできます。

ステップ 3 Microsoft アカウントのユーザー名とパスワードを使用して、Microsoft Azure ポータルにログインします。

ステップ 4 [リソースグループ（Resource Groups）] ブレードにアクセスするには、サービスのメニューから [リソースグループ（Resource groups）] をクリックします。サブスクリプション内のすべてのリソースグループがブレードに一覧表示されます。

新しいリソースグループを作成するか、既存の空のリソースグループを選択します。たとえば、*ASAv_AutoScale*。

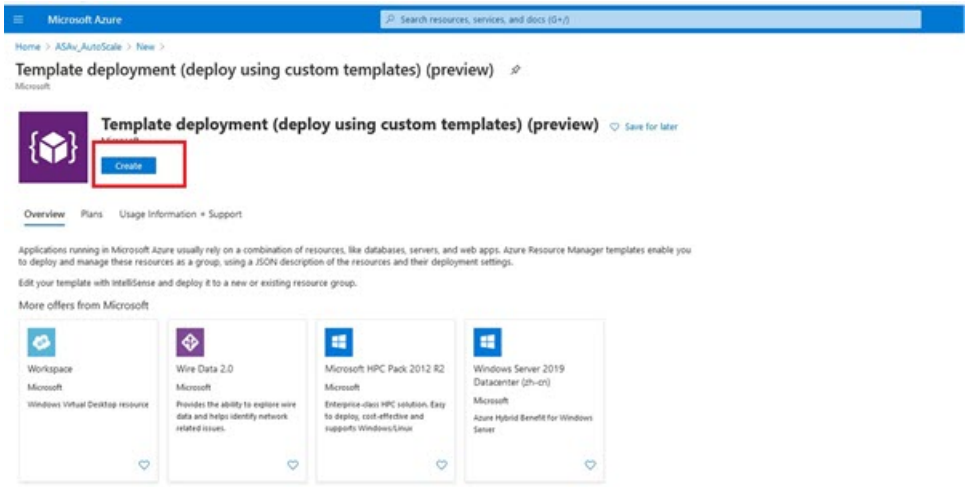
図 *17* : *Azure* ポータル



ステップ 5 [リソースの作成（+）（Create a resource (+)）] をクリックして、テンプレート展開用の新しいリソースを作成します。[リソースグループの作成（Create Resource Group）] ブレードが表示されます。

ステップ 6 [マーケットプレイスの検索（Search the Marketplace）] で、「テンプレートの展開（カスタムテンプレートを使用した展開）（Template deployment (deploy using custom templates)）」と入力し、Enter を押します。

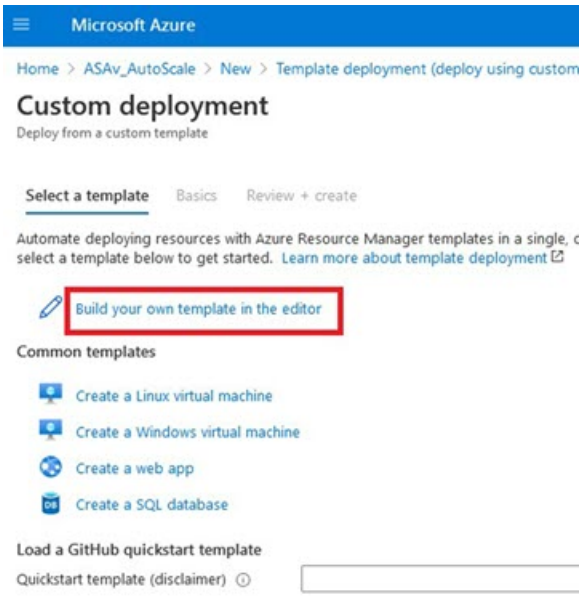
図 *18* : カスタムテンプレートの展開



ステップ 7 [作成（Create）] をクリックします。

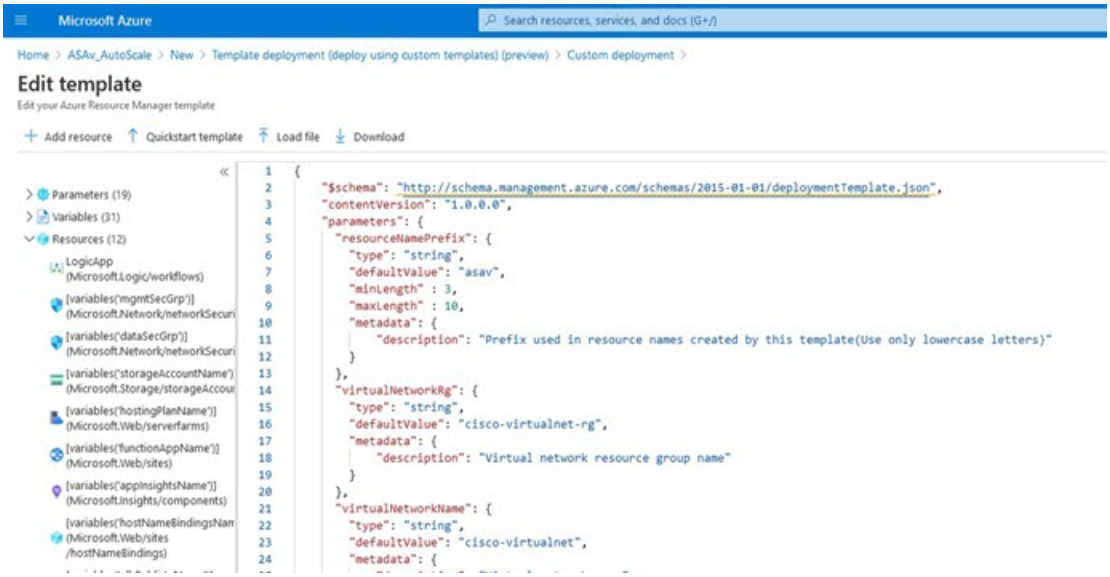
ステップ 8 テンプレートを作成するためのオプションは複数あります。[エディタで独自のテンプレートを作成する（Build your own template in editor）] を選択します。

図 *19* : 独自のテンプレートの作成



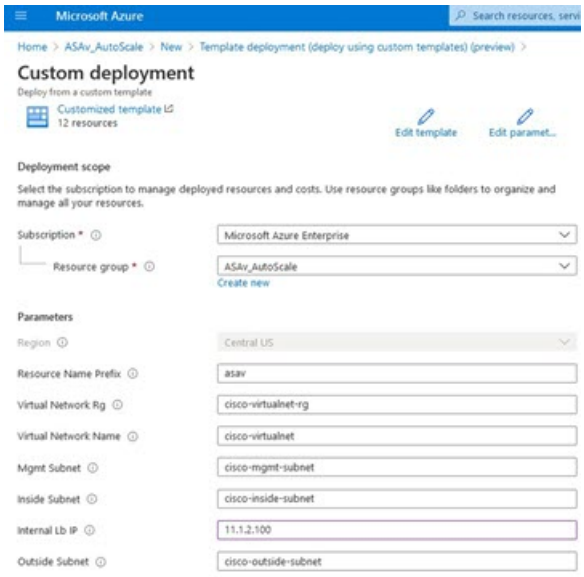
ステップ 9 [テンプレートの編集（Edit template）] ウィンドウで、すべてのデフォルトコンテンツを削除し、更新した *azure_asav_autoscale.json* からコンテンツをコピーして、[保存（Save）] をクリックします。

図 *20 : Edit Template*



ステップ 10 次のセクションで、すべてのパラメータを入力します。各パラメータの詳細については、「入力パラメータ （148 ページ）」を参照してください。次に、[購入（Purchase）] をクリックします。

図 *21 : ARM* テンプレートパラメータ

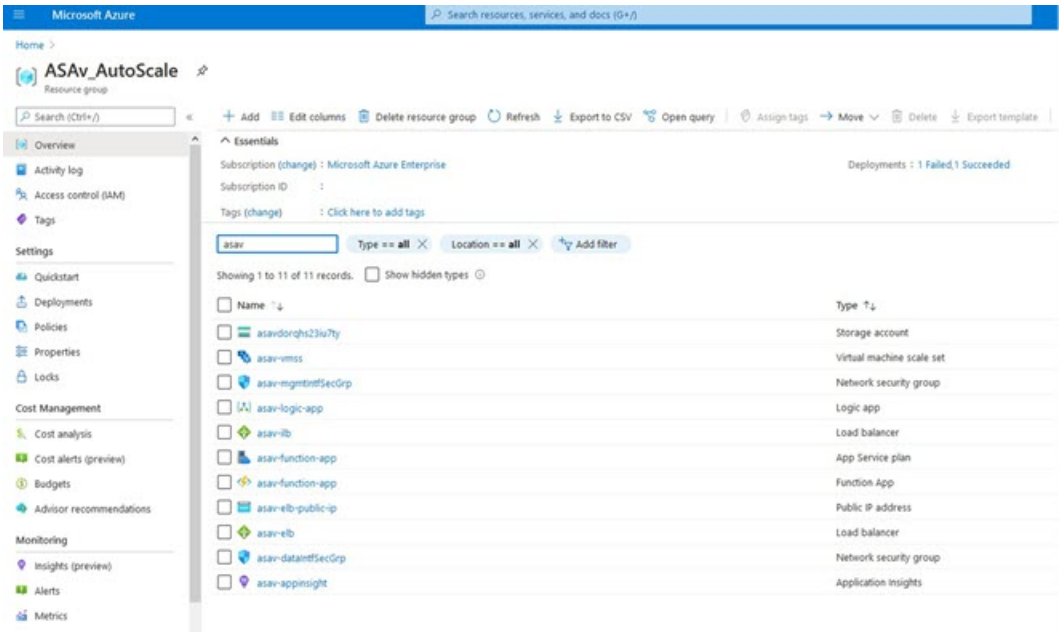


（注） [パラメータの編集（Edit Parameters）] をクリックして、JSON ファイルを編集するか、または事前入力されたコンテンツをアップロードできます。

ARM テンプレートの入力検証機能は限られているため、入力を検証するのはユーザーの責任です。

ステップ 11 テンプレートの展開が成功すると、ASAv Auto Scale for Azure ソリューションに必要なすべてのリソースが作成されます。次の図のリソースを参照してください。[タイプ（Type）]列には、Logic App、VMSS、ロードバランサ、パブリック IP アドレスなどの各リソースが示されます。

図 *22 : ASAv* 自動スケールテンプレートの展開



Azure Function App の展開

ARM テンプレートを展開すると、Azure によってスケルトン Function App が作成されます。このアプリは、Auto Scale Manager ロジックに必要な関数を使用して手動で更新および設定する必要があります。

始める前に

- ASM_Function.zip パッケージをビルドします。「ソースコードからの Azure 関数の構築（171 ページ）」を参照してください。

ステップ 1 ARM テンプレートを展開したときに作成した Function App に移動し、関数が存在しないことを確認します。ブラウザで次の URL にアクセスします。

https://<Function App Name>.scm.azurewebsites.net/DebugConsole

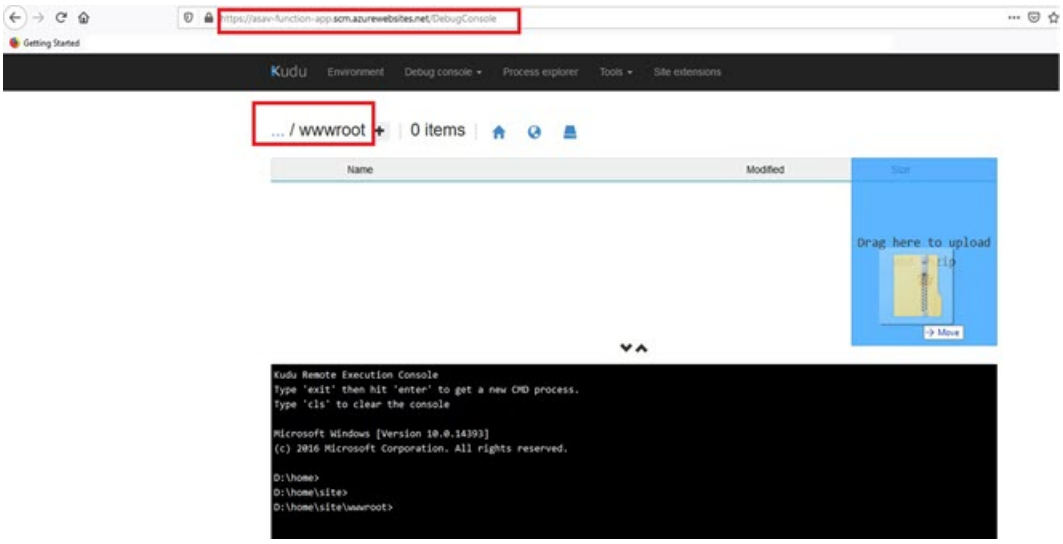
「Auto Scale ARM テンプレートの展開 （153 ページ）」の例の場合、次のようになります。

https://asav-function-app.scm.azurewebsites.net/DebugConsole

ステップ 2 ファイルエクスプローラで、site/wwwroot に移動します。

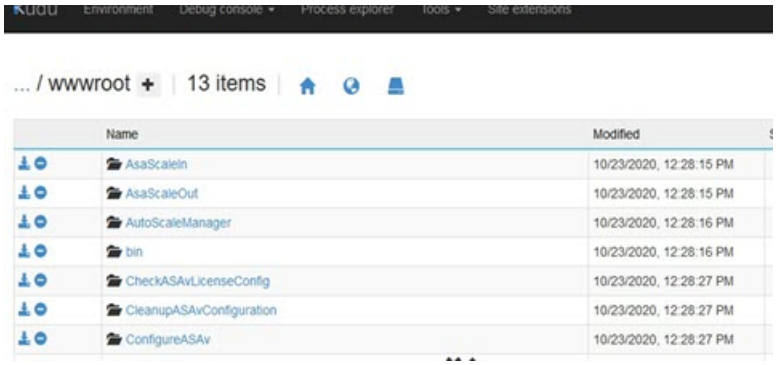
ステップ 3 ASM_Function.zip をファイルエクスプローラの右隅にドラッグアンドドロップします。

図 *23 : ASAv Auto Scale* 機能のアップロード



ステップ 4 アップロードが成功すると、すべてのサーバーレス関数が表示されます。

図 *24 : ASAv* のサーバーレス機能

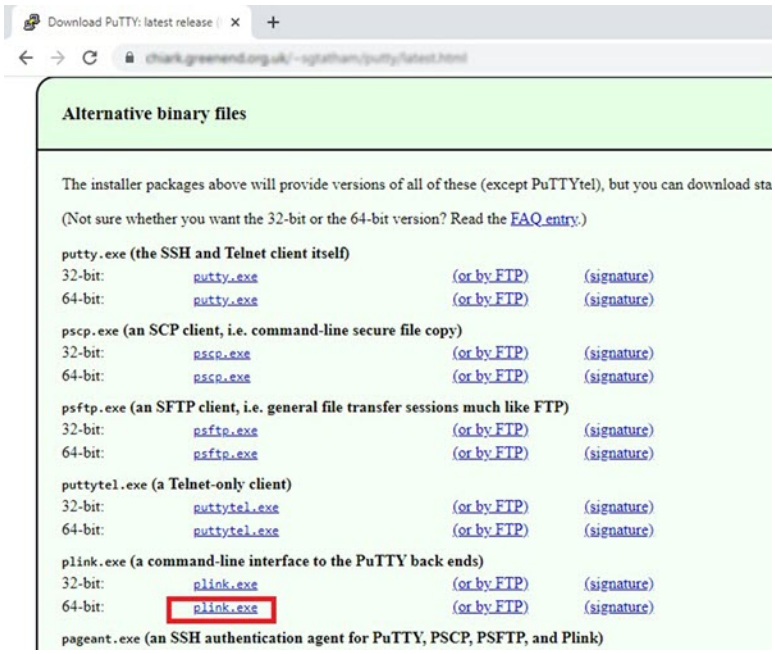


ステップ 5 PuTTY SSH クライアントをダウンロードします。

Azure 関数は、SSH 接続を介して ASAv にアクセスする必要があります。ただし、サーバーレスコードで使用されるオープンソースライブラリは、ASAv で使用される SSH キー交換アルゴリズムをサポートしていません。したがって、事前に構築された SSH クライアントをダウンロードする必要があります。

www.putty.org から PuTTY コマンドライン インターフェイスを PuTTY バックエンド（plink.exe）にダウンロードします。

図 *25 : PuTTY* のダウンロード



ステップ 6 SSH クライアントの実行ファイル **plink.exe** の名前を **asassh.exe** に変更します。

ステップ 7 **asassh.exe** をファイルエクスプローラの右隅（前のステップで **ASM_Function.zip** をアップロードした場所）にドラッグアンドドロップします。

ステップ 8 SSH クライアントが Function App とともに存在することを確認します。必要に応じてページを更新します。

設定の微調整

Auto Scale Manager を微調整したり、デバッグで使用したりするために使用できる設定がいくつかあります。これらのオプションは、ARM テンプレートには表示されませんが、Function App で編集できます。

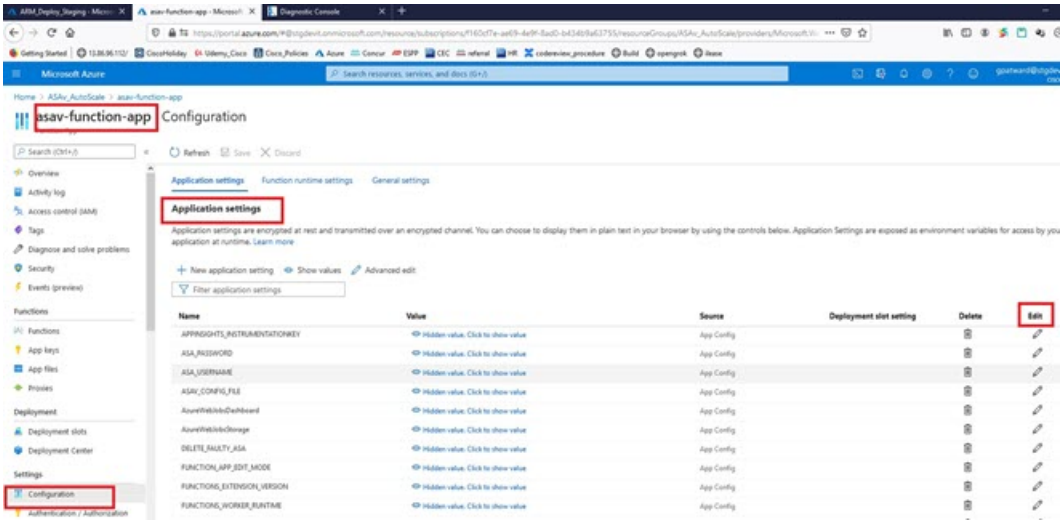
始める前に

（注） 設定はいつでも編集できます。設定を編集する場合は、次の手順に従います。

- Function App を無効にします。
- 既存のスケジュール済みタスクが終了するまで待ちます。
- 設定を編集して保存します。
- Function App を有効にします。

ステップ 1 Azure ポータルで、ASAv Function App を検索して選択します。

図 *26 : ASAv* 機能アプリケーション



ステップ 2 ここでは、ARM テンプレートを介して渡された設定も編集できます。変数名は、ARM テンプレートとは異なる場合がありますが、変数の目的は名前から簡単に識別できます。

ほとんどのオプションは、名前を見ればわかります。次に例を示します。

- [構成名（Configuration Name）] :「DELETE_FAULTY_ASA」（[デフォルト値]（Default value ）] : YES）

  スケールアウト中に、新しい ASAv インスタンスが起動し、 構成ファイルを介して設定されますFMC に登録されます。設定が失敗した場合、このオプションに基づいて、Auto Scale Manager がその ASAv インスタンスを保持するか、削除するかを決定します。（[はい（Yes）] : 障害のある ASAv を削除します。[いいえ（No）] : 設定が失敗した場合でも、ASAv インスタンスを保持します）。

- Function App 設定では、Azure サブスクリプションにアクセスできるユーザーは、すべての変数（「password」などのセキュアな文字列を含んでいる変数を含む）をクリアテキスト形式で表示できます。

  この点に関するセキュリティ上の懸念がある場合（たとえば、Azure サブスクリプションが組織内の低い権限を持つユーザー間で共有されている場合）、ユーザーは Azure の Key Vault サービスを使用してパスワードを保護できます。この設定をすると、関数の設定でクリアテキストの「password」を入力する代わりに、ユーザーは、パスワードが保存されている Key Vault によって生成された、セキュアな識別子を入力する必要があります。

  （注） Azure のドキュメントを検索して、アプリケーションデータを保護するためのベストプラクティスを見つけてください。

仮想マシンスケールセットでの IAM ロールの設定

Azure Identity and Access Management（IAM）は、Azure Security and Access Control の一部として使用され、ユーザーの ID を管理および制御します。Azure リソースのマネージド ID は、Azure Active Directory で自動的にマネージド ID が Azure サービスに提供されます。

これにより、明示的な認証ログイン情報がなくても、Function App が仮想マシンスケールセット（VMSS）を制御できます。

ステップ 1 Azure ポータルで、VMSS に移動します。

ステップ 2 [アクセス制御（IAM）（Access control (IAM)）] をクリックします。

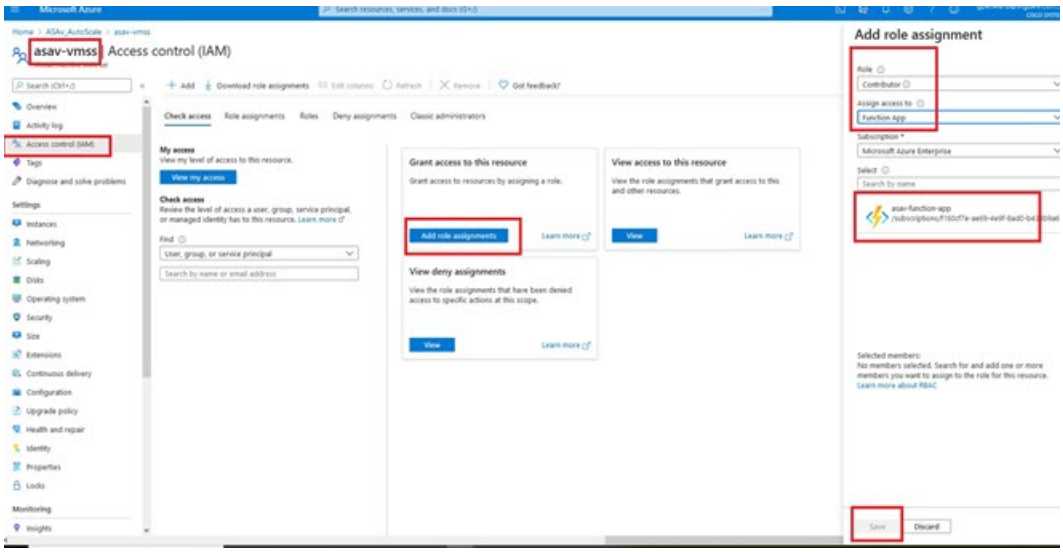
ステップ 3 [追加（Add）] をクリックしてロールの割り当てを追加します。

ステップ 4 [ロール割り当ての追加（Add role assignment）] ドロップダウンから、[共同作成者（Contributor）] を選択します。

ステップ 5 [アクセスの割り当て先（Assign access to）] ドロップダウンから、[Function App] を選択します。

ステップ 6 ASAv Function App を選択します。

図 *27 : AIM* ロールの割り当て



ステップ 7 [保存（Save）] をクリックします。

（注） まだ ASAv インスタンスが起動していないことも確認する必要があります。

Azure セキュリティグループの更新

ARM テンプレートは、管理インターフェイス用とデータインターフェイス用の 2 つのセキュリティグループを作成します。管理セキュリティグループは、ASAv 管理アクティビティに必

要なトラフィックのみを許可します。ただし、データインターフェイスのセキュリティグループはすべてのトラフィックを許可します。

展開のトポロジとアプリケーションのニーズに基づいてセキュリティグループのルールを微調整します。

（注） データインターフェイスのセキュリティグループは、少なくともロードバランサからの SSH トラフィックを許可する必要があります。

Azure Logic App の更新

Logic App は、Auto Scale 機能の Orchestrator として機能します。ARM テンプレートによってスケルトン Logic App が作成されます。このアプリケーションを手動で更新して、Auto Scale Orchestrator として機能するために必要な情報を提供する必要があります。

ステップ 1 リポジトリから、LogicApp.txt ファイルをローカルシステムに取得し、次のように編集します。

重要 手順をすべて読んで理解してから続行してください。

手動の手順は、ARM テンプレートでは自動化されないため、Logic App のみ後で個別にアップグレードできます。

a) 必須: すべての「SUBSCRIPTION_ID」を検索し、サブスクリプション ID 情報に置き換えます。
b) 必須: すべての「RG_NAME」を検索し、リソースグループ名に置き換えます。
c) 必須: すべての「FUNCTIONAPPNAME」を検索し、Function App 名に置き換えます。

次の例は、LogicApp.txt ファイルの行の一部を示しています。

```
  "AutoScaleManager": {
      "inputs": {
          "function": {
              "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/AutoScaleManager"

          }
.
.
                      },
                      "Deploy_Changes_to_ASA": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
                                  "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeployConfiguration"

                              }
.
.
                      "DeviceDeRegister": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
```

```
                                "function": {
                                    "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeviceDeRegister"

                                }
                            },
                            "runAfter": {
                                "Delay_For_connection_Draining": [
```

d) （任意）トリガー間隔を編集するか、デフォルト値（5）のままにします。これは、Auto Scale 機能が定期的にトリガーされる時間間隔です。次の例は、LogicApp.txt ファイルの行の一部を示しています。

```
"triggers": {
    "Recurrence": {
        "conditions": [],
        "inputs": {},
        "recurrence": {
            "frequency": "Minute",
            "interval": 5
        },
```

e) （任意）ドレインする時間を編集するか、デフォルト値（5）のままにします。これは、スケールイン操作中にデバイスを削除する前に、ASAv から既存の接続をドレインする時間間隔です。次の例は、LogicApp.txt ファイルの行の一部を示しています。

```
"actions": {
     "Branch_based_on_Scale-In_or_Scale-Out_condition": {
         "actions": {
             "Delay_For_connection_Draining": {
                 "inputs": {
                     "interval": {
                         "count": 5,
                         "unit": "Minute"
                     }
```

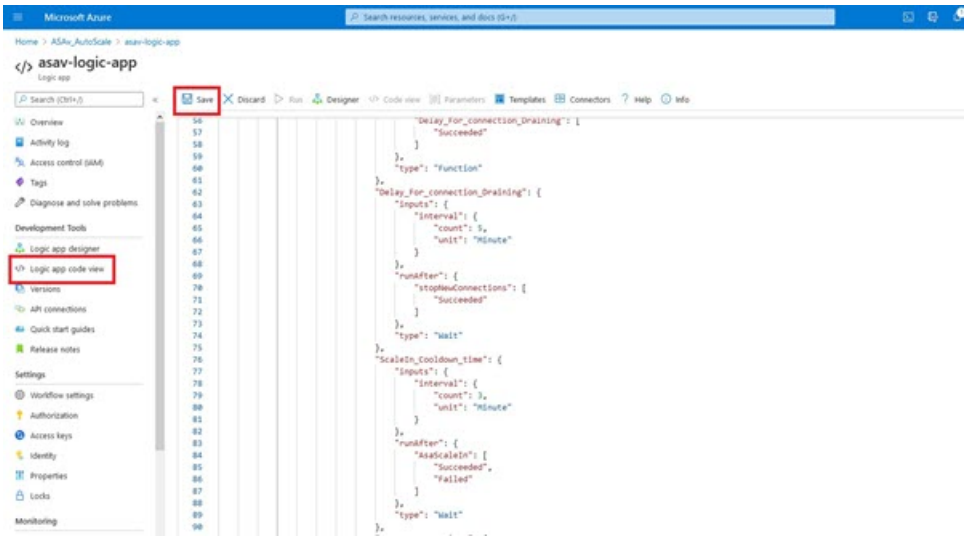
f) （任意）クールダウン時間を編集するか、デフォルト値（10）のままにします。これは、スケールアウト完了後に NO ACTION を実行する時間です。次の例は、LogicApp.txt ファイルの行の一部を示しています。

```
"actions": {
    "Branch_based_on_Scale-Out_or_Invalid_condition": {
        "actions": {
            "Cooldown_time": {
                "inputs": {
                    "interval": {
                        "count": 10,
                        "unit": "Second"
                    }
```

（注） これらの手順は、Azure ポータルからも実行できます。詳細については、Azure のドキュメントを参照してください。

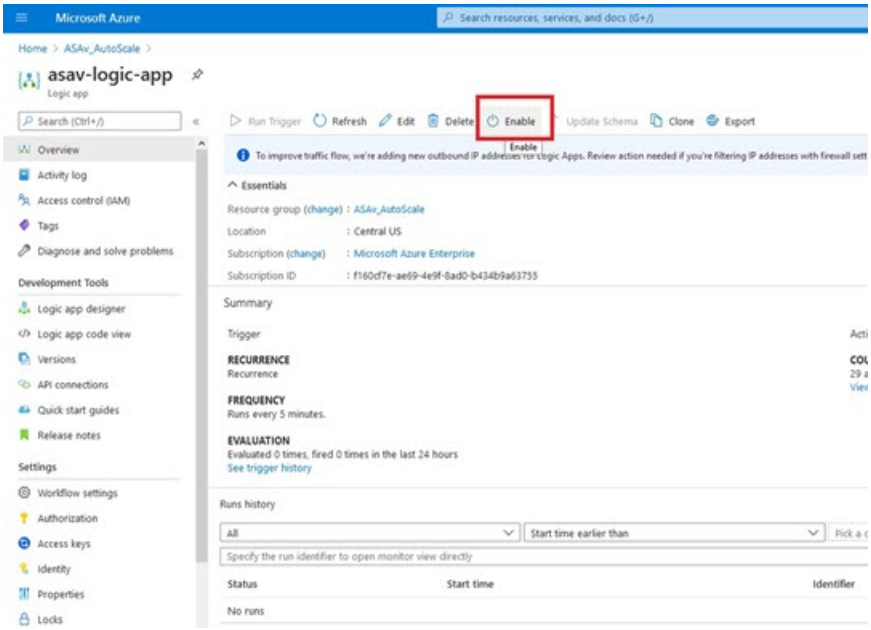
ステップ 2 [Logic Appコードビュー（Logic App code view）] に移動し、デフォルトの内容を削除して、編集した LogicApp.txt ファイルの内容を貼り付け、[保存（Save）] をクリックします。

図 *28 : Logic App* コードビュー



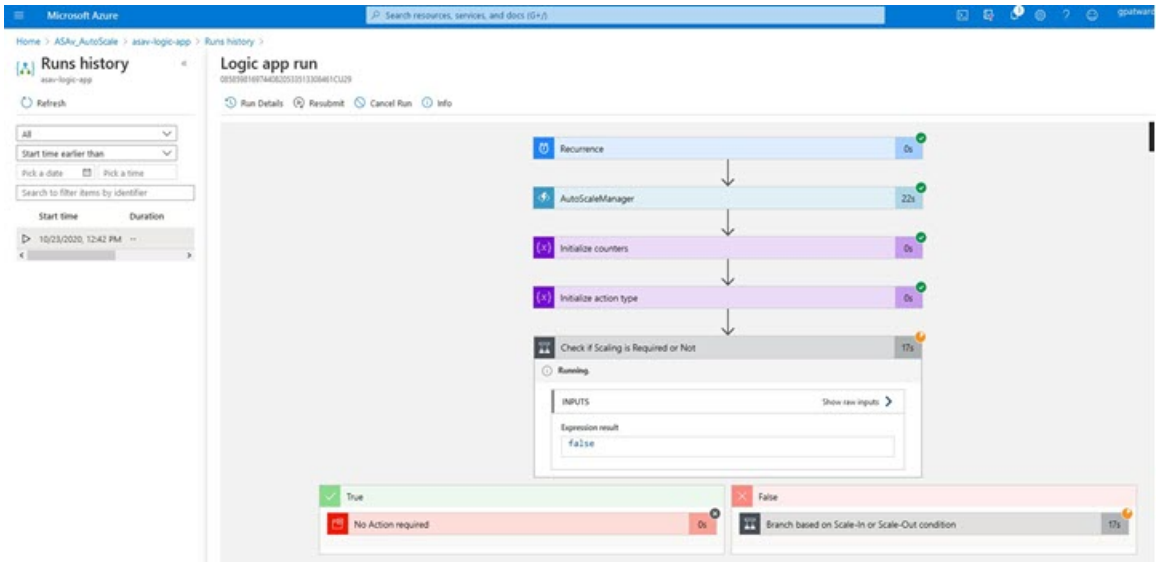
ステップ 3 Logic App を保存すると、[無効（Disabled）] 状態になります。Auto Scale Manager を起動する場合は、[有効化（Enable）] をクリックします。

図 *29 : Logic App* の有効化



ステップ 4 有効にすると、タスクの実行が開始されます。[実行中（Running）] ステータスをクリックしてアクティビティを表示します。

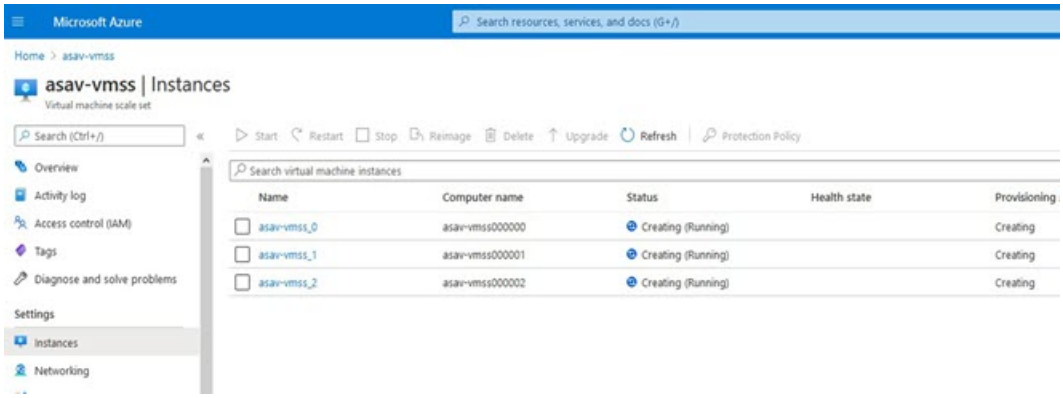
図 *30 : Logic App* の実行ステータス



ステップ 5 Logic App が起動すると、導入関連のすべての手順が完了します。

ステップ 6 ASAv インスタンスが作成されていることを VMSS で確認します。

図 *31 :* 稼働中の*ASAv* インスタンス



この例では、ARM テンプレートの展開で「'minAsaCount'」が「3」に設定され、「initDeploymentMode」が「BULK」に設定されているため、3 つの ASAv インスタンスが起動されます。

FTDvASAv の更新

ASAv アップグレードは、仮想マシンスケールセット（VMSS）のイメージアップグレードの形式でのみサポートされます。したがって、ASAv は Azure REST API インターフェイスを介してアップグレードします。

（注） 任意の REST クライアントを使用して ASAv をアップグレードできます。

始める前に

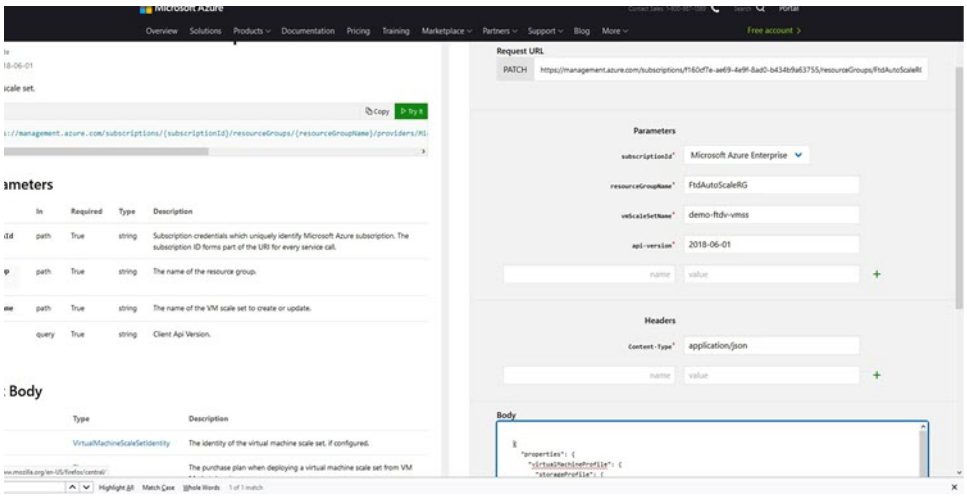
- 市場で入手可能な新しい ASAv イメージバージョンを取得します（例：914.001）。
- 元のスケールセットの展開に使用する SKU を取得します（例：asav-azure-byol）。
- リソースグループと仮想マシンスケールセット名を取得します。

ステップ 1 ブラウザで次の URL にアクセスします。

https://docs.microsoft.com/en-us/rest/api/compute/virtualmachinescalesets/update#code-try-0

ステップ 2 [パラメータ（Parameters）] セクションに詳細を入力します。

図 *32 : FTDvASAv* の更新



ステップ 3 新しい ASAv イメージバージョン、SKU、トリガー RUN を含む JSON 入力を **[本文（Body）]** セクションに入力します。

```
{
 "properties": {
        "virtualMachineProfile": {
              "storageProfile": {
                      "imageReference": {
                           "publisher": "cisco",
                           "offer": "cisco-asav",
                           "sku": "asav-azure-byol",
                            "version": "650.32.0"
                      }
                  },
            }
      }
```

```
}
```

ステップ 4 VMSS が変更を受け入れると、Azure から成功の応答が返ってきます。

新しいイメージは、スケールアウト操作の一環として起動される新しい ASAv インスタンスで使用されます。

- 既存の ASAv インスタンスは、スケールセットに存在している間、古いソフトウェアイメージを使用し続けます。
- 前述の動作を上書きし、既存の ASAv インスタンスを手動でアップグレードできます。これを行うには、VMSS の [アップグレード（Upgrade）] ボタンをクリックします。選択した ASAv インスタンスが再起動されて、アップグレードされます。アップグレードされた ASAv インスタンスは手動で再登録および再設定する必要があります。**この方法は推奨されません。**

Auto Scale ロジック

スケールアウトロジック

- **POLICY-1**：設定された期間に、**いずれか** ASAvの平均負荷がスケールアウトしきい値を超えるとスケールアウトがトリガーされます。
- **POLICY-2**：設定された期間に、**すべて**の ASAv デバイスの平均負荷がスケールアウトしきい値を超えるとスケールアウトがトリガーされます。「

スケールインロジック

- 設定された期間に、**すべて**の ASAv デバイスの CPU 使用率が設定されたスケールインしきい値を下回った場合。

注意

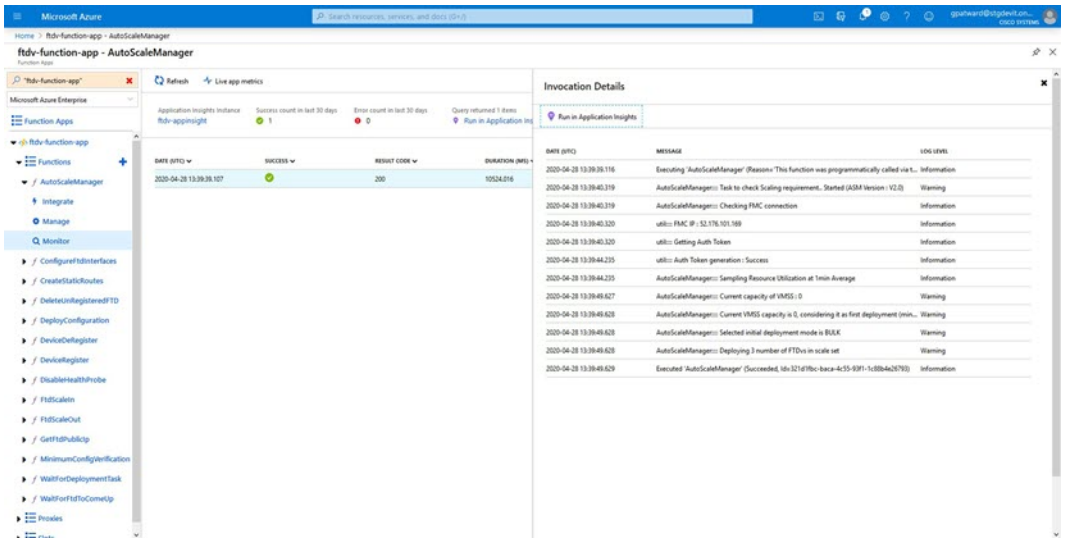
- スケールイン/スケールアウトは 1 つずつ行われます（つまり、一度に 1 つの ASAv だけがスケールインまたはスケールアウトされます）。
- 上記のロジックは、ロードバランサがすべての ASAv デバイスに接続を均等に分散しようとし、平均してすべての ASAv デバイスが均等にロードされるという前提に基づいています。

Auto Scale のロギングとデバッグ

サーバーレスコードの各コンポーネントには、独自のロギングメカニズムがあります。また、ログはアプリケーションインサイトにパブリッシュされます。

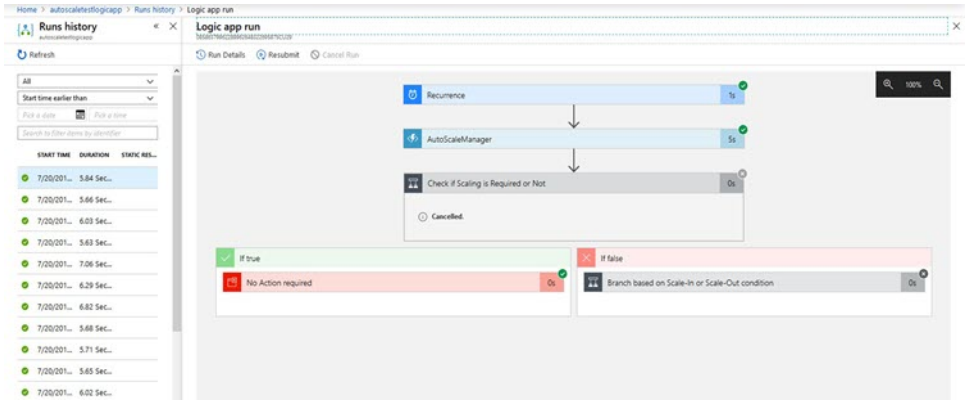
- 個々の Azure 関数のログを表示できます。

図 *33 : Azure* 関数ログ



- Logic App とその個々のコンポーネントの実行ごとに同様のログを表示できます。

図 *34 : Logic App* の実行ログ



- 必要な場合は、Logic App で実行中のタスクをいつでも停止または終了できます。ただし、現在実行中の ASAv デバイスが起動または終了すると、一貫性のない状態になります。
- 各実行または個々のタスクにかかった時間は、Logic App で確認できます。
- Function App は、新しい zip をアップロードすることでいつでもアップグレードできます。Logic App を停止し、すべてのタスクの完了を待ってから、Function App をアップグレードします。

Auto Scale のガイドラインと制約事項

ASAv Auto Scale for Azure を導入する場合は、次のガイドラインと制限事項に注意してください。

- スケーリングの決定は、CPU 使用率に基づきます。
- ASAv 管理インターフェイスは、パブリック IP アドレスを持つように設定されます。
- IPv4 だけがサポートされます。
- ARM テンプレートの入力検証機能は限られているため、入力を正しく検証するのはユーザーの責任です。
- Azure 管理者は、Function App 環境内の機密データ（管理者ログイン情報やパスワードなど）をプレーンテキスト形式で確認できます。Azure Key Vault サービスを使用して、センシティブデータを保護できます。

Auto Scale のトラブルシューティング

次に、ASAv Auto Scale for Azure 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。

- ASAv に SSH 接続できない：複雑なパスワードがテンプレートを介して ASAv に渡されているか確認します。セキュリティグループで SSH 接続が許可されているか確認します。
- ロードバランサのヘルスチェックエラー：ASAv がデータインターフェイスの SSH に応答しているか確認します。セキュリティグループの設定を確認します。
- トラフィックの問題：ロードバランサルール、ASAv で設定された NAT ルールおよびスタティックルートを確認します。テンプレートとセキュリティグループルールで提供される Azure 仮想ネットワーク/サブネット/ゲートウェイの詳細を確認します。
- Logic App が VMSS にアクセスできない：VMSS の IAM ロール設定が正しいか確認します。
- Logic App の実行時間が長すぎる：スケールアウトされた ASAv デバイスで SSH アクセスを確認します。Azure VMSS で ASAv デバイスの状態を確認します。
- サブスクリプション ID 関連の Azure 関数のスローエラー：アカウントでデフォルトのサブスクリプションが選択されていることを確認します。
- スケールイン操作の失敗：Azure でのインスタンスの削除には長時間かかることがあります。このような状況では、スケールイン操作がタイムアウトし、エラーが報告されますが、最終的にはインスタンスが削除されます。
- 設定を変更する前に、Logic App を無効にし、実行中のすべてのタスクが完了するまで待ちます。

ソースコードからの Azure 関数の構築

システム要件

- Microsoft Windows デスクトップ/ラップトップ。
- Visual Studio（Visual Studio 2019 バージョン 16.1.3 でテスト済み）

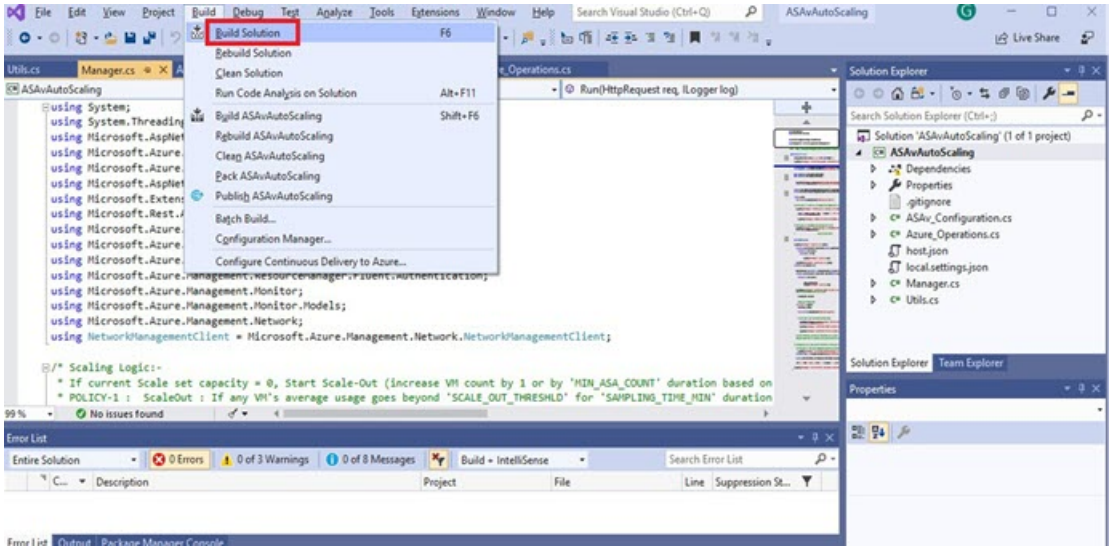
（注） Azure 関数は C# を使用して記述されます。

- 「Azure 開発」ワークロードを Visual Studio にインストールする必要があります。

Visual Studio を使用したビルド

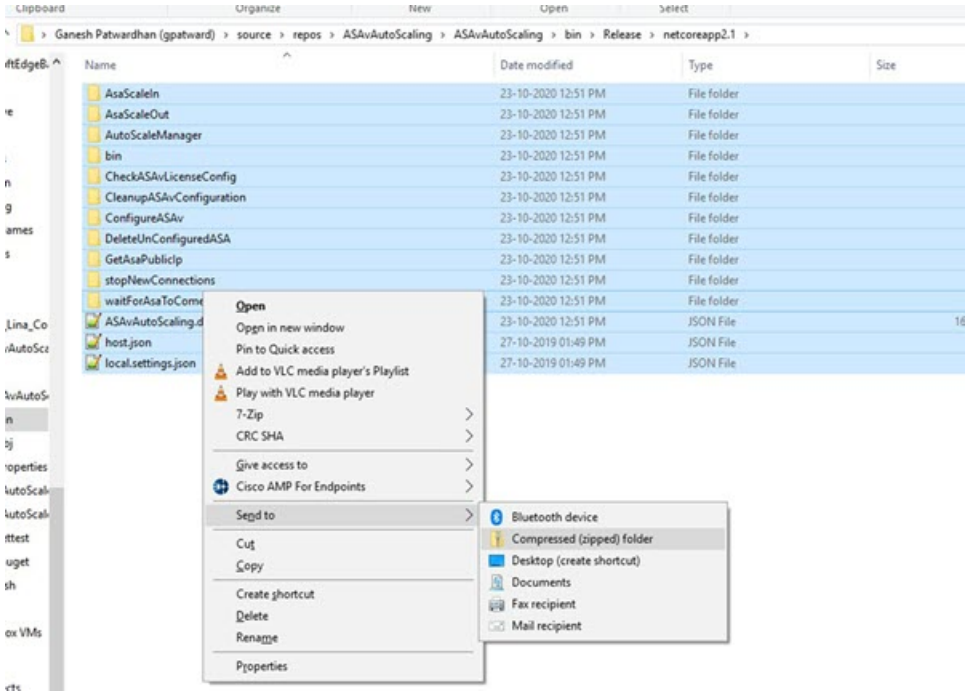
1. 「code」フォルダをローカルマシンにダウンロードします。
2. 「ASAAutoScaling」フォルダに移動します。
3. Visual Studio でプロジェクトファイル「ASAAutoScaling」を開きます。
4. クリーンアップしてビルドするには、Visual Studio の標準手順を使用します。

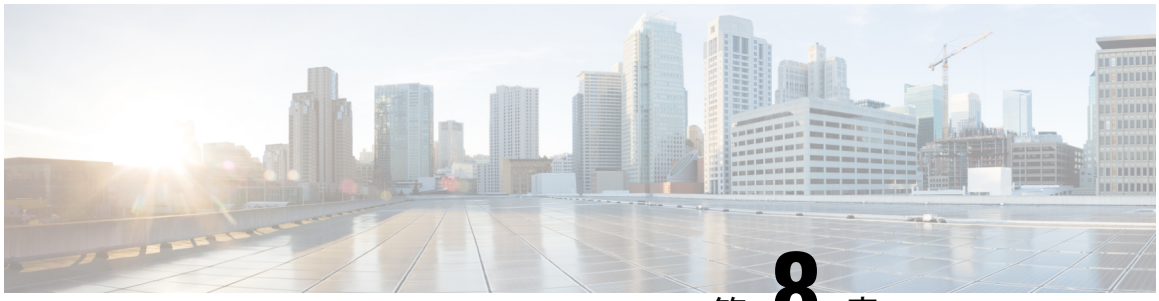
図 *35 : Visual Studio* ビルド



5. ビルドが正常にコンパイルされたら、**\bin\Release\netcoreapp2.1** フォルダに移動します。
6. すべての内容を選択し、**[送信先（Send to）]>[圧縮（ZIP）フォルダ（Compressed (zipped) folder）]** の順にクリックして、ZIP ファイルを ASM_Function.zip として保存します。

図 *36 : ASM_Function.zip* のビルド





第 8 章

Rackspace Cloud への ASAv の導入

Rackspace Cloud に ASAv 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 ASAv vCPU/メモリ構成ですべての ASAv ライセンスを使用できるようになりました。これにより、ASAv 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。

Rackspace Cloud への ASAv の導入について

Rackspace は、あらゆる主要なパブリックおよびプライベートクラウド テクノロジーにわたる専門知識とマネージドサービスを提供するリーディングプロバイダです。Rackspace Cloud は、ユーティリティ コンピューティング ベースで課金が行われるクラウドコンピューティング製品およびサービスのセットです。

Rackspace Cloud で ASAv for Rackspace を仮想アプライアンスとして導入できます。この章では、単一インスタンスの ASAv アプライアンスをインストールして構成する方法について説明します。

Rackspace Cloud のインスタンスタイプは、フレーバと呼ばれます。フレーバという用語は、RAM サイズ、vCPU、ネットワークスループット（RXTX ファクタ）、ディスク容量から成るサーバーの組み合わせを指します。次の表に、ASAv の導入に適した Rackspace フレーバを示します。

表 *20 : Rackspace* でサポートされるフレーバ

フレーバ	属性		総帯域幅
	vCPU	メモリ（**GB**）	
汎用 1-2	2	2	400 Mbps
汎用 1-4	4	4	800 Mbps
汎用 1-8	8	8	1.6 Gbps
コンピューティング 1-4	2	3.75	312.5 Mbps
コンピューティング 1-8	4	7.5	625 Mbps
コンピューティング 1-15	8	15	1.3 Gbps
メモリ 1-15	2	15	625 Mbps
メモリ 1-15	4	30	1.3 Gbps
メモリ 1-15	8	60	2.5 Gbps

Rackspace のフレーバについて

Rackspace 仮想クラウドサーバーのフレーバは、次のクラスに分類されます。

- **汎用 v1**
  - 汎用ワークロードから高パフォーマンスの Web サイトまで、さまざまなユースケースに役立ちます。
  - vCPU はオーバーサブスクライブされ、「バースト可能」です。つまり、物理ホスト上のクラウドサーバーに割り当てられる vCPU の数は、物理 CPU スレッドの数よりも多くなります。
- **コンピューティング v1**
  - Web サーバー、アプリケーションサーバー、およびその他の CPU 集約型のワークロード向けに最適化されています。
  - vCPU は「予約済み」です。つまり、物理ホスト上のクラウドサーバーに割り当てられる vCPU の数は、そのホスト上の物理 CPU スレッドの数よりも多くなることはありません。
- **メモリ v1**
  - メモリ集約型のワークロードに推奨されます。

- **I/O v1**
  - 高速ディスク I/O のメリットを得やすい高パフォーマンスのアプリケーションおよびデータベースに最適です。

ASAv と Rackspace の前提条件

- Rackspace アカウントを作成します。

  すべての Rackspace Public Cloud アカウントは、デフォルトで Managed Infrastructure サービスレベルに設定されます。クラウドコントロールパネル内で Managed Operations サービスレベルにアップグレードできます。クラウドコントロールパネルの上部で、アカウントのユーザー名をクリックし、[Upgrade Service Level] を選択します。

- ASAv へのライセンス付与。ASAv 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。「ASAv のライセンス（1 ページ）」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
  - 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 ASAv 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 ASAv 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）：ASAv 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）：DMZ ネットワークに ASAv を接続するために使用されます。

- ASA および ASAv システムの互換性と要件については、Cisco ASA の互換性 [英語] を参照してください。

Rackspace Cloud ネットワーク

クラウド構成には、必要に応じて接続された複数の種類のネットワークを含めることができます。クラウドサーバーのネットワーキング機能は、多くの場合、他のネットワークと同じ方法

で管理できます。ASAv の導入では、主に、Rackspace Cloud の次の 3 種類の仮想ネットワークと情報を交換します。

- PublicNet：クラウドサーバー、クラウドロードバランサ、ネットワークアプライアンスなどのクラウド インフラストラクチャ コンポーネントをインターネットに接続します。
  - PublicNet を使用して、ASAv をインターネットに接続します。
  - ASAv は、Management0/0 インターフェイスを介してこのネットワークに接続します。
  - PublicNet は、IPv4 と IPv6 のデュアルスタックです。PublicNet を使用してサーバーを作成すると、そのサーバーはデフォルトで IPv4 アドレスと IPv6 アドレスを受け取ります。
- ServiceNet：各 Rackspace クラウドリージョン内の IPv4 専用の内部マルチテナントネットワーク。
  - ServiceNet は、構成内のサーバー間でトラフィック（East-West トラフィック）を伝送するように最適化されます。
  - クラウドファイル、クラウドロードバランサ、クラウドデータベース、クラウドバックアップなどのリージョン別サービスへの無料アクセスをサーバーに提供します。
  - ネットワーク 10.176.0.0/12 および 10.208.0.0/12 は ServiceNet 用に予約されています。ServiceNet 接続を備えるサーバーは、これらのネットワークのいずれかの IP アドレスを使用してプロビジョニングされます。
  - ASAv は、Gigabit0/0 インターフェイスを介してこのネットワークに接続します。
- プライベート Cloud Networks：Cloud Networks を使用すると、クラウドで分離された安全なネットワークを作成および管理できます。
  - これらのネットワークは単一のテナントであり、ネットワークトポロジ、IP アドレッシング（IPv4 または IPv6）、および接続するクラウドサーバーを完全に制御できます。
  - Cloud Networks はリージョンを対象範囲とし、特定のリージョン内の任意のクラウドサーバーに接続できます。
  - API を介して、または Rackspace Cloud コントロールパネルを使用して、Cloud Networks を作成および管理できます。

    ASAv は、Gigabit0/1 ～ Gigabit0/8 のインターフェイスを介してこれらのネットワークに接続します。

Rackspace の第 0 日の構成

Rackspace Cloud に VM を展開すると、Rackspace のプロビジョニング情報を持つファイルを含む CD-ROM デバイスが VM に接続されます。プロビジョニング情報には次の項目があります。

- ホスト名
- 必要なインターフェイスの IP アドレス
- スタティック IP ルート
- ユーザー名とパスワード（オプションの SSH 公開キー）
- DNS サーバー
- NTP サーバー

これらのファイルは初期展開時に読み込まれ、ASA の構成が生成されます。

ASAv ホスト名

デフォルトでは、ASAv ホスト名は、ASAv の構築を開始するときにクラウドサーバーに割り当てる名前です。

```
hostname rackspace-asav
```

ASA ホスト名構成では、RFC 1034 および 1101 に準拠するホスト名のみ使用できます。

- 先頭と末尾が文字または数字である必要があります。
- 内側の文字は、文字、数字、またはハイフンである必要があります。

（注） ASAv では、これらのルールに準拠するように、元のクラウドサーバー名にできるだけ近い名前にクラウドサーバー名が変更されます。クラウドサーバー名の先頭と末尾に特殊文字がある場合はそれを削除し、ルールに準拠しない内側の文字をハイフンに置き換えます。

たとえば、クラウドサーバーの名前が **ASAv-9.13.1.200** の場合、ホスト名は **ASAv-9-13-1-200** になります。

Interfaces

インターフェイスは次のように設定されます。

- Management0/0
  - PublicNet に接続されているため、「outside」という名前が付けられます。
  - Rackspace は、IPv4 と IPv6 の両方のパブリックアドレスを PublicNet インターフェイスに割り当てます。
- Gigabit0/0
  - ServiceNet に接続されているため、「management」という名前が付けられます。

- Rackspace は、Rackspace リージョンの ServiceNet サブネットから IPv4 アドレスを割り当てます。

- Gigabit0/1 ー Gigabit0/8
  - プライベート Cloud Networks に接続されているため、「inside」、「inside02」、「inside03」などの名前が付けられます。
  - Rackspace は、Cloud Networks サブネットから IP アドレスを割り当てます。

3 つのインターフェイスを持つ ASAv のインターフェイス構成は次のようになります。

```
interface GigabitEthernet0/0
 nameif management
 security-level 0
 ip address 10.176.5.71 255.255.192.0
!
interface GigabitEthernet0/1
 nameif inside
 security-level 100
 ip address 172.19.219.7 255.255.255.0
!
interface Management0/0
 nameif outside
 security-level 0
 ip address 162.209.103.109 255.255.255.0
 ipv6 address 2001:4802:7800:1:be76:4eff:fe20:1763/64
```

スタティック ルート

Rackspace は、次のスタティック IP ルートをプロビジョニングします。

- PublicNet インターフェイス（**outside**）経由のデフォルト IPv4 ルート。
- PublicNet インターフェイス経由のデフォルト IPv6 ルート。
- ServiceNet インターフェイス（**management**）上のインフラストラクチャ サブネット ルート。

```
route outside 0.0.0.0 0.0.0.0 104.130.24.1 1
ipv6 route outside ::/0 fe80::def
route management 10.176.0.0 255.240.0.0 10.176.0.1 1
route management 10.208.0.0 255.240.0.0 10.176.0.1 1
```

ログイン クレデンシャル

Rackspaceによって作成されたパスワードを使用して、「admin」という名前のユーザーが作成されます。Rackspace 公開キーを使用してクラウドサーバーが展開されている場合、ユーザー「admin」の公開キーが作成されます。

```
username admin password <admin_password> privilege 15
username admin attributes
```

Rackspace Cloud への ASAv の導入

```
ssh authentication publickey <public_key>
```

day0 SSH 構成：

- PublicNet インターフェイス（**outside**）経由の SSH が IPv4 と IPv6 に対して有効になります。
- ServiceNet インターフェイス（**management**）経由の SSH が IPv4 に対して有効になります。
- Rackspace の要求に応じて、より強力なキー交換グループを設定します。

```
aaa authentication ssh console LOCAL
ssh 0 0 management
ssh 0 0 outside
ssh ::0/0 outside
ssh version 2
ssh key-exchange group dh-group14-sha1
```

DNS と NTP

Rackspace は、DNS と NTP に使用される 2 つの IPv4 サービスアドレスを提供します。

```
dns domain-lookup outside
dns server-group DefaultDNS
 name-server 69.20.0.164
 name-server 69.20.0.196

ntp server 69.20.0.164
ntp server 69.20.0.196
```

Rackspace Cloud への ASAv の導入

Rackspace Cloud で ASAv を仮想アプライアンスとして導入できます。この手順では、単一インスタンスの ASAv アプライアンスをインストールする方法を示します。

始める前に

ホスト名の要件、インターフェイスのプロビジョニング、ネットワーク情報など、ASAv の導入を成功させるために Rackspace Cloud で有効にする構成パラメータの説明については、Rackspace の第 0 日の構成 （176 ページ）のトピックを参照してください。

ステップ 1 Rackspace mycloud ポータルで、**[SERVERS]** > **[CREATE RESOURCES]** > **[Cloud Server]** に移動します。

ステップ 2 [Create Server] ページの [Server Details] で次のように入力します。

a) [サーバー名（Server Name）] フィールドに ASAv マシンの名前を入力します。

b) [Region] ドロップダウンリストからリージョンを選択します。

ステップ 3 [Image] で、**[Linux/Appliances]** > **[ASAv]** > **[Version]** を選択します。

（注） 通常、新しい ASAv を導入する場合は、サポートされている最新バージョンを選択します。

ステップ 4 [Flavor] で、リソースのニーズに合った [Flavor Class] を選択します。適切な VM のリストについては、表 20 : Rackspace でサポートされるフレーバ （174 ページ）を参照してください。

重要 9.13(1) 以降は、ASAv の最小メモリ要件は 2GB です。1 つ以上の vCPU を使用して ASAv を導入する場合、ASAv の最小メモリ要件は 4GB です。

ステップ 5 （オプション）[Advanced Options] で、SSH キーを設定します。

Rackspace Cloud の SSH キーの詳細については、「Managing access with SSH keys」を参照してください。

ステップ 6 ASAv の該当する [推奨のインストール（Recommended Installs）] および [項目別のチャージ（Itemized Charges）] を確認し、[サーバーの作成（Create Server）] をクリックします。

root 管理者のパスワードが表示されます。パスワードをコピーし、ダイアログを閉じます。

ステップ 7 サーバーを作成すると、サーバーの詳細ページが表示されます。サーバーのステータスがアクティブになるまで待ちます。通常、これには数分かかります。

次のタスク

- ASAv に接続します。
- SSH を介して入力できる CLI コマンドを使用するか、または ASDM を使用して、設定を続行します。ASDM にアクセスする手順については、ASDM の起動 （313 ページ）を参照してください。

CPU 使用率とレポート

CPU 使用率レポートには、指定された時間内に使用された CPU の割合の要約が表示されます。通常、コアはピーク時以外には合計 CPU 容量の約 30 ～ 40% で動作し、ピーク時は約 60 ～ 70% の容量で動作します。

ASA Virtual の vCPU 使用率

ASA Virtual の vCPU 使用率には、データパス、制御ポイント、および外部プロセスで使用されている vCPU の量が表示されます。

Rackspace で報告される vCPU 使用率には、ASA Virtual の使用率に加えて、次のものが含まれます。

- ASA Virtual アイドル時間
- ASA Virtual マシンに使用された %SYS オーバーヘッド

- vSwitch、vNIC および pNIC の間を移動するパケットのオーバーヘッド。このオーバーヘッドは非常に大きくなる場合があります。

CPU 使用率の例

CPU 使用率の統計情報を表示するには、**show cpu usage** コマンドを使用します。

例

```
Ciscoasa#show cpu usage

CPU □□□□5□□□ 1%□1 □□□ 2%□5 □□□ 1%
```

報告された vCPU の使用率が大幅に異なる例を次に示します。

- ASA Virtual レポート：40%
- DP：35%
- 外部プロセス：5%
- ASA（ASA Virtual レポート）：40%
- ASA アイドル ポーリング：10%
- オーバーヘッド：45%

オーバーヘッドは、ハイパーバイザ機能の実行、および vSwitch を使用した NIC と vNIC の間のパケット転送に使用されています。

Rackspace CPU 使用率レポート

使用可能なクラウドサーバーの CPU、RAM、およびディスク容量の構成情報の表示に加えて、ディスク、I/O、およびネットワーク情報も表示できます。この情報を使用して、ニーズに適したクラウドサーバーを決定してください。コマンドライン nova クライアントまたは **Cloud Control Panel** インターフェイスを使用して、使用可能なサーバーを表示できます。

コマンドラインで、次のコマンドを実行します。

```
nova flavor-list
```

使用可能なすべてのサーバー構成が表示されます。リストには、次の情報が含まれています。

- ID：サーバー構成 ID
- 名前：RAM サイズとパフォーマンスタイプでラベル付けされた構成名
- Memory_MB：構成の RAM の量
- ディスク：GB 単位のディスクサイズ（汎用クラウドサーバーの場合、システムディスクのサイズ）

- エフェメラル：データディスクのサイズ
- スワップ：スワップ領域のサイズ
- VCPU：構成に関連付けられた仮想 CPU の数
- RXTX_Factor：サーバーに接続された PublicNet ポート、ServiceNet ポート、および分離されたネットワーク（クラウドネットワーク）に割り当てられる帯域幅の量（Mbps 単位）
- Is_Public：未使用

ASA Virtual と Rackspace のグラフ

ASA Virtual と Rackspace の間には CPU % の数値に違いがあります。

- Rackspace グラフの数値は ASA Virtual の数値よりも常に大きくなります。
- Rackspace ではこの値は「%CPU usage」と呼ばれ、ASA Virtual ではこの値は「%CPU utilization」と呼ばれます。

用語「%CPU utilization」と「%CPU usage」は別のものを意味しています。

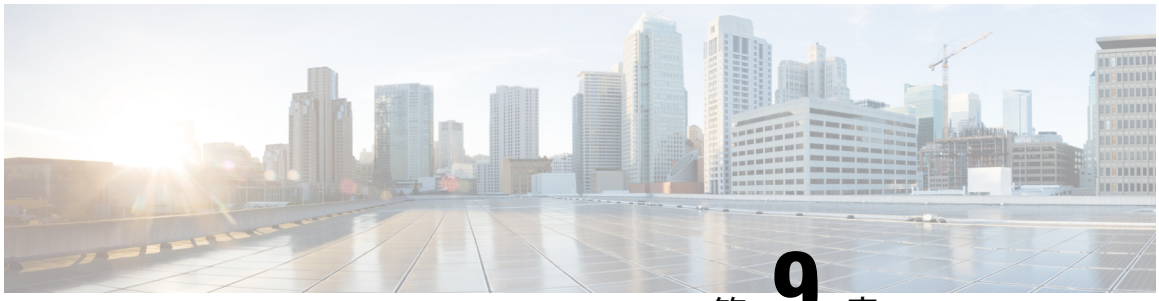
- CPU utilization は、物理 CPU の統計情報を提供します。
- CPU usage は CPU のハイパースレッディングに基づいた論理 CPU の統計情報を提供します。しかし、1 つの vCPU のみが使用されるため、ハイパースレッディングは動作しません。

Rackspace では「%CPU usage」は次のように計算されます。

アクティブに使用された仮想 CPU の量。使用可能な CPU の合計に対する割合として指定されます。

この計算は、ホストから見た CPU 使用率であり、ゲスト オペレーティング システムから見た CPU 使用率ではありません。また、これは仮想マシンで使用可能なすべての仮想 CPU の平均 CPU 使用率になります。

たとえば、1 個の仮想 CPU を搭載した 1 つの仮想マシンが、4 個の物理 CPU を搭載した 1 台のホストで実行されており、その CPU 使用率が 100% の場合、仮想マシンは、1 個の物理 CPU をすべて使用しています。仮想 CPU の使用率は、「MHz 単位の使用率 / 仮想 CPU の数 x コア周波数」として計算されます。



第 9 章

Hyper-V を使用した ASAv の導入

Microsoft Hyper-V を使用して ASAv を導入できます。

重要 9.13(1) 以降は、ASAv の最小メモリ要件は 2GB です。現在の ASAv が 2GB 未満のメモリで動作している場合、ASAvマシンのメモリを増やさないと、以前のバージョンから9.13(1)以降にアップグレードできません。また、バージョン 9.13(1) を使用して新しい ASAv マシンを再導入できます。

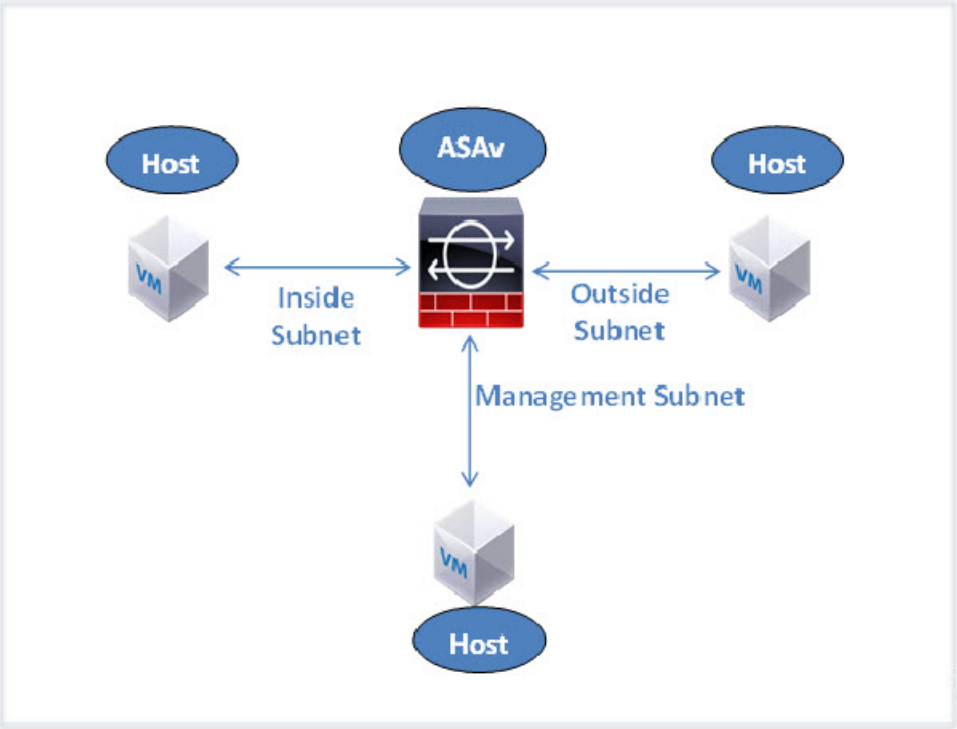
Hyper-V を使用した ASAv の導入について

スタンドアロンの Hyper-V サーバー上に、または Hyper-V マネージャを介して Hyper-V を導入できます。PowerShell CLI コマンドを使用したインストール手順については、「コマンドラインを使用した Hyper-V への ASAv のインストール」（46 ページ）を参照してください。Hyper-V マネージャを使用したインストール手順については、「Hyper-V マネージャを使用した Hyper-V への ASAv のインストール」（46 ページ）を参照してください。Hyper-V はシリアル コンソール オプションを提供していません。管理インターフェイスを介して SSH または ASDM を通じ

て Hyper-V を管理できます。SSH の設定については、「SSH の設定」の 54 ページを参照してください。

次の図は、ルーテッド ファイアウォール モードでの ASAv の推奨トポロジを示しています。ASAv 向けに Hyper-V でセットアップされた 3 つのサブネット（管理、内部、および外部）があります。

図 *37* : ルーテッド ファイアウォール モードの *ASAv* の推奨トポロジ



ASAv および Hyper-V のガイドラインと制限事項

- プラットフォーム サポート
  - Cisco UCS B シリーズ サーバー
  - Cisco UCS C シリーズ サーバー
  - Hewlett Packard Proliant DL160 Gen8
- サポートされる OS
  - Windows Server 2012
  - ネイティブ Hyper-V

（注） ASAv は現在、仮想化に使用されている最新の 64 ビット高性能プラットフォームで稼働します。

- ファイル形式

  Hyper-V への ASAv の初期導入では、VHDX 形式がサポートされています。

- 第 0 日用（Day 0）構成

  必要な ASA CLI 設定コマンドを含むテキスト ファイルを作成します。手順については、「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。

- 第 0 日用構成のファイアウォール トランスペアレント モード

  設定行「firewall transparent」は、第 0 日用コンフィギュレーション ファイルの先頭に配置する必要があります。ファイル内のそれ以外の場所にあると、異常な動作が起きる場合があります。手順については、「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。

- フェールオーバー

  Hyper-V 上の ASAv はアクティブ/スタンバイフェールオーバーをサポートしています。ルーテッド モードとトランスペアレント モードの両方でアクティブ/スタンバイ フェールオーバーを実行するには、すべての仮想ネットワーク アダプタで MAC アドレス スプーフィングを有効化する必要があります。「MAC アドレス スプーフィングの設定」の 53 ページを参照してください。スタンドアロン ASAv のトランスペアレントモードの場合、管理インターフェイスの MAC アドレススプーフィングは有効にしないでください。アクティブ/アクティブ フェールオーバーはサポートされていません。

- Hyper-V は最大 8 つのインターフェイスをサポートします。Management 0/0 および GigabitEthernet 0/0 ～ 0/6。フェールオーバー リンクとして GigabitEthernet を使用できます。

- VLANs

  トランク モードでインターフェイスに VLAN を設定するには、**Set-VMNetworkAdapterVLan** Hyper-V Powershell コマンドを使用します。管理インターフェイスの NativeVlanID は、特定の VLAN として、または VLAN がない場合は「0」として設定できます。トランク モードは、Hyper-V ホストをリブートした場合は保持されません。各リブート後に、トランク モードを再設定する必要があります。

- レガシー ネットワーク アダプタはサポートされていません。

- 第 2 世代仮想マシンはサポートされていません。

- Microsoft Azure はサポートされていません。

ASAv と Hyper-V の前提条件

- MS Windows 2012 に Hyper-V をインストールします。
- 第 0 日用コンフィギュレーション テキスト ファイルを使用する場合は、それを作成します。

  ASAv の初回導入前に、第 0 日用構成を追加する必要があります。追加しない場合は、第 0 日用構成を使用するために、ASAv から write erase を実行する必要があります。手順については、「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。
- Cisco.com から ASAv VHDX ファイルをダウンロードします。

  http://www.cisco.com/go/asa-software

（注） Cisco.com のログインおよびシスコ サービス契約が必要です。

- Hyper-V スイッチには、3 つ以上のサブネット/VLAN が構成されます。
- Hyper-V システム要件については、Cisco ASA の互換性 [英語] を参照してください。

第 0 日のコンフィギュレーション ファイルの準備

ASAv を起動する前に、第 0 日用のコンフィギュレーション ファイルを準備できます。このファイルは、ASAv の起動時に適用される ASAv の設定を含むテキストファイルです。この初期設定は、「day0-config」というテキスト ファイルとして指定の作業ディレクトリに格納され、さらに day0.iso ファイルへと処理されます。この day0.iso ファイルが最初の起動時にマウントされて読み取られます。第 0 日用コンフィギュレーション ファイルには、少なくとも、管理インターフェイスをアクティブ化するコマンドと、公開キー認証用 SSH サーバーをセットアップするコマンドを含める必要がありますが、すべての ASA 設定を含めることもできます。day0.iso ファイル（カスタム day0 またはデフォルトの day0.iso）は、最初の起動中に使用できなければなりません。

始める前に

この例では Linux が使用されていますが、Windows の場合にも同様のユーティリティがあります。

- 初期導入時に自動的に ASAv にライセンスを付与するには、Cisco Smart Software Manager からダウンロードした Smart Licensing Identity（ID）トークンを「idtoken」というテキスト ファイルに格納し、第 0 日用構成ファイルと同じディレクトリに保存します。
- トランスペアレントモードで ASAv を導入する場合は、トランスペアレントモードで実行される既知の ASA 構成ファイルを、第 0 日用構成ファイルとして使用する必要がありま

す。これは、ルーテッド ファイアウォールの第 0 日用コンフィギュレーション ファイルには該当しません。

- ASAv の初回起動前に、第 0 日用構成ファイルを追加する必要があります。ASAv の初回起動後に第 0 日用構成ファイルを使用する場合は、**write erase** コマンドを実行し、第 0 日用構成 ファイルを適用してから、ASAv を起動する必要があります。

ステップ 1 「day0-config」というテキスト ファイルに ASAv の CLI 設定を記入します。3 つのインターフェイスの設定とその他の必要な設定を追加します。

最初の行は ASA のバージョンで始める必要があります。day0-config は、有効な ASA 構成である必要があります。day0-config を生成する最適な方法は、既存の ASA または ASAv から実行コンフィギュレーションの必要な部分をコピーする方法です。day0-config 内の行の順序は重要で、既存の show run コマンド出力の順序と一致している必要があります。

例：

```
ASA Version 9.5.1
!
interface management0/0
nameif management
 security-level 100
 ip address 192.168.1.2 255.255.255.0
 no shutdown
interface gigabitethernet0/0
 nameif inside
 security-level 100
 ip address 10.1.1.2 255.255.255.0
 no shutdown
interface gigabitethernet0/1
 nameif outside
 security-level 0
 ip address 198.51.100.2 255.255.255.0
 no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
```

ステップ 2 （任意）Cisco Smart Software Manager により発行された Smart License ID トークン ファイルをコンピュータにダウンロードします。

ステップ 3 （任意）ダウンロードしたファイルから ID トークンをコピーし、ID トークンのみを含むテキスト ファイルを作成します。

ステップ 4 （任意）ASAv の初期導入時に自動的にライセンスを許諾する場合は、day0-config ファイルに次の情報が含まれていることを確認してください。

- 管理インターフェイスの IP アドレス
- （任意）SSmart Licensing で使用する HTTP プロキシ
- HTTP プロキシ（指定した場合）または tools.cisco.com への接続を有効にする route コマンド
- tools.cisco.com を IP アドレスに解決する DNS サーバー

- 要求する ASAv ライセンスを指定するための Smart Licensing の設定
- （任意）CSSM での ASAv の検索を容易にするための一意のホスト名

ステップ 5 テキスト ファイルを ISO ファイルに変換して仮想CD-ROM を生成します。

```
stack@user-ubuntu:-/KvmAsa$ sudo genisoimage -r -o day0.iso day0-config idtoken
I: input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 252
Total directory bytes: 0
Path table size (byptes): 10
Max brk space used 0
176 extents written (0 MB)
stack@user-ubuntu:-/KvmAsa$
```

この ID トークンによって、Smart Licensing サーバーに ASAv が自動的に登録されます。

ステップ 6 ステップ 1 から 5 を繰り返し、導入する ASAv ごとに、適切な IP アドレスを含むデフォルトの構成ファイルを作成します。

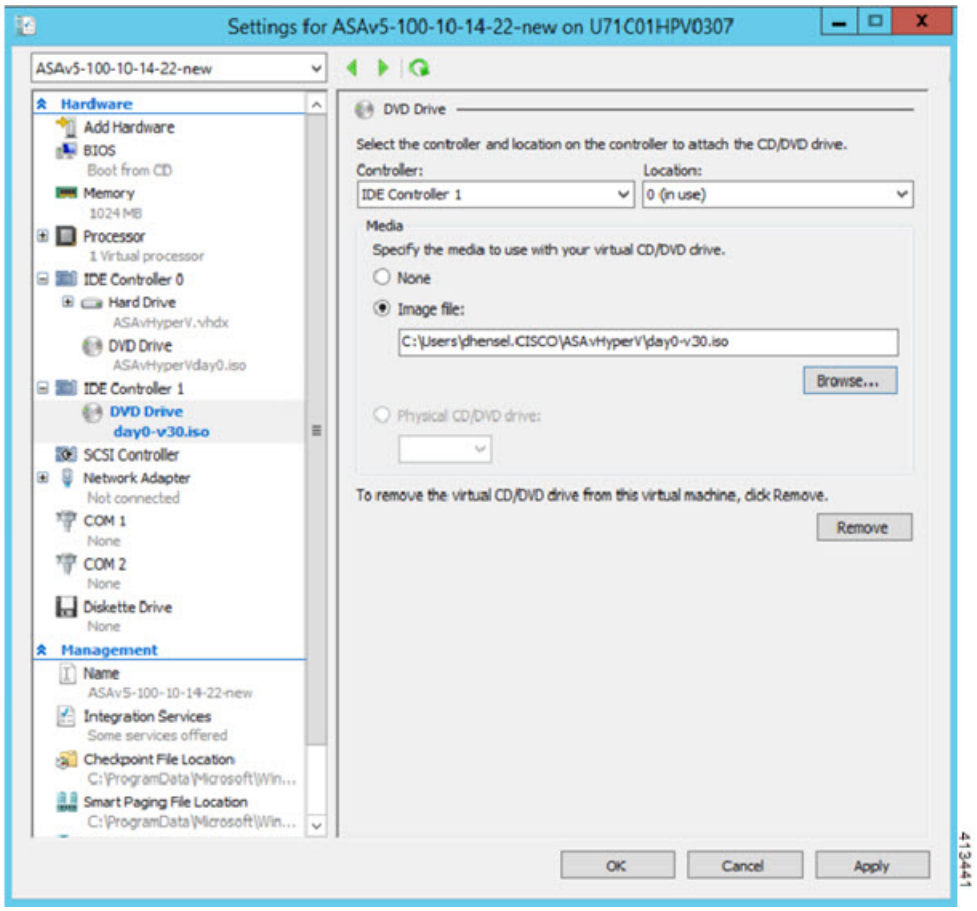
Hyper-V マネージャを使用した ASAv と第 0 日用構成ファイルの導入

第 0 日用コンフィギュレーション ファイルを設定したら（「第 0 日のコンフィギュレーション ファイルの準備」）、Hyper-V マネージャを使用して導入できます。

ステップ 1 **[Server Manager]** > **[Tools]** > **[Hyper-V Manager]** に移動します。

ステップ 2 Hyper-V マネージャの右側にある [Settings] をクリックします。[Settings] ダイアログボックスが開きます。左側の [Hardware] の下で、[IDE Controller 1] をクリックします。

図 *38 : Hyper-V* マネージャ



ステップ 3 右側のペインの [Media] の下で、[Image file] のラジオ ボタンを選択して、第 0 日用 ISO コンフィギュレーション ファイルを保存するディレクトリを参照し、[Apply] をクリックします。ASAv は、初回起動時に、第 0 日用構成ファイルの内容に基づいて構成されます。

コマンドラインを使用した Hyper-V への ASAv のインストール

Windows PowerShell コマンドラインを介して Hyper-V に ASAv をインストールできます。スタンドアロンの Hyper-V サーバー上にいる場合は、コマンド ラインを使用して Hyper-V をインストールする必要があります。

ステップ 1 Windows Powershell を開きます。

ステップ 2 ASAv を導入します。

例：

```
new-vm -name $fullVMName -MemoryStartupBytes $memorysize -Generation 1 -vhdpath
C:\Users\jsmith.CISCO\ASAvHyperV\$ImageName.vhdx -Verbose
```

ステップ 3 ASAv のモデルに応じて、CPU 数をデフォルトの 1 から変更します。

例：

```
set-vm -Name $fullVMName -ProcessorCount 4
```

ステップ 4 （任意）インターフェイス名をわかりやすい名前に変更します。

例：

```
Get-VMNetworkAdapter -VMName $fullVMName -Name "Network Adapter" | Rename-vmNetworkAdapter -NewName
mgmt
```

ステップ 5 （任意）ネットワークで必要な場合は、VLAN ID を変更します。

例：

```
Set-VMNetworkAdapterVlan -VMName $fullVMName -VlanId 1151 -Access -VMNetworkAdapterName "mgmt"
```

ステップ 6 Hyper-V が変更を反映するように、インターフェイスを更新します。

例：

```
Connect-VMNetworkAdapter -VMName $fullVMName -Name "mgmt" -SwitchName 1151mgmtswitch
```

ステップ 7 内部インターフェイスを追加します。

例：

```
Add-VMNetworkAdapter -VMName $fullVMName -name "inside" -SwitchName 1151mgmtswitch
Set-VMNetworkAdapterVlan -VMName $fullVMName -VlanId 1552 -Access -VMNetworkAdapterName "inside"
```

ステップ 8 外部インターフェイスを追加します。

例：

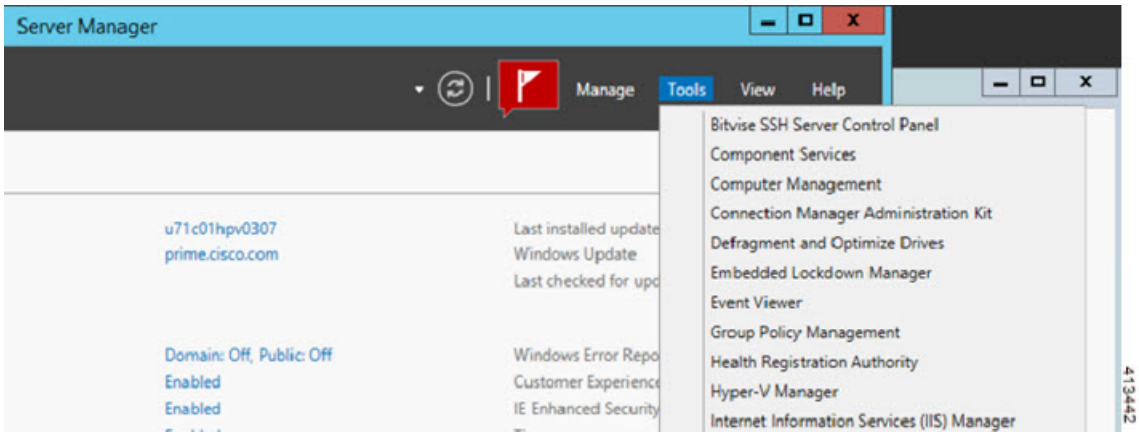
```
Add-VMNetworkAdapter -VMName $fullVMName -name "outside" -SwitchName 1151mgmtswitch
Set-VMNetworkAdapterVlan -VMName $fullVMName -VlanId 1553 -Access -VMNetworkAdapterName “outside"
```

Hyper-V マネージャを使用した Hyper-V への ASAv のインストール

Hyper-V マネージャを使用して、Hyper-V に ASAv をインストールできます。

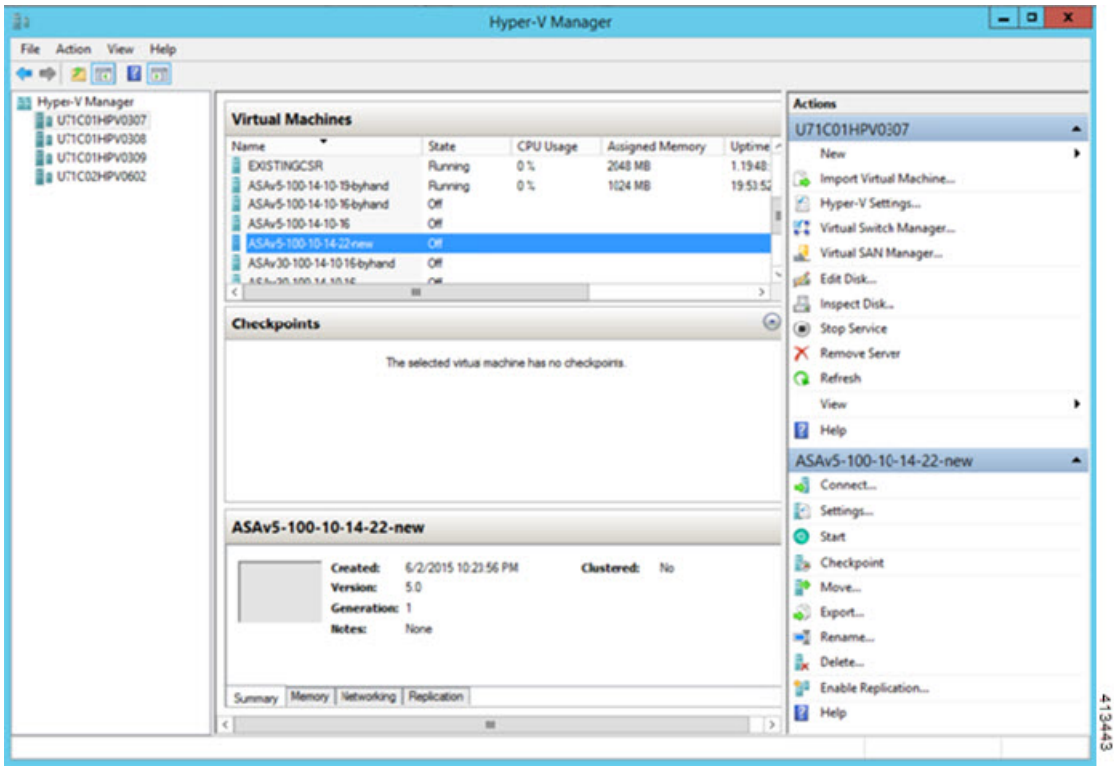
ステップ 1 **[Server Manager]** > **[Tools]** > **[Hyper-V Manager]** に移動します。

図 *39 : Server Manager*



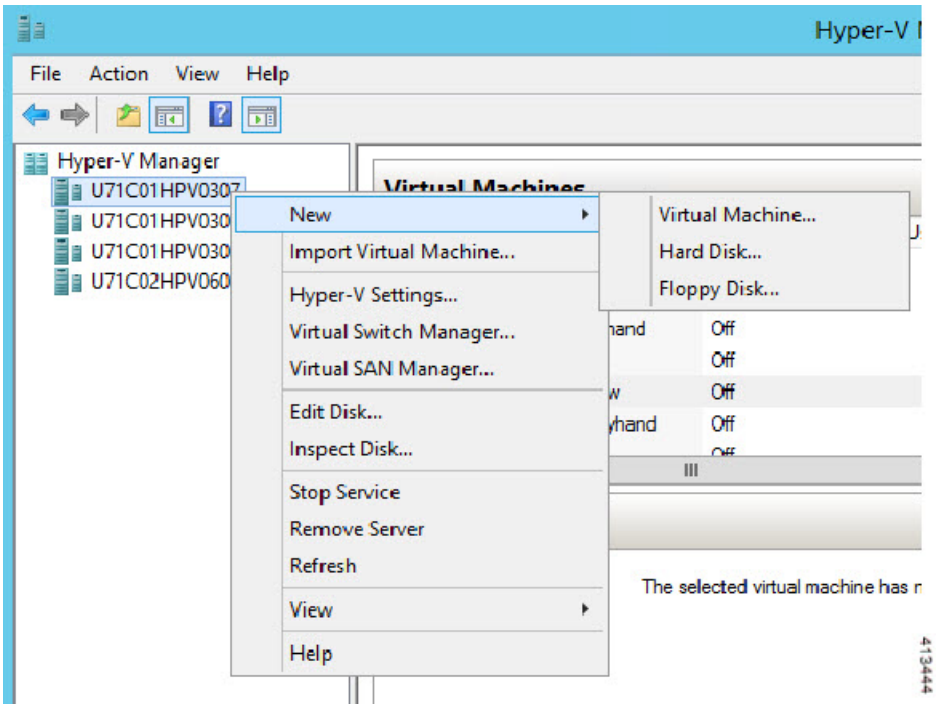
ステップ 2 Hyper-V マネージャが表示されます。

図 *40 : Hyper-V* マネージャ



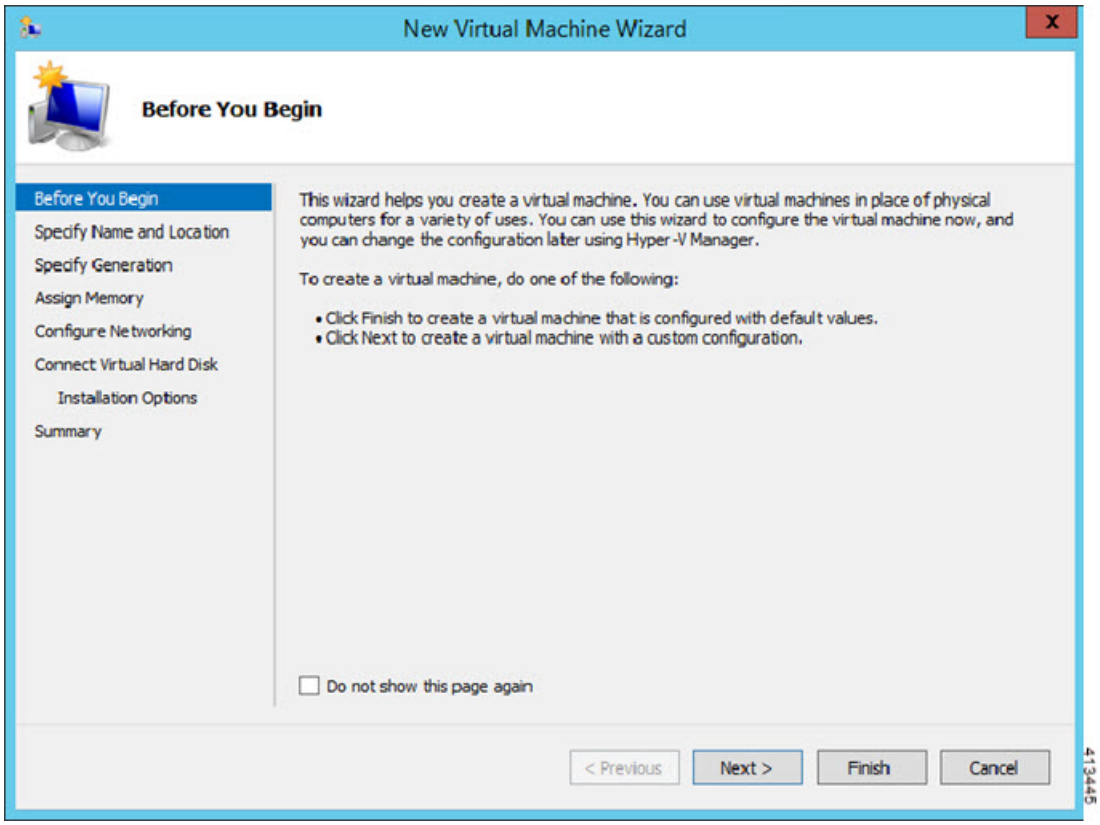
ステップ 3 右側のハイパーバイザのリストから、目的のハイパーバイザを右クリックし、**[New]** > **[Virtual Machine]** を選択します。

図 *41* : 新規仮想マシンの起動



ステップ 4 [New Virtual Machine] ウィザードが表示されます。

図 *42 : [New Virtual Machine]* ウィザード



ステップ 5 ウィザードを通じて作業し、次の情報を指定します。

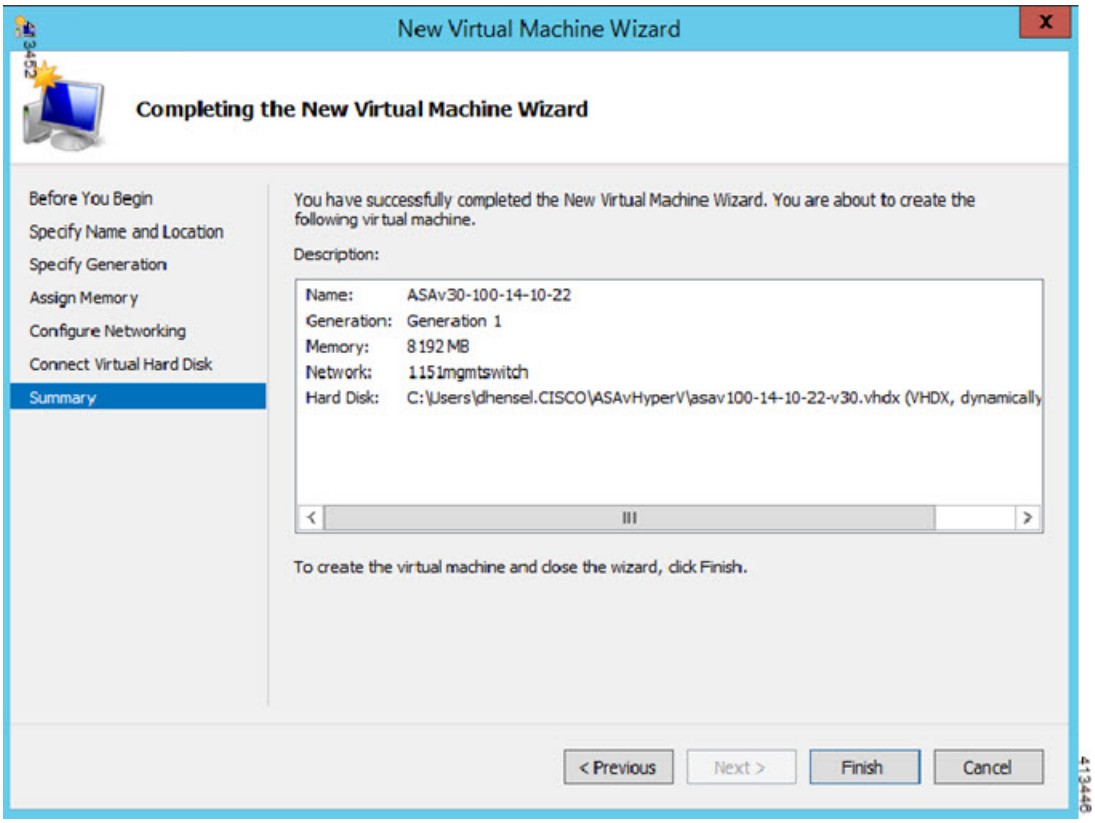
- ASAv の名前と場所
- ASAv の世代

  ASAv でサポートされている唯一の世代は [世代1（Generation 1）] です。
- ASAv のメモリ量（100Mbps の場合は 1024 MB、1Gbps の場合は 2048 MB、2Gbps の場合は 8192 MB）
- ネットワーク アダプタ（セットアップ済みの仮想スイッチに接続）
- 仮想ハード ディスクと場所

  [Use an existing virtual hard disk] を選択し、VHDX ファイルの場所を参照します。

ステップ 6 [終了（Finish）] をクリックすると、ASAv 構成を示すダイアログボックスが表示されます。

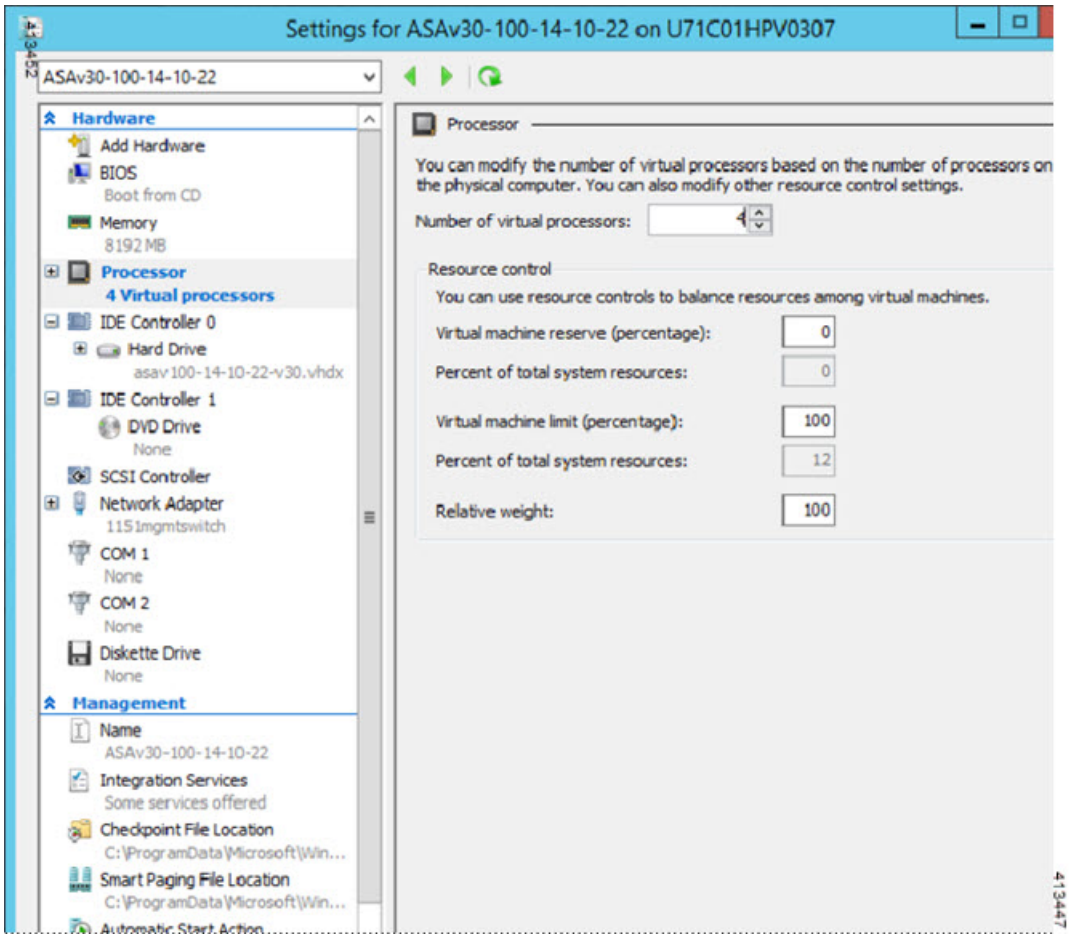
図 *43* : 新規仮想マシンの概要



ステップ 7 ASAv に 4 つの vCPU がある場合は、ASAv を起動する前に、vCPU 値を変更する必要があります。Hyper-V マネージャの右側にある [Settings] をクリックします。[Settings] ダイアログボックスが開きます。左側の [Hardware] メニューで、[Processor] をクリックし、[Processor] ペインを表示します。[Number of virtual processors] を 4 に変更します。

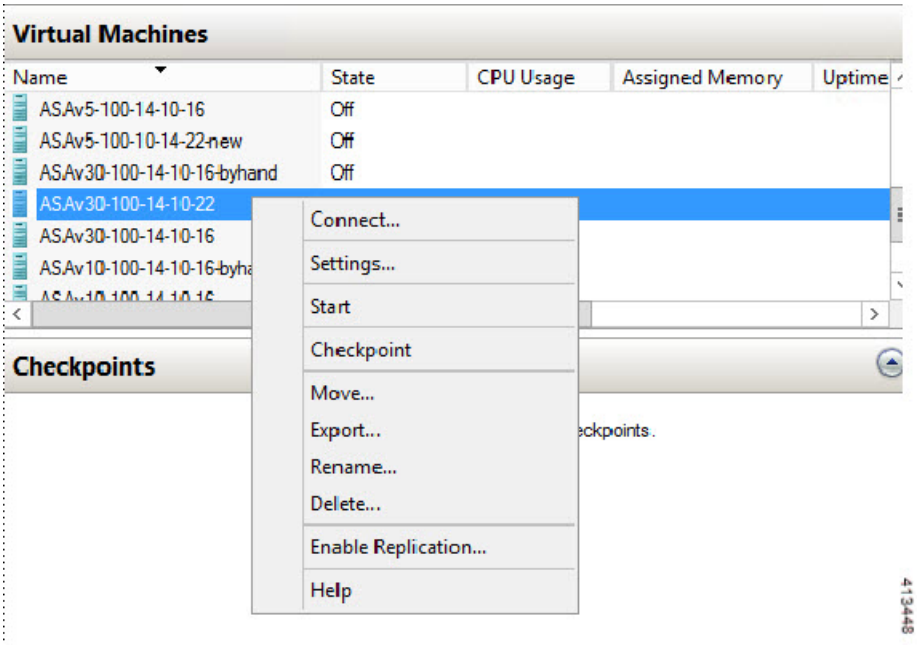
100Mbps および 1Gbps の権限付与では 1 個の vCPU、2Gbps の権限付与では 4 個の vCPU となります。デフォルトは 1 です。

図 *44* : 仮想マシンのプロセッサの設定



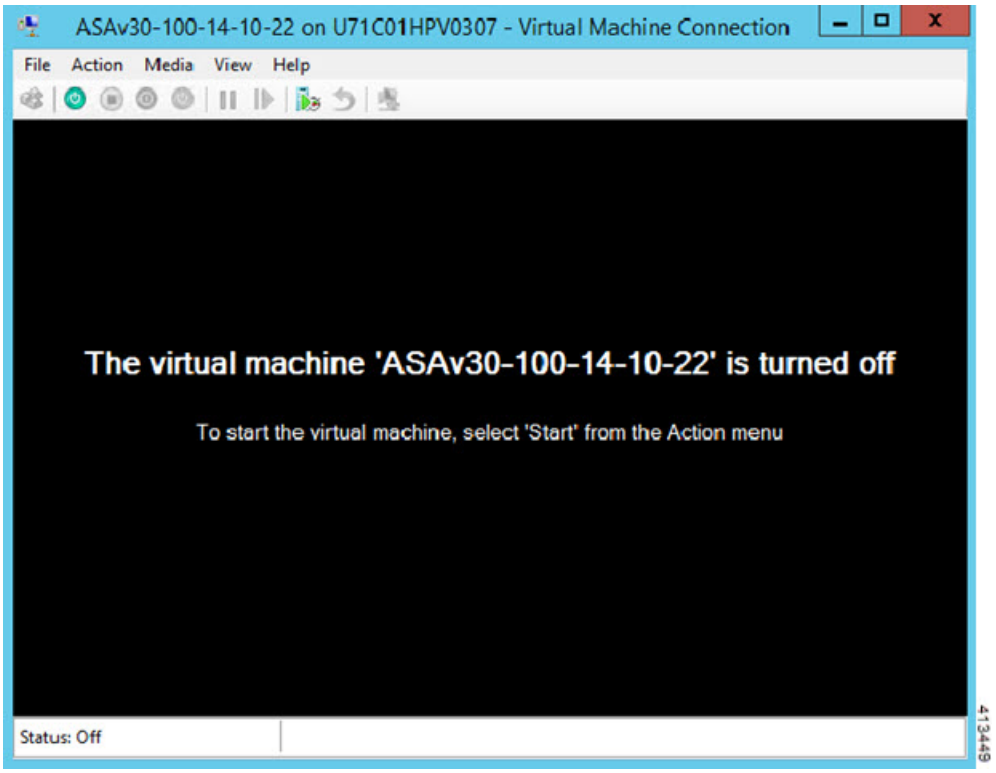
ステップ 8 [仮想マシン（Virtual Machines）] メニューで、リスト内の ASAv の名前を右クリックし、[接続（Connect）] をクリックして、ASAv に接続します。コンソールが開き、停止されている ASAv が表示されます。

図 *45* : 仮想マシンへの接続



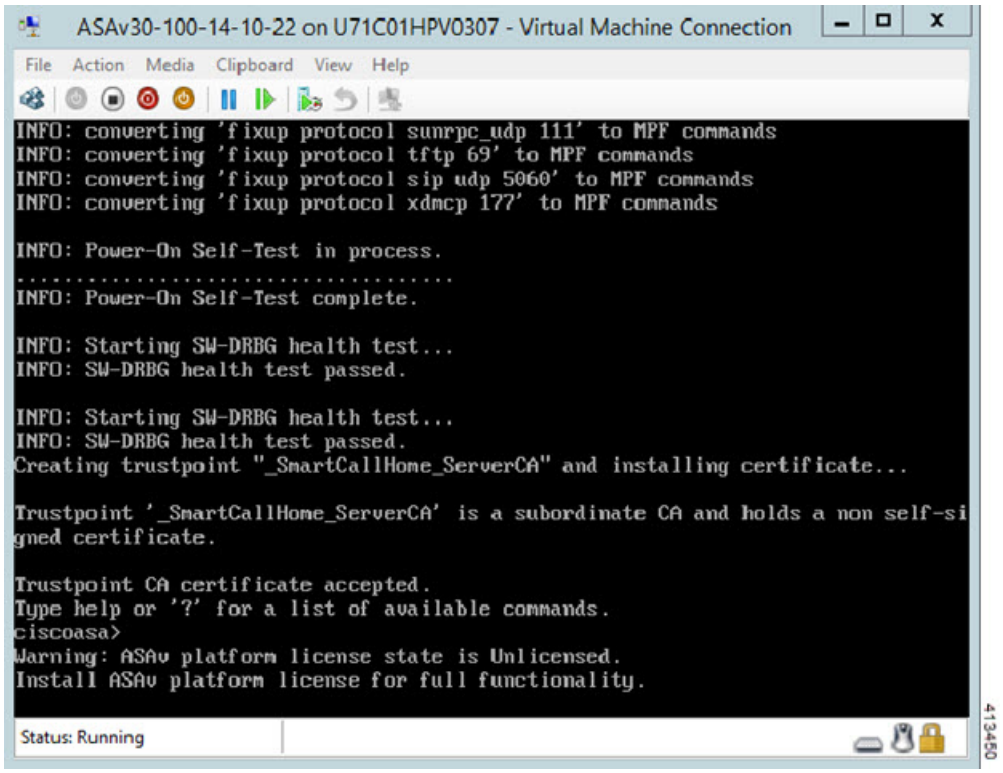
ステップ 9 [仮想マシンの接続（Virtual Machine Connection）] コンソールウィンドウで、青緑色の開始ボタンをクリックして、ASAv を起動します。

図 *46* : 仮想マシンの開始



ステップ 10 ASAv の起動の進行状況がコンソールに表示されます。

図 47 : 仮想マシンの起動の進行状況



Hyper-V マネージャからのネットワーク アダプタの追加

新しく導入された ASAv のネットワークアダプタは 1 つだけです。さらに 2 つ以上のネットワーク アダプタを追加する必要があります。この例では、内部ネットワーク アダプタを追加します。

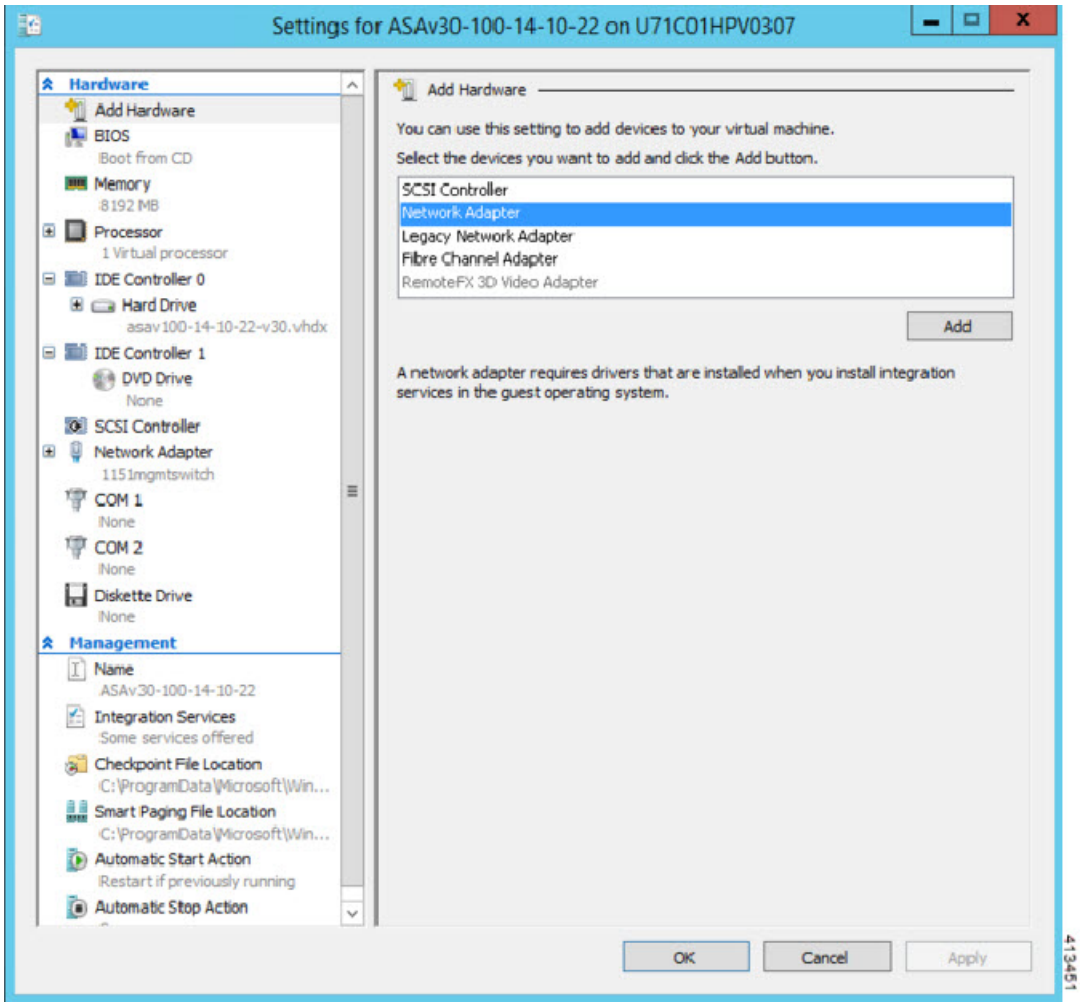
始める前に

- ASAv はオフ状態である必要があります。

ステップ 1 Hyper-V マネージャの右側にある [Settings] をクリックします。[Settings] ダイアログボックスが開きます。左側の [Hardware] メニューで、[Add Hardware] をクリックし、次に [Network Adapter] をクリックします。

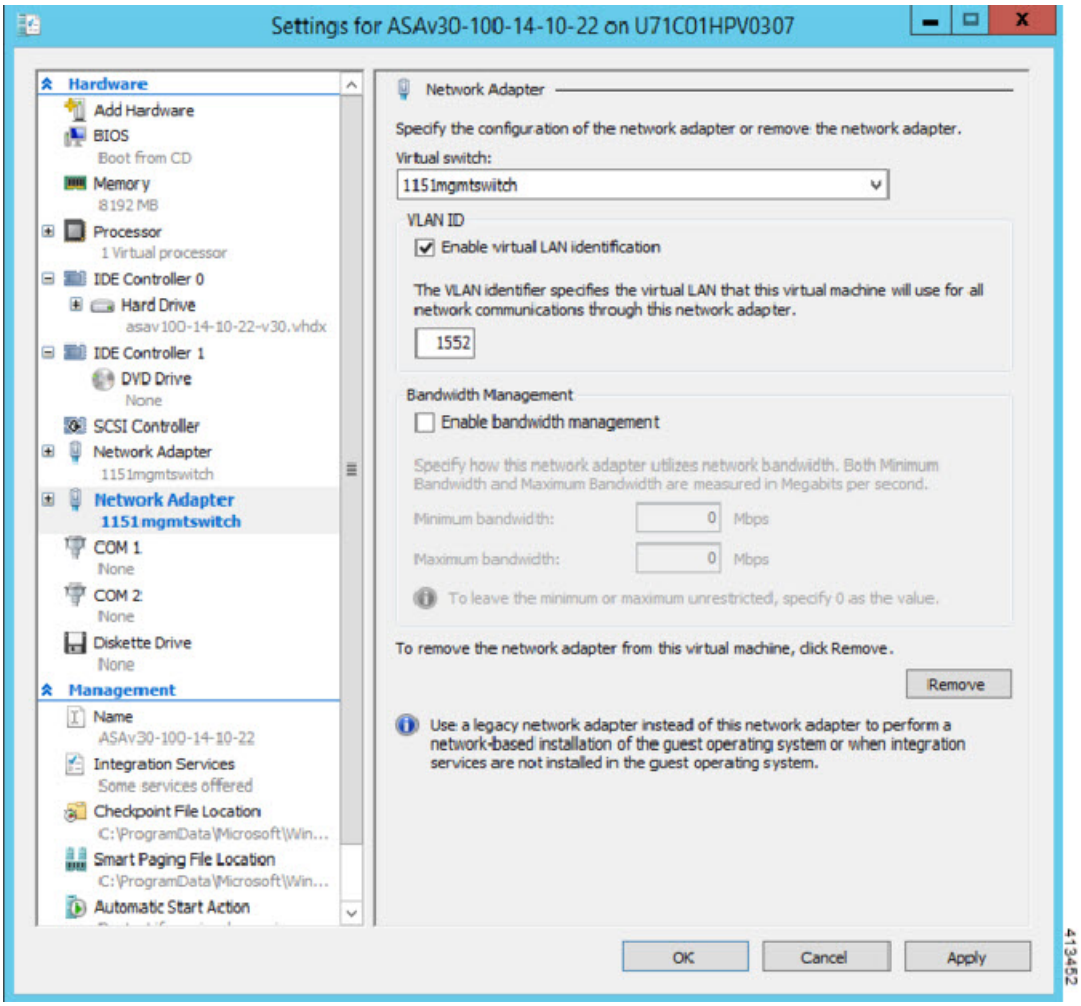
（注） レガシー ネットワーク アダプタを使用しないでください。

図 *48* : ネットワーク アダプタの追加



ステップ 2 ネットワーク アダプタの追加後、仮想スイッチとその他の機能を変更できます。また、必要に応じてVLAN ID を設定できます。

図 *49* : ネットワーク アダプタ設定の変更



ネットワーク アダプタの名前の変更

Hyper-V では、「Network Adapter」という汎用ネットワーク インターフェイス名が使用されます。このため、ネットワーク インターフェイスがすべて同じ名前であると、紛らわしい場合があります。Hyper-V マネージャを使用して名前を変更することはできません。Windows Powershell コマンドを使用して変更する必要があります。

ステップ 1 Windows Powershell を開きます。

ステップ 2 必要に応じてネットワーク アダプタを変更します。

例：

```
$NICRENAME= Get-VMNetworkAdapter -VMName 'ASAvVM' -Name "Network Adapter"
rename-VMNetworkAdapter -VMNetworkAdapter $NICRENAME[0] -newname inside
rename-VMNetworkAdapter -VMNetworkAdapter $NICRENAME[1] -newname outside
```

MAC アドレス スプーフィング

ASAvがトランスペアレントモードでパケットを渡し、HAアクティブ/スタンバイフェールオーバーに対応できるように、すべてのインターフェイスの MAC アドレススプーフィングを有効にする必要があります。Hyper-V マネージャ内で、または Powershell コマンドを使用して、これを実行できます。

Hyper-V マネージャを使用した MAC アドレス スプーフィングの設定

Hyper-V マネージャを使用して、MAC スプーフィングを Hyper-V に設定できます。

ステップ 1 **[Server Manager]** > **[Tools]** > **[Hyper-V Manager]** に移動します。

Hyper-V マネージャが表示されます。

ステップ 2 Hyper-V マネージャの右側の [Settings] をクリックして、設定ダイアログ ボックスを開きます。

ステップ 3 左側の [Hardware] メニューで次の操作をします。

1. [Inside] をクリックして、メニューを展開します。
2. [Advanced Features] をクリックして、MAC アドレス オプションを表示します。
3. [Enable MAC address spoofing] ラジオ ボタンをクリックします。

ステップ 4 外部インターフェイスでも、この手順を繰り返します。

コマンド ラインを使用した MAC アドレス スプーフィングの設定

Windows Powershell コマンド ラインを使用して、MAC スプーフィングを Hyper-V に設定できます。

ステップ 1 Windows Powershell を開きます。

ステップ 2 MAC アドレス スプーフィングを設定します。

例：

```
Set-VMNetworkAdapter -VMName $vm_name\
-ComputerName $computer_name -MacAddressSpoofing On\
-VMNetworkAdapterName $network_adapter\r"
```

SSH の設定

Hyper-V マネージャの [仮想マシンの接続（Virtual Machine Connection）] から管理インターフェイスを介して SSH アクセスできるように ASAv を設定できます。第 0 日用コンフィギュレーション ファイルを使用している場合は、ASAv への SSH アクセスを追加できます。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。

ステップ 1 RSAキー ペアが存在することを確認します。

例：

```
asav# show crypto key mypubkey rsa
```

ステップ 2 RSAキー ペアがない場合は、RSAキー ペアを生成します。

例：

```
asav(conf t)# crypto key generate rsa modulus 2048

username test password test123 privilege 15
aaa authentication ssh console LOCAL
ssh 10.7.24.0 255.255.255.0 management
ssh version 2
```

ステップ 3 別の PC から SSH を使用して ASAv にアクセスできることを確認します。

CPU 使用率とレポート

CPU使用率レポートには、指定された時間内に使用されたCPUの割合の要約が表示されます。通常、コアはピーク時以外には合計 CPU 容量の約 30 ～ 40% で動作し、ピーク時は約 60 ～ 70% の容量で動作します。

ASA Virtual の vCPU 使用率

ASA Virtual の vCPU 使用率には、データパス、制御ポイント、および外部プロセスで使用されている vCPU の量が表示されます。

Hyper-V で報告される vCPU 使用率には、ASA Virtual の使用率に加えて、次のものが含まれます。

- ASA Virtual アイドル時間

- ASA Virtual マシンに使用された %SYS オーバーヘッド

CPU 使用率の例

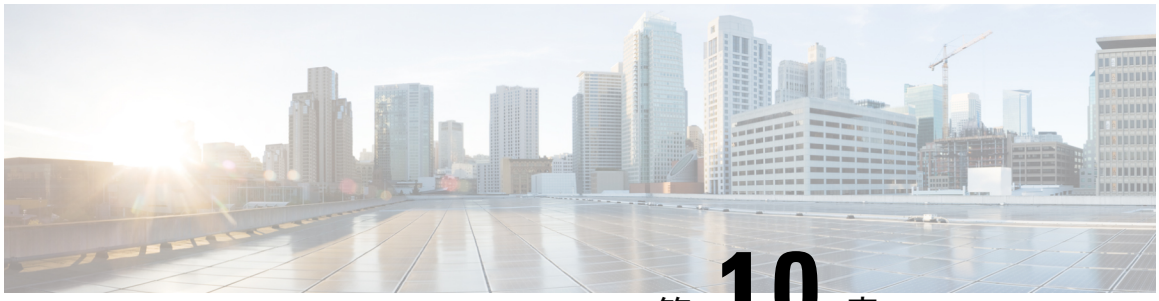
CPU 使用率の統計情報を表示するには、**show cpu usage** コマンドを使用します。

例

```
Ciscoasa#show cpu usage

CPU □□□□5□□□ 1%□1 □□□ 2%□5 □□□ 1%
```

報告された vCPU の使用率が大幅に異なる例を次に示します。

- ASA Virtual レポート：40%
- DP：35%
- 外部プロセス：5%
- ASA（ASA Virtual レポート）：40%
- ASA アイドル ポーリング：10%
- オーバーヘッド：45%



第 10 章

Oracle Cloud Infrastructure への ASAv の展開

Oracle Cloud Infrastructure（OCI）に ASAv を導入できます。

OCI への ASAv の展開について

OCI は、オラクル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。

ASAv は、物理 ASAv 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。ASAv は、パブリック OCI で展開できます。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。

OCI のコンピューティングシェイプ

シェイプは、インスタンスに割り当てられる CPU の数、メモリの量、およびその他のリソースを決定するテンプレートです。ASAv は、次の「標準：汎用」の OCI シェイプタイプをサポートします。

表 *21 :* でサポートされるコンピューティングシェイプ *ASAv*

仮想マシンシェイプ	属性		インターフェイス
	oCPU	メモリ（GB）	
VM.Standard2.4	4	60	最小 3 つ、最大 4 つ
VM.Standard2.8	8	120	最小 3 つ、最大 8 つ

- ASAv には、少なくとも 3 つのインターフェイスが必要です。
- OCI では、1 つの oCPU は 2 つの vCPU に相当します。
- サポートされる vCPU の最大数は 16（8 個の oCPU）です。

ユーザーは、OCI でアカウントを作成し、Oracle Cloud Marketplace の Cisco ASA 仮想ファイアウォール（ASAv）製品を使用してコンピューティング インスタンスを起動し、OCI のシェイプを選択します。

ASAv と OCI の前提条件

- https://www.oracle.com/cloud/sign-in.html でアカウントを作成します。
- ASAv へのライセンス付与。ASAv 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。「Licenses: Smart Software Licensing」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
  - 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 ASAv 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 ASAv 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）：ASAv 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）：DMZ ネットワークに ASAv を接続するために使用されます。
- ASAv システム要件については、Cisco ASA の互換性 [英語] を参照してください。

ASAv および OCI のガイドラインと制限事項

サポートされる機能

OCI 上の ASAv は、次の機能をサポートしています。

- OCI 仮想クラウドネットワーク（VCN）での展開
- インスタンスあたり最大 16 個の vCPU（8 個の oCPU）
- 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
- Single Root I/O Virtualization（SR-IOV）をサポート

サポートされない機能

OCI 上の ASAv は、次の機能をサポートしていません。

- ASAv 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
- マルチ コンテキスト モード

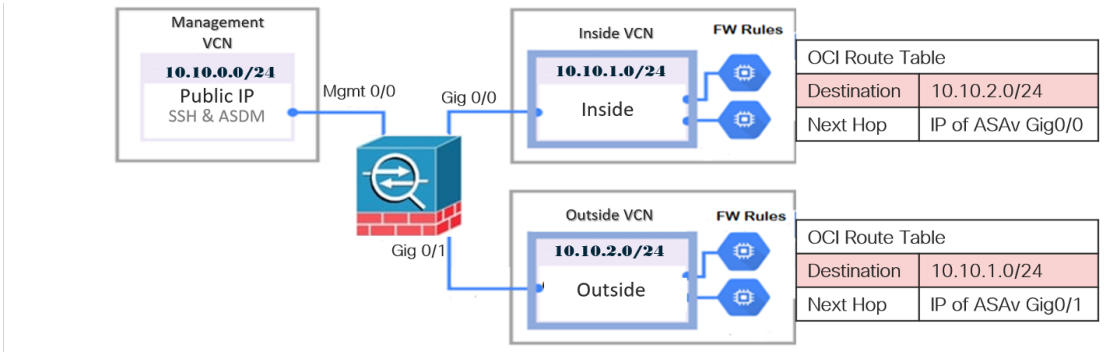
制限事項

- OCI に ASAv を展開する場合、Mellanox 5 は SR-IOV モードの vNIC としてサポートされません。
- 静的設定と DHCP 設定の両方で ASAv に必要な個別のルーティングルール。

OCI 上の ASAv のネットワークトポロジの例

次の図は、ASAv 用の 3 つのサブネット（管理、内部、外部）が OCI 内に設定されているルーテッド ファイアウォール モードの ASAv の推奨ネットワークトポロジを示しています。

図 *50 : OCI* 上の *ASAv* の展開例



OCI への ASAv の導入

次の手順では、OCI 環境を準備し、ASAv インスタンスを起動する方法について説明します。OCI ポータルにログインし、OCI Marketplace で Cisco ASA 仮想ファイアウォール（ASAv）製品を検索し、コンピューティングインスタンスを起動します。ASAv の起動後に、トラフィックの送信元と接続先に応じて、トラフィックをファイアウォールに転送するようにルートテーブルを設定する必要があります。

仮想クラウドネットワーク（VCN）の作成

ASAv 展開用の仮想クラウドネットワーク（VCN）を設定します。少なくとも、ASAv の各インターフェイスに 1 つずつ、合計 3 つの VCN が必要です。

次の手順に進み、管理 VCN を完了できます。次に、[Networking] に戻り、内部インターフェイスおよび外部インターフェイスの VCN を作成します。

始める前に

（注） ナビゲーションメニューからサービスを選択すると、左側のメニューにコンパートメントリストが表示されます。コンパートメントはリソースの整理に役立ち、リソースへのアクセスを制御しやすくなります。ルートコンパートメントは、テナントがプロビジョニングされるときに Oracle によって作成されます。管理者は、ルートコンパートメントにさらに多くのコンパートメントを作成し、アクセスルールを追加して、どのユーザーがそれらのコンパートメントを表示してアクションを実行できるかを制御できます。詳細については、Oracle のドキュメント『コンパートメントの管理（Managing Compartments）』を参照してください。

ステップ 1 OCI にログインし、地域を選択します。

OCI 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、画面の右上隅に表示されます。ある地域内のリソースは、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 **[Networking]** > **[Virtual Cloud Networks]** を選択し、[Create Virtual Cloud Networks] をクリックします。

ステップ 3 [Name] に、VCN のわかりやすい名前を入力します（例：*ASAvManagement*）。

ステップ 4 VCN の **CIDR ブロック**を入力します。

ステップ 5 **[VCN の作成（Create VCN）]** をクリックします。

ネットワーク セキュリティ グループの作成

ネットワーク セキュリティ グループは、一連の vNIC と、vNIC に適用される一連のセキュリティルールで構成されます。

ステップ 1 **[ネットワーキング（Networking）] > [仮想クラウドネットワーク（Virtual Cloud Networks）] > [仮想クラウドネットワークの詳細（Virtual Cloud Network Details）] > [ネットワーク セキュリティ グループ（Network Security Groups）]**を選択し、**[ネットワーク セキュリティ グループの作成（Create Network Security Group）]** をクリックします。

ステップ 2 [Name] に、ネットワーク セキュリティ グループのわかりやすい名前を入力します（例：*ASAv-Mgmt-Allow-22-443* ）。

ステップ 3 [Next] をクリックします。

ステップ 4 セキュリティルールを追加します。

a) ASAv コンソールへの SSH アクセスに TCP ポート 22 を許可するルールを追加します。

b) ASDM への HTTPS アクセスに TCP ポート 443 を許可するルールを追加します。

ASAv は ASDM を介して管理できます。管理するには、HTTPS 接続用にポート 443 を開く必要があります。

ステップ 5 [作成（Create）] をクリックします。

インターネットゲートウェイの作成

管理サブネットを公的にアクセス可能にするには、インターネットゲートウェイが必要です。

ステップ 1 **[ネットワーキング（Networking）] > [仮想クラウドネットワーク（Virtual Cloud Networks）] > [仮想クラウドネットワークの詳細（Virtual Cloud Network Details）] > [インターネットゲートウェイ（Internet Gateways）]**を選択し、**[インターネットゲートウェイの作成（Create Internet Gateway）]** をクリックします。

ステップ 2 [Name] にインターネットゲートウェイのわかりやすい名前を入力します（例：*ASAv-IG*）。

ステップ 3 **[インターネットゲートウェイの作成（Create Internet Gateway）** をクリックします。

ステップ 4 インターネットゲートウェイへのルートを追加します。

a) **[ネットワーキング（Networking）] > [仮想クラウドネットワーク（Virtual Cloud Networks）] > [仮想クラウドネットワークの詳細（Virtual Cloud Network Details）] > [ルートテーブル（Route Tables）]**を選択します。

b) ルートルールを追加するには、デフォルトのルートテーブルのリンクをクリックします。

c) **[ルートルールの追加（Add Route Rules）]** をクリックします。

d) **[ターゲットタイプ（Target Type）]** ドロップダウンから、**[インターネットゲートウェイ（Internet Gateway）]** を選択します。

e) 宛先の IPv4 CIDR ブロックを入力します（例：0.0.0.0/0）。

f) **[ターゲット インターネット ゲートウェイ（Target Internet Gateway）]** ドロップダウンから、作成したゲートウェイを選択します。
g) **[ルートルールの追加（Add Route Rules）]** をクリックします。

サブネットの作成

各 VCN には、少なくとも 1 つのサブネットがあります。管理 VCN の管理サブネットを作成します。また、内部 VCN の内部サブネット、および外部 VCN の外部サブネットも必要です。

ステップ 1 **[ネットワーキング（Networking）]** > **[仮想クラウドネットワーク（Virtual Cloud Networks）]** > **[仮想クラウドネットワークの詳細（Virtual Cloud Network Details）]** > **[サブネット（Subnets）]**を選択し、**[サブネットの作成（Create Subnet）]** をクリックします。

ステップ 2 [Name] にサブネットのわかりやすい名前を入力します（例：*Management*）。

ステップ 3 **[サブネットタイプ（Subnet Type）]** を選択します（推奨されるデフォルトの **[地域（Regional）]** のままにします）。

ステップ 4 **CIDR ブロック**を入力します（例：10.10.0.0/24）。サブネットの内部（非公開）IP アドレスは、この CIDR ブロックから取得されます。

ステップ 5 **[ルートテーブル（Route Table）]** ドロップダウンから、以前に作成したルートテーブルのいずれかを選択します。

ステップ 6 サブネットの **[サブネットアクセス（Subnet Access）]** を選択します。

管理サブネットの場合、これは**パブリックサブネット**である必要があります。

ステップ 7 **[DHCP オプション（DHCP Option）]** を選択します。

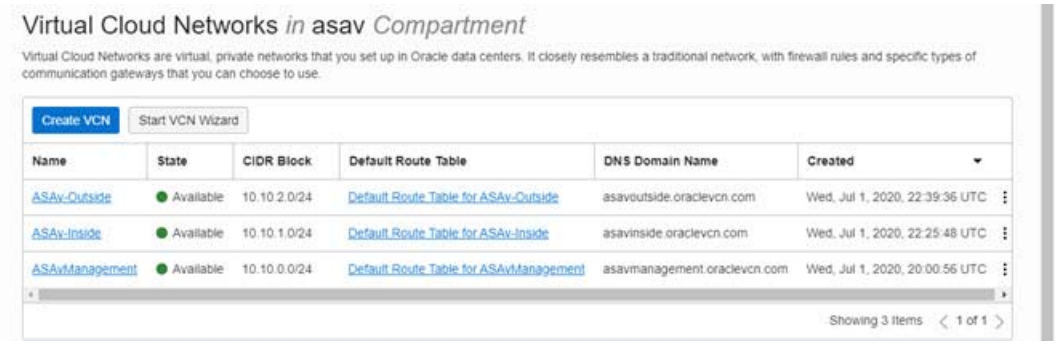
ステップ 8 以前作成した **[セキュリティリスト（Security List）]** を選択します。

ステップ 9 **[サブネットの作成（Create Subnet）]** をクリックします。

次のタスク

VCN（管理、内部、外部）を設定すると、ASAv を起動できます。ASAv VCN 構成の例については、次の図を参照してください。

図 *51 : ASAv* クラウドネットワーク



OCI での ASAv インスタンスの作成

Oracle Cloud Marketplace の Cisco ASA 仮想ファイアウォール（ASAv）製品を使用して、コンピューティング インスタンスを介して OCI に ASAv を導入します。CPU の数、メモリの量、ネットワークリソースなどの特性に基づいて、最適なマシンシェイプを選択します。

ステップ 1 OCI ポータルにログインします。

地域は、画面の右上隅に表示されます。目的の地域内に存在していることを確認してください。

ステップ 2 **[マーケットプレイス（Marketplace）] > [アプリケーション（Applications）]**を選択します。

ステップ 3 マーケットプレイスで「Cisco ASA virtual firewall (ASAv)」を検索して、製品を選択します。

ステップ 4 契約条件を確認し、[Oracle の利用規約とパートナーの契約条件を確認して同意します。（I have reviewed and accept the Oracle Terms of Use and the Partner terms and conditions.）] チェックボックスをオンにします。

ステップ 5 [インスタンスの起動（Launch Instance）] をクリックします。

ステップ 6 [Name] に、インスタンスのわかりやすい名前を入力します（例：*ASAv-9-15*）。

ステップ 7 [シェイプの変更（Change Shape）] をクリックし、ASAv に必要な oCPU の数、RAM の量、およびインターフェイスの数が指定されたシェイプ（VM.Standard2.4 など）を選択します（表 21 : でサポートされるコンピューティングシェイプ ASAv （203 ページ）を参照）。

ステップ 8 **[仮想クラウドネットワーク（Virtual Cloud Network）]** ドロップダウンから、[管理 VCN（Management VCN）] を選択します。

ステップ 9 自動入力されていない場合は、**[サブネット（Subnet）]** ドロップダウンから [管理サブネット（Management subnet）] を選択します。

ステップ 10 **[ネットワーク セキュリティ グループを使用してトラフィックを制御する（Use Network Security Groups to Control Traffic）]** にチェックを入れ、管理 VCN に設定したセキュリティグループを選択します。

ステップ 11 **[パブリック IP アドレスの割り当て（Assign a Public Ip Address）]** オプションボタンをクリックします。

ステップ 12 **[SSH キーの追加（Add SSH keys）]** の下で、**[公開キーの貼り付け（Paste Public Keys）]** オプションボタンをクリックして、SSH キーを貼り付けます。

Linux ベースのインスタンスは、パスワードの代わりに SSH キーペアを使用してリモートユーザーを認証します。キーペアは、秘密キーと公開キーで構成されます。インスタンスを作成するときに、秘密キー

をコンピュータに保持し、公開キーを提供します。ガイドラインについては、『Linux インスタンスでのキーペアの管理（Managing Key Pairs on Linux Instances）』https://docs.cloud.oracle.com/en-us/iaas/Content/Compute/Tasks/managingkeypairs.htmを参照してください。

ステップ 13 **[詳細オプションの表示（Show Advanced Options）]** リンクをクリックして、オプションを展開します。

ステップ 14 （任意）[スクリプトの初期化（Initialization Script）] の下で、[クラウド初期化スクリプトの貼り付け（Paste Cloud-Init Script）] オプションボタンをクリックして、 ASAv の第 0 日用構成を指定します。第 0 日用構成は、ASAv の起動時に適用されます。

次に、**[クラウド初期化スクリプト（Cloud-Init Script）]** フィールドにコピーして貼り付けることができる day0 構成の例を示します。

ASA コマンドの詳細については、『ASA 構成ガイド』および『ASA コマンドリファレンス』を参照してください。

重要 この例からテキストをコピーする場合は、サードパーティのテキストエディタまたは検証エンジンでスクリプトを検証して、形式エラーを防止し、無効な Unicode 文字を削除する必要があります。

```
!ASA Version 9.18.1
interface management0/0
management-only
nameif management
security-level 100
ip address dhcp setroute

no shut
!
same-security-traffic permit inter-interface
same-security-traffic permit intra-interface
!
crypto key generate rsa modulus 2048
ssh 0 0 managementssh timeout 60
ssh version 2
username admin nopassword privilege 15
username admin attributes
service-type admin
http server enable
http 0 0 management
aaa authentication ssh console LOCAL
```

ステップ 15 [作成（Create）] をクリックします。

次のタスク

[作成（Create）] ボタンをクリックした後、状態が [プロビジョニング（Provisioning）] として表示される ASAv インスタンスをモニターします。

重要 ステータスをモニターすることが重要です。ASAv インスタンスの状態が [プロビジョニング（Provisioning）] から [実行中（Running）] に移行したら、ASAv ブートが完了する前に必要に応じて VNIC を接続する必要があります。

インターフェイスの接続

ASAv は、1 つの VNIC が接続された状態で実行状態になります（**[コンピューティング（Compute）] > [インスタンス（Instances）] > [インスタンスの詳細（Instance Details）] > [接続された VNIC （Attached VNICs）]**を参照）。これはプライマリ VNIC と呼ばれ、管理 VCN にマッピングされます。ASAv が最初の起動を完了する前に、vNIC が ASAv で正しく検出されるように、以前作成した他の VCN サブネット（内部、外部）の vNIC を接続する必要があります。

ステップ 1 新しく起動した ASAv インスタンスを選択します。

ステップ 2 **[接続された VNIC（Attached VNICs）] > [VNIC の作成（Create VNIC）]**の順に選択します。

ステップ 3 [名前（Name）] に、VNIC のわかりやすい名前を入力します（例：*Inside*）。

ステップ 4 [仮想クラウドネットワーク（Virtual Cloud Network）] ドロップダウンから VCN を選択します。

ステップ 5 [サブネット（Subnet）] ドロップダウンからサブネットを選択します。

ステップ 6 **[ネットワーク セキュリティ グループを使用してトラフィックを制御する（Use Network Security Groups to Control Traffic）]** をオンにして、選択した VCN 用に設定したセキュリティグループを選択します。

ステップ 7 [送信元と宛先のチェックをスキップ（Skip Source Destination Check）] をオンにします。

ステップ 8 （オプション）[プライベート IP アドレス（Private IP Address）] を指定します。これは、VNIC に対して特定の IP を選択する場合にのみ必要です。

IP を指定しない場合、OCI はサブネットに割り当てられた CIDR ブロックから IP アドレスを割り当てます。

ステップ 9 [変更の保存 (Save Changes)] をクリックし、VNIC を作成します。

ステップ 10 展開で必要となる各 VNIC について、この手順を繰り返します。

接続された VNIC のルートルールの追加

内部および外部のルートテーブルにルートテーブルルールを追加します。

ステップ 1 **[Networking] > [Virtual Cloud Networks]** を選択し、VCN に関連付けられているデフォルトルートテーブル（内部または外部）をクリックします。

ステップ 2 **[ルートルールの追加（Add Route Rules）]** をクリックします。

ステップ 3 [ターゲットタイプ（Target Type）] ドロップダウンから、[プライベート IP（Private IP）] を選択します。

ステップ 4 [宛先タイプ（Destination Type）] ドロップダウンから、[CIDR ブロック（CIDR Block）] を選択します。

ステップ 5 [宛先のIPv4 CIDRブロック（Destination IPv4 CIDR Block）] に宛先の IPv4 CIDR ブロックを入力します（例：0.0.0.0/0）。

ステップ 6 [ターゲット選択（Target Selection）] フィールドに VNIC のプライベート IP アドレスを入力します。

VNIC に IP アドレスを明示的に割り当てていない場合は、VNIC の詳細（**[コンピューティング（Compute）]**> **[インスタンス（Instances）]** > **[インスタンスの詳細（Instance Details）]** > **[接続された VNIC（Attached VNICs）]**）で自動割り当てされた IP アドレスを確認できます。

ステップ 7 **[ルートルールの追加（Add Route Rules）]** をクリックします。

ステップ 8 展開で必要となる各 VNIC について、この手順を繰り返します。

（注） ASA Virtual の（静的および DHCP）設定に必要な個別のルーティングルール。

OCI 上の ASAv インスタンスへのアクセス

セキュアシェル（SSH）接続を使用して、実行中のインスタンスに接続できます。

- ほとんどの UNIX スタイルのシステムには、デフォルトで SSH クライアントが含まれています。
- Windows 10 および Windows Server 2019 システムには、OpenSSH クライアントが含まれている必要があります。Oracle Cloud Infrastructure によって生成された SSH キーを使用してインスタンスを作成した場合に必要になります。
- その他の Windows バージョンの場合は、http://www.putty.org から無償の SSH クライアントである PuTTY をダウンロードできます。

前提条件

インスタンスに接続するには、次の情報が必要です。

- インスタンスのパブリック IP アドレス。アドレスは、コンソールの [インスタンスの詳細（Instance Details）] ページから取得できます。ナビゲーションメニューを開きます。**[コアインフラストラクチャ（Core Infrastructure）]** の下で、**[コンピューティング（Compute）]** に移動し、**[インスタンス（Instances）]** をクリックします。次に、インスタンスを選択します。あるいは、コアサービス API の ListVnicAttachments および GetVnic 操作を使用できます。
- インスタンスのユーザー名とパスワード。
- インスタンスを起動したときに使用した SSH キーペアの秘密キー部分へのフルパス。キーペアの詳細については、「Managing Key Pairs on Linux Instances」を参照してください。

（注） 第 0 日用構成で指定したログイン情報を使用するか、インスタンスの起動時に作成した SSH キーペアを使用して、ASAv インスタンスにログインできます。

SSH を使用した ASAv インスタンスへの接続

UNIX スタイルのシステムから ASAv 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。

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>
```

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。

<username> は、ASAv 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。

OpenSSH を使用した ASAv インスタンスへの接続

Windows システムから ASAv インスタンスに接続するには、OpenSSH を使用してインスタンスにログインします。

ステップ 1 このキーペアを初めて使用する場合は、自分だけがファイルを読み取れるようにファイルの権限を設定する必要があります。

次の手順を実行します。

a) Windows Explorer で、秘密キーファイルに移動し、ファイルを右クリックして **[プロパティ（Properties）]** をクリックします。
b) **[セキュリティ（Security）]** タブで、**[詳細設定（Advanced）]** をクリックします。
c) **[オーナー（Owner）]** が自分のユーザーアカウントであることを確認します。

d) **[継承の無効化（Disable Inheritance）]** をクリックし、**[継承された権限をこのオブジェクトの明示的な権限に変換する（Convert inherited permissions into explicit permissions on this object）]** を選択します。
e) 自分のユーザーアカウントではない各権限エントリを選択し、**[削除（Remove）]** をクリックします。
f) 自分のユーザーアカウントのアクセス権限が **[フルコントロール（Full Control）]** であることを確認します。
g) 変更を保存します。

ステップ 2 インスタンスに接続するには、Windows PowerShell を開き、次のコマンドを実行します。

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

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。

<username> は、ASAv 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。

PuTTY を使用した ASAv インスタンスへの接続

PuTTY を使用して Windows システムから ASAv インスタンスに接続するには、次の手順を実行します。

ステップ 1 PuTTY を開きます。

ステップ 2 **[カテゴリ（Category）]** ペインで、**[セッション（Session）]** を選択し、次の内容を入力します。

- **ホスト名または IP アドレス**：

  <username>@<public-ip-address>

  ここで、

  <username> は、ASAv 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パブリック IP アドレスです。

- **ポート**： 22
- **接続タイプ**： SSH

ステップ 3 **[カテゴリ（Category）]** ペインで、**[Window]** を展開し、**[変換（Translation）]** を選択します。

ステップ 4 **[リモート文字セット（Remote character set）]** ドロップダウンリストで、**[UTF-8]** を選択します。

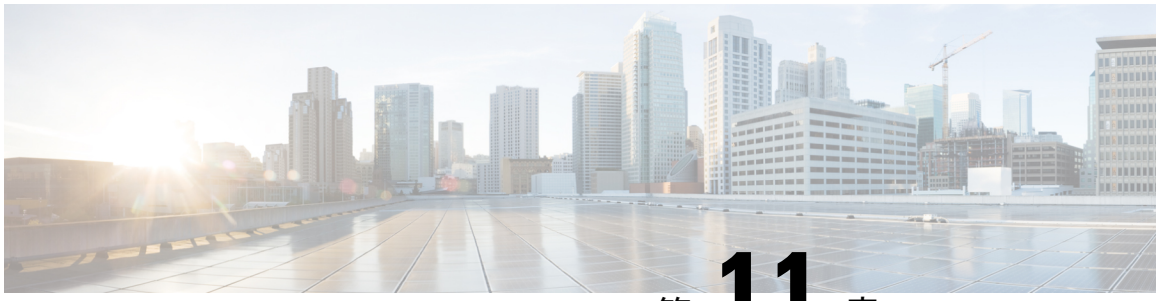
Linux ベースのインスタンスでデフォルトのロケール設定は UTF-8 です。これにより、PuTTY は同じロケールを使用するように設定されます。

ステップ 5 **[カテゴリ（Category）]** ペインで、**[接続（Connection）]**、**[SSH]** の順に展開し、**[認証（Auth）]** をクリックします。

ステップ 6 **[参照（Browse）]** をクリックして、秘密キーを選択します。

ステップ 7 **[開く（Open）]** をクリックして、セッションを開始します。

インスタンスに初めて接続する場合は、「サーバーのホストキーがレジストリにキャッシュされていない（the server's host key is not cached in the registry）」というメッセージが表示されることがあります。**[はい（Yes）]** をクリックして、接続を続行します。



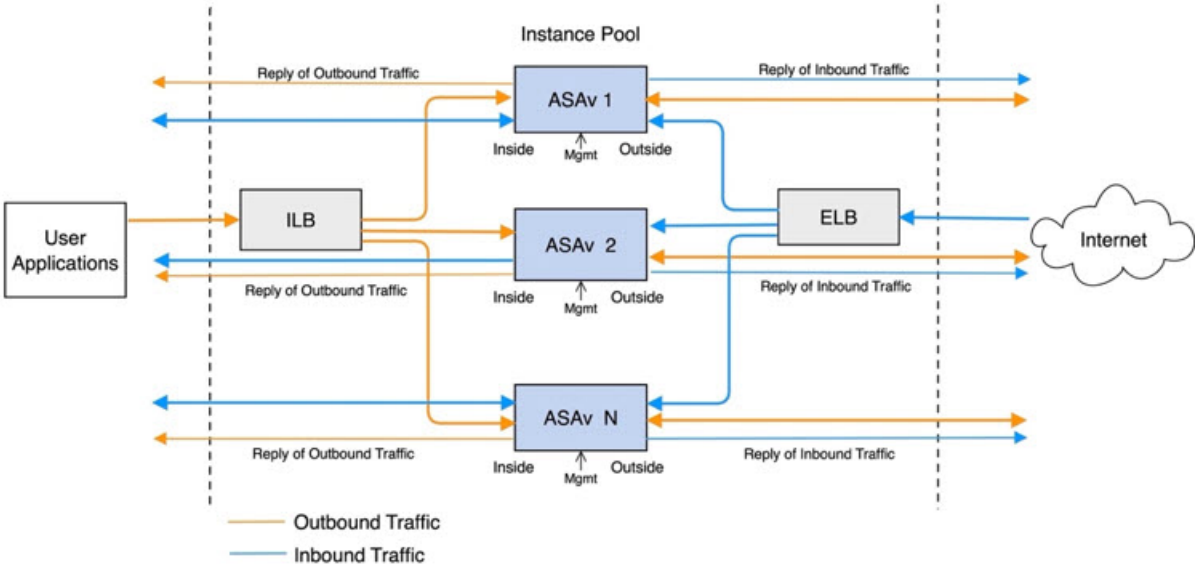
第 11 章

OCI への ASAv Auto Scale ソリューションの導入

Auto Scale の導入例

この ASAv の導入例：OCI Auto Scale ソリューションは、導入例の図に示されています。インターネット向けのロードバランサには、リスナーとターゲットグループの組み合わせを使用してポートが有効になっているパブリック IP アドレスがあります。

図 *52* : 導入例の図



ポートベースの分岐は、ネットワークトラフィックに実装できます。この分岐は、NAT ルールによって実現できます。分岐の設定例については、以下のセクションで説明します。

前提条件

権限およびポリシー

ソリューションを導入するために必要な OCI の権限とポリシーは次のとおりです。

1. **ユーザーおよびグループ**

（注） ユーザーとグループを作成するには、OCIユーザーまたはテナンシー管理者である必要があります。

Oracle Cloud Infrastructure のユーザーアカウントと、そのユーザーアカウントが属するグループを作成します。ユーザーアカウントを持つ関連グループが存在する場合は、作成する必要はありません。ユーザーとグループの作成手順については、「グループとユーザーの作成」を参照してください。

2. **グループ ポリシー**

ポリシーを作成したら、それをグループにマッピングする必要があります。ポリシーを作成するには、**[OCI]** > **[アイデンティティとセキュリティ（Identity & Security）]** > **[ポリシー（Policies）]** > **[ポリシーの作成（Create Policy）]**に移動します。次のポリシーを作成して、目的のグループに追加します。

- グループ <*Group_Name*> がコンパートメント <*Compartment_Name*> でメトリックを使用することを許可します。
- グループ <*Group_Name*> がコンパートメント <*Compartment_Name*> でアラームを管理することを許可します。
- グループ <*Group_Name*> がコンパートメント <*Compartment_Name*> で ONS トピックを管理することを許可します。
- グループ <*Group_Name*> がコンパートメント <*Compartment_Name*> でメトリックを検査することを許可します。
- グループ <*Group_Name*> がコンパートメント <*Compartment_Name*> でメトリックを読み取ることを許可します。
- グループ <*Group_Name*> がコンパートメント <*Compartment_Name*> でタグの名前空間を使用することを許可します。
- グループ <*Group_Name*> がコンパートメント <*Compartment_Name*>でロググループを読み取ることを許可します。
- グループ <*Group_Name*> がインスタンスプール コンパートメント <*Compartment_Name*> を使用することを許可します。
- グループ <*Group_Name*> がテナントでクラウドシェルを使用することを許可します。
- グループ <*Group_Name*> がテナントのオブジェクトストレージ名前空間を読み取ることを許可します
- グループ <*Group_Name*> がテナント内のリポジトリを管理することを許可します。

（注） テナントレベルでポリシーを作成することもできます。ユーザーの責任と判断のもとで、すべての権限を自由に指定できます。

3. Oracle 関数の権限

Oracle 関数が別の Oracle Cloud Infrastructure リソースにアクセスできるようにするには、関数をダイナミックグループに含めてから、そのリソースへのダイナミック グループ アクセスを許可するポリシーを作成します。

4. ダイナミックグループの作成

ダイナミックグループを作成するには、**[OCI]**>**[アイデンティティとセキュリティ（Identity & Security）]** > **ダイナミックグループ（Dynamic Group）]** > **[ダイナミックグループの作成（Create Dynamic Group）]**に移動します。

ダイナミックグループの作成時に次のルールを指定します。

```
ALL {resource.type = 'fnfunc', resource.compartment.id = '<Your_Compartment_OCID>'}
```

ダイナミックグループの詳細については、次を参照してください。

- https://docs.oracle.com/en-us/iaas/Content/Functions/Tasks/functionsaccessingociresources.htm
- https://docs.oracle.com/en-us/iaas/Content/Identity/Tasks/managingdynamicgroups.htm

5. ダイナミックグループのポリシーの作成

   ポリシーを追加するには、**[OCI]**> **[アイデンティティとセキュリティ（Identity & Security）]**>**[ポリシー（Policies）]**>**[ポリシーの作成（Create Policy）]**に移動します。次のポリシーをグループに追加します。

   ```
   Allow dynamic-group <Dynamic_Group_Name> to manage all-resources in compartment
   <Compartment_OCID>
   ```

GitHub からのファイルのダウンロード

ASAv：OCI Auto Scale ソリューションは、GitHub リポジトリ形式で配布されます。リポジトリからファイルをプルまたはダウンロードできます。

Python3 環境

make.py ファイルは、複製されたリポジトリ内にあります。このプログラムは、Oracle 関数とテンプレートファイルを ZIP ファイルに圧縮します。それらをターゲットフォルダーにコピーします。これらのタスクを実行するには、Python 3 環境が設定されている必要があります。

（注） この Python スクリプトは Linux 環境でのみ使用できます。

インフラストラクチャ設定

次を設定する必要があります。

1. **VCN**

   ASAv アプリケーションの要件に応じて VCN を作成します。インターネットへのルートが割り当てられたサブネットが 1 つ以上あるインターネットゲートウェイを備えた VPC を作成します。

   VCN の作成については、「https://docs.oracle.com/en-us/iaas/Content/GSG/Tasks/creatingnetwork.htm」を参照してください。

2. **アプリケーションサブネット**

   ASAv アプリケーションの要件に応じてサブネットを作成します。このユースケースに従ってソリューションを導入するには、ASAv インスタンスの運用に 3 つのサブネットが必要です。

   サブネットの作成については、https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/managingVCNs_topic-Overview_of_VCNs_and_Subnets.htm# を参照してください。

3. **外部サブネット**

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」のルートが必要です。このサブネットには、Cisco ASAv の外部インターフェイスとインターネット向けロードバランサが含まれています。アウトバウンドトラフィック用に NAT ゲートウェイが追加されていることを確認します。

詳細については、次のマニュアルを参照してください。

- https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/managingIGs.htm
- https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/NATgateway.htm#To_create_a_NAT_gateway

4. **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。

（注） ASAv 正常性プローブの場合、ポート 80 を介してメタデータサーバー（169.254.169.254）に到達できます。

5. **管理サブネット**

   管理サブネットは、ASAv への SSH 接続をサポートするようにパブリックにする必要があります。

6. **セキュリティ グループ：ASAv インスタンスのネットワーク セキュリティ グループ**

   次の要件に対応した ASAv インスタンスのセキュリティグループを設定します。

   - Oracle 関数（同じ VCN 内）は、ASAv の管理アドレスへの SSH 接続を実行します。
   - 管理ホストでは、 SSH を介した ASAv インスタンスへのアクセスが必要になる場合があります。
   - ASAv はライセンスのために CSSM/Satellite サーバーとの通信を開始します。

7. **オブジェクトストレージの名前空間**

   このオブジェクトストレージの名前空間は、configuration.txt ファイルを持つ静的 Web サイトをホストするために使用されます。configuration.txt ファイルの事前認証済みリクエストを作成する必要があります。この事前認証された URL は、テンプレートの展開時に使用されます。

（注） アップロードされた次の設定に、HTTP URL を介して ASAv インスタンスからアクセスできることを確認します。

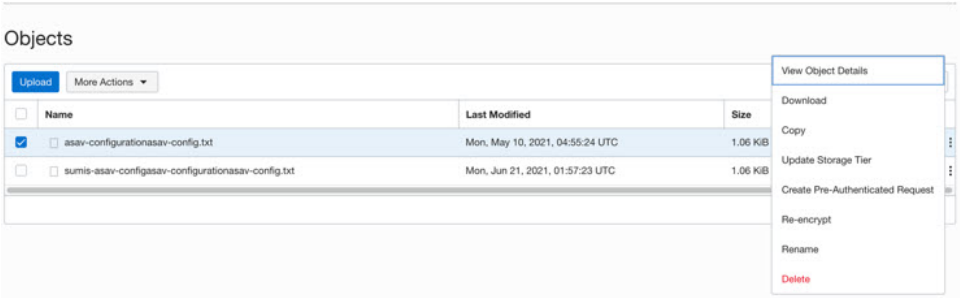
ASAv を起動すると、`$ copy /noconfirm <configuration.txt file's pre-authenticated request URL > disk0:Connfiguration.txt` コマンドが実行されます。

このコマンドにより、ASAv の起動を configuration.txt ファイルで設定できるようになります。

8. **configuration.txt ファイルのアップロード**

   ASAv 構成ファイルの事前認証済みリクエスト URL を作成するには、次の手順を実行します。

   1. **[バケット（Buckets）] > [バケットの作成（Create Bucket）]**の順にクリックします。
   2. [アップロード（Upload）] をクリックします。
   3. 構成ファイルがアップロードされたら、下の図に示すように、[事前認証済みリクエストの作成（Create Pre-Authenticated Request）] を選択します。



（注） これで、オラクル関数から構成ファイルにアクセスできるようになります。

ネットワーク構成

1. **インバウンドトラフィック**

   ステップ 2で説明されているように、configuration.txt 内の <*Application VM IP*> アドレスが正しいことを確認します。

2. **アウトバウンドトラフィック**

   - ステップ 2で説明されているように、configuration.txt 内の <*External Server IP*> アドレスが正しいことを確認します。
   - 外部 VCN に 1 つの NAT ゲートウェイがあることを確認します。

- 次の図の例に示すように、 NAT ゲートウェイを経由する外部 VCN のルートテーブル内の同じ <*External Server IP*> アドレスを追加してください。

パスワードの暗号化

（注） この手順の詳細については、「Vault とシークレットの作成」を参照してください。

ASAv のパスワードは、自動スケーリング中に使用されるすべての ASAv インスタンスを設定するために使用されます。また、ASAvインスタンスのCPU使用率データを取得するために使用されます。

したがって、パスワードを時々保存して処理する必要があります。頻繁な変更と脆弱性のため、プレーンテキスト形式での「パスワードの編集や保存はできません。パスワードには、暗号化された形式のみを使用する必要があります。

暗号化された形式のパスワードを取得するには、次の手順を実行します。

ステップ 1 Vault を作成します。

OCI Vault は、マスター暗号化キーを安全に作成および保存するサービスと、それらを使用する際に暗号化および復号化する方法を提供します。したがって、Vaultは、自動スケールソリューションの残りの部分と同じコンパートメントに作成する必要があります（まだ作成していない場合）。

[OCI] > **[アイデンティティとセキュリティ（Identity & Security）]** > **[Vault]** > **[新規Vaultの選択または作成（Choose or Create New Vault）]**に移動します。

。

ステップ 2 マスター暗号化キーを作成します。

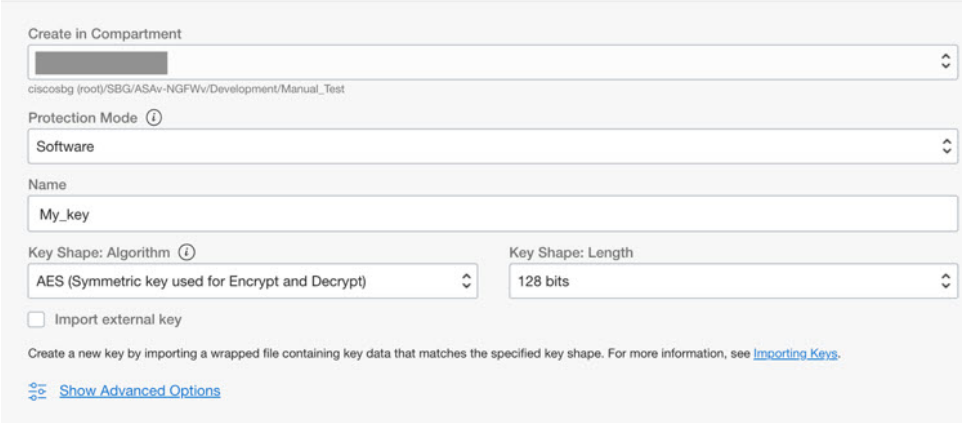
プレーンテキストのパスワードを暗号化するには、マスター暗号化キーが 1 つ必要です。

[OCI] > **[アイデンティティとセキュリティ（Identity & Security）]** > **[Vault]** > **[キーの選択または作成（Choose or Create Key）]**に移動します。

任意のビット長で、指定されたアルゴリズムのいずれかから任意のキーを選択します。

1. AES：128、192、256
2. RSA：2048、3072、4096
3. ECDSA：256、384、521

図 *53* : キーの作成



ステップ 3 暗号化されたパスワードを作成します。

1. **[OCI] > [CloudShell（OCI Cloud Terminal）を開く（Open CloudShell (OCI Cloud Terminal)**に移動します。
2. <*Password*> をお使いのパスワードに置き換えて、次のコマンドを実行します。

```
echo -n '<Password>' | base64
```

3. 選択した Vault から、暗号化エンドポイントとマスター暗号化キーの OCID をコピーします。次のように値を置き換えてから、暗号化コマンドを実行します。
   - KEY_OCID：キーの OCID
   - Cryptographic_Endpoint_URL：Vault の暗号化エンドポイント URL
   - Password：パスワード

   暗号化コマンド

```
oci kms crypto encrypt --key-id Key_OCID --endpoint
Cryptographic_Endpoint_URL --plaintext <base64-value-of-password>
```

4. 上記のコマンドの出力から暗号文をコピーし、必要に応じて使用します。

ASA 構成ファイルの準備

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かを確認します。

ステップ 1 展開する前に、次の入力パラメータを収集します。

パラメータ	データタイプ	説明
tenancy_ocid	文字列	アカウントが属するテナントの OCID。テナントの OCID を見つける方法については、こちらを参照してください。 テナントの OCID は `ocid1.tenancy.oc1..<unique_ID>` のようになります。
compartment_id	文字列	リソースを作成するコンパートメントの OCID。 例： `ocid1.compartment.oc1..<unique_ID>`
compartment_name	文字列	コンパートメント名
region	文字列	リソースを作成するリージョンの一意の識別子。 例： `us-phoenix-1, us-ashburn-1`
lb_size	文字列	事前にプロビジョニングする外部および内部ロードバランサの合計帯域幅（入力および出力）を決定するテンプレート。 サポートされる値：100 Mbps、10 Mbps、10 Mbps-Micro、400 Mbps、8000 Mbps 例：100 Mbps
availability_domain	カンマ区切り値	例：Tpeb:PHX-AD-1 （注） クラウドシェルで **oci iam availability-domain list** コマンドを実行して、可用性ドメイン名を取得します。
min_and_max_instance_count	カンマ区切り値	インスタンスプールに保持するインスタンスの最小数と最大数。 例：1,5

パラメータ	データタイプ	説明
autoscale_group_prefix	文字列	テンプレートを使用して作成したリソースの名前に付けるプレフィックス。たとえば、リソースプレフィックスとして「autoscale」を指定すると、すべてのリソースはautoscale_resource1、autoscale_resource2 のように名前が付けられます。
asav_config_file_url	URL	ASAv の構成用にオブジェクトストレージにアップロードする構成ファイルの URL。 （注） 構成ファイルの事前認証済みリクエスト URL を指定する必要があります 例： https://objectstorage.<region-name>.oraclecloud.com/<object-storage-name>/oci-asav-configuration.txt
mgmt_subnet_ocid	文字列	使用する管理サブネットのOCID。
inside_subnet_ocid	文字列	使用する内部サブネットのOCID。
outside_subnet_ocid	文字列	使用する外部サブネットのOCID。
mgmt_nsg_ocid	文字列	使用する管理サブネットのネットワーク セキュリティ グループのOCID。
inside_nsg_ocid	文字列	使用する内部サブネットのネットワーク セキュリティ グループのOCID。
outside_nsg_ocid	文字列	使用する外部サブネットのネットワーク セキュリティ グループのOCID。
elb_listener_port	カンマ区切り値	外部ロードバランサリスナーの通信ポートのリスト。 例： 80

パラメータ	データタイプ	説明
ilb_listener_port	カンマ区切り値	内部ロードバランサリスナーの通信ポートのリスト。 例： 80
health_check_port	文字列	ヘルスチェックを実行するロードバランサのバックエンドサーバーポート。 例：8080
instance_shape	文字列	作成するインスタンスのシェープ。シェイプにより、インスタンスに割り当てられるCPUの数、メモリの量、およびその他のリソースが決定されます。 サポートされているシェープ：「VM.Standard2.4」および「VM.Standard2.8」
lb_bs_policy	文字列	内部および外部ロードバランサのバックエンドセットに使用するロードバランサポリシー。ロードバランサポリシーの仕組みについて詳しくは、こちらを参照してください。 サポートされている値：「ROUND_ROBIN」、「LEAST_CONNECTIONS」、「IP_HASH」
image_name	文字列	インスタンスの構成に使用するマーケットプレイスのイメージ名。 デフォルト値：「Cisco ASA 仮想ファイアウォール（ASAv）」 （注） カスタムイメージを展開する場合は、custom_image_ocid パラメータを設定する必要があります。

パラメータ	データタイプ	説明
image_version	文字列	使用する OCI Marketplace で利用可能な ASAv イメージのバージョン。現在、9.15.1.15 および 9.16.1 バージョンが利用可能です。 デフォルト値：「Cisco ASA 仮想ファイアウォール（ASAv）」
scaling_thresholds	カンマ区切り値	スケールインとスケールアウトで使用する CPU 使用率のしきい値。スケールインとスケールアウトのしきい値をカンマで区切って入力します。 例：15,50 15 はスケールインのしきい値、50 はスケールアウトのしきい値です。
custom_image_ocid	文字列	マーケットプレイスイメージを使用しない場合に、インスタンス構成に使用するカスタムイメージの OCID。 （注） custom_image_ocid はオプションパラメータです
asav_password	文字列	ASAv を構成するために SSH 接続する際の、ASAv の暗号化形式のパスワード。パスワードを暗号化する方法については、コンフィギュレーションガイドを使用するか、こちらを参照してください。
cryptographic_endpoint	文字列	暗号化エンドポイントは、パスワードの復号化に使用される URL です。Vault で検索できます。
master_encryption_key_id	文字列	パスワードの暗号化に使用されたキーの OCID。Vault で検索できます。

パラメータ	データタイプ	説明
プロファイル名（Profile Name）		OCI のユーザーのプロファイル名です。ユーザーのプロファイルセクションの下にあります。 例：oracleidentitycloudservice/<user>@<mail>.com
オブジェクトストレージの名前空間（Object Storage Namespace）		テナントの作成時に作成される一意の識別子です。この値は**[OCI]**>**[管理（Administration）]**>**[（テナントの詳細Tenancy Details）]**で確認できます。
認証トークン（Authorization Token）		OCI コンテナレジストリに Oracle 関数をプッシュすることを許可する Docker へのログイン時のパスワードとして使用されます。トークンを取得するには、**[OCI]** > **[アイデンティティ（Identity）]** > **[ユーザー（Users）]**>**[ユーザの詳細（User Details）]** > **[認証トークン（Auth Tokens）]** > **[トークンの生成（Generate Token）]**に移動します。

ステップ 2 ロードバランサの正常性プローブとアクセスポリシーのオブジェクト、ライセンス、NAT ルールを設定します。

```
! Default route via outside
route outside 0.0.0.0 0.0.0.0 <Outside Subnet gateway> 2

! Health Check Configuration
object network metadata-server
host 169.254.169.254
object service health-check-port
service tcp destination eq <health-check-port>
object service http-port
service tcp destination eq <traffic port>
route inside 169.254.169.254 255.255.255.255 <Inside Subnet GW> 1

! Health check NAT
nat (outside,inside) source static any interface destination static interface metadata-server service
 health-check-port http-port
nat (inside,outside) source static any interface destination static interface metadata-server service
 health-check-port http-port

! Outbound NAT
object network inside-subnet
subnet <Inside Subnet> <Inside Subnet Gateway>
object network external-server
host <External Server IP>
nat (inside,outside) source static inside-subnet interface destination static interface external-server
```

```
! Inbound NAT
object network outside-subnet
subnet <Outside Subnet> <Outside Subnet GW>
object network http-server-80
host <Application VM IP>
nat (outside,inside) source static outside-subnet interface destination static interface http-server-80


!
dns domain-lookup outside
DNS server-group DefaultDNS

! License Configuration
call-home
profile license
destination transport-method http
destination address http <URL>
debug menu license 25 production
license smart
feature tier standard
throughput level <Entitlement>
licence smart register idtoken <License token> force
!
```

これらの正常性プローブ接続およびデータプレーンがアクセスポリシーで許可されている必要があります。

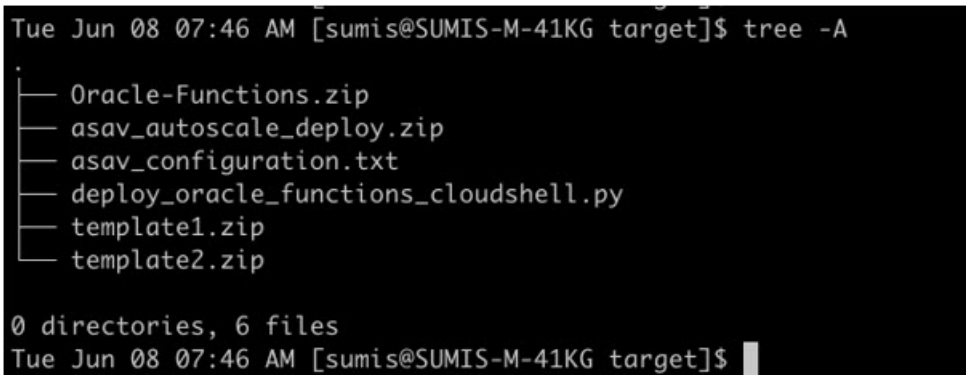
ステップ 3 設定の詳細を使用して *configuration.txt* ファイルを更新します。

ステップ 4 ユーザーが作成したオブジェクトストレージスペースに *configuration.txt* ファイルをアップロードし、アップロードしたファイルの事前認証リクエストを作成します。

（注） スタックの展開で、configuration.txt の事前認証済みリクエスト URL が使用されていることを確認します。

ステップ 5 ZIP ファイルを作成します。

make.py ファイルは、複製されたリポジトリ内にあります。`python3 make.py build` コマンドを実行して、zip ファイルを作成します。対象フォルダには以下のファイルがあります。



（注） クラウドシェルを使用して Auto Scale ソリューションを展開する場合は、`python3 make.py build` を実行する前に *easy_deploy/deployment_parameters.json* ファイルを更新します。更新については、「ステップ 1」および「Oracle 関数の展開」を参照してください。

OCI への Auto Scale の展開

展開の前提条件となる手順を完了したら、OCI スタックの作成を開始します。手動展開を実行するか、（クラウドシェルを使用した Auto Scale の導入）を実行できます。該当する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。

手動展開

エンドツーエンドの Auto Scale ソリューションの展開は、次の 3 つの手順で構成されます。Terraform Template-1 スタックの展開 、Oracle 関数の展開、次いで Terraform Template-2 の展開

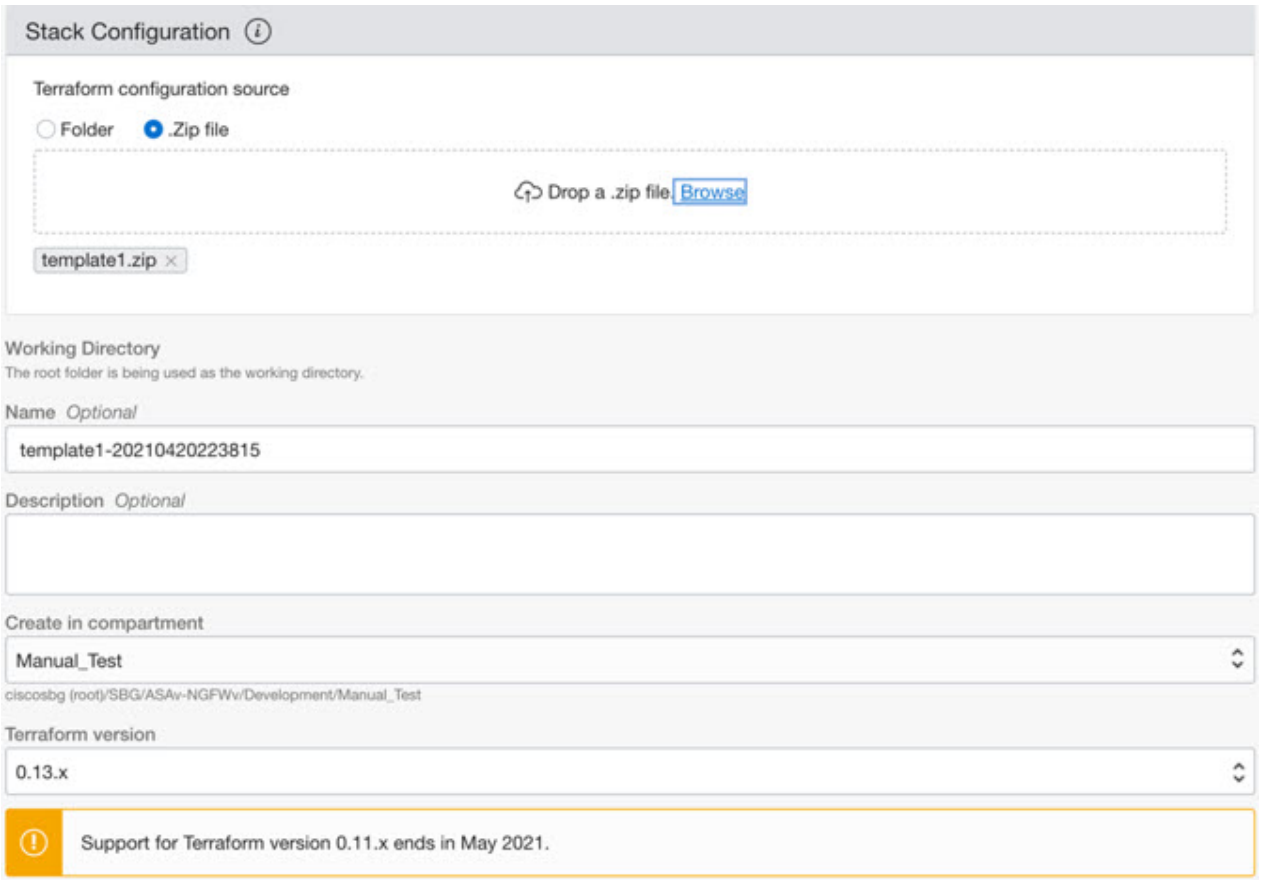
Terraform Template-1 スタックの展開

ステップ 1 OCI ポータルにログインします。

地域は、画面の右上隅に表示されます。目的の地域内に存在していることを確認してください。

ステップ 2 **[デベロッパーサービス（Developer Service）] > [リソースマネージャ（Resource Manager）] > [スタック（Stack）] > [スタックの作成（Create Stack）]**の順に選択します。

[マイ設定（My Configuration）] を選択し、次の図に示すように、対象フォルダ内にある *Terraform template1.zip* ファイルを Terraform の設定ソースとして選択します。



ステップ 3 [トランスフォームバージョン（Transform version）] ドロップダウンリストで、0.13.x または 0.14.x を選択します。

ステップ 4 次の手順では、ステップ 1で収集した詳細情報をすべて入力します。

（注） 有効な入力パラメータを入力しnoてください。そうしないと、以降の手順でスタックの展開に失敗する可能性があります。

ステップ 5 次の手順で**[Terraformアクション（Terraform Actions）]** > **[適用（Apply）]**を選択します。

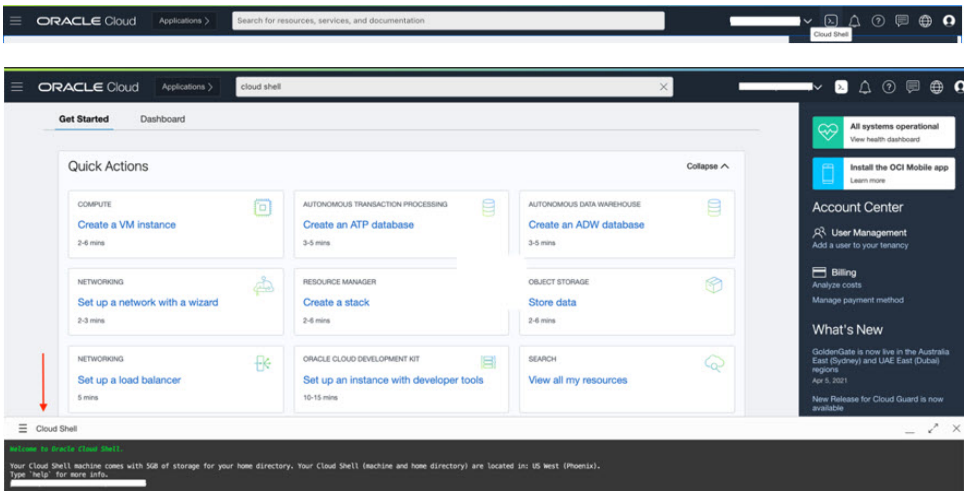
正常に展開されたら、Oracle 関数の展開に進みます。

Oracle 関数の展開

（注） この手順は、*Terraform Template-1* の導入が成功した後にのみ実行する必要があります。

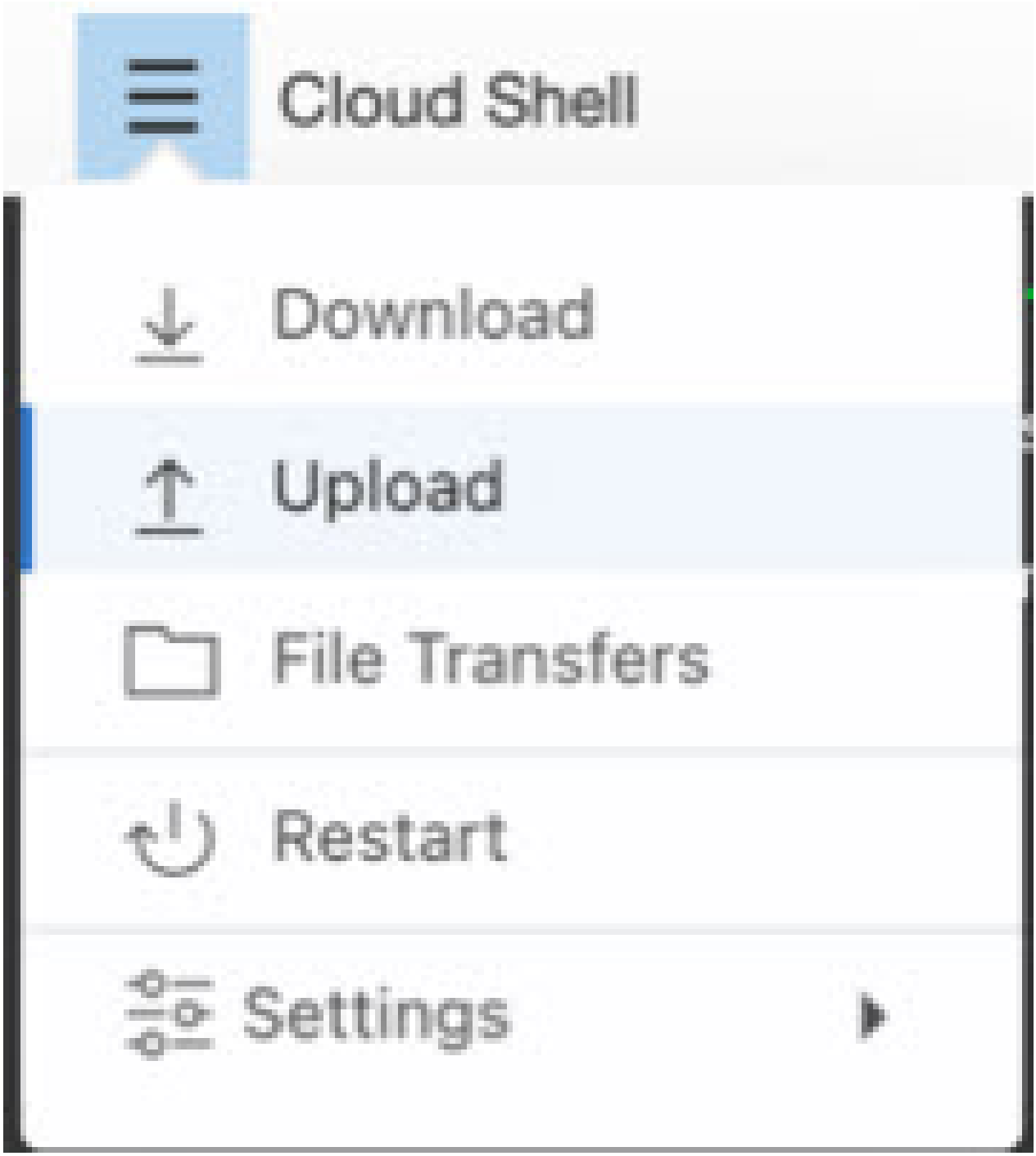
OCI では、Oracle 関数は Docker イメージとしてアップロードされ、OCI コンテナレジストリに保存されます。Oracle 関数は、導入時に OCI アプリケーション（Terraform Template-1 で作成）の 1 つにプッシュする必要があります。

ステップ 1 OCI のクラウドシェルを開きます。



ステップ 2 *deploy_oracle_functions_cloudshell.py* と *Oracle-Functions.zip* をアップロードします。

クラウドシェルのハンバーガーメニューから [アップロード（Upload）] を選択します。



ステップ 3 **ls** コマンドを使用してファイルを確認します。

```
$ ls
Deploy_Oracle_Functions.py  Oracle-Functions.zip
```

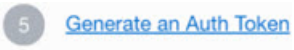
ステップ 4 `python3 Deploy_Oracle_Functions.py -h` を実行します。以下の図に示すように、`deploy_oracle_functions_cloudshell.py` スクリプトには、いくつかの入力パラメータが必要です。詳細は help 引数を使用して確認できます。

```
                                  $ python3 Deploy_Oracle_Functions.py -h
usage: Deploy_Oracle_Functions.py [-h] -a  -r  -p  -c  -o  -t

*** Script to deploy Oracle Function for OCI ASAv Autoscale Solution ***

Instruction to find values of required arguments:
Application Name: Name of Application created by first Terraform Template
Region Identifier: OCI -> Administration -> Region Management
Profile Name: OCI -> Profile
Compartment OCID: OCI -> Identity -> Compartment -> Compartment Details
Object Storage Namespace: OCI -> Administration -> Tenancy Details
Authorization Token: OCI -> Identity -> Users -> User Details -> Auth Tokens -> Generate Token

optional arguments:
  -h, --help  show this help message and exit
  -a          Name of Application in OCI to which functions will be deployed
  -r          Region Identifier
  -p          Profile Name of User
  -c          Compartment OCID
  -o          Object Storage Namespace
  -t          Authorization Token for Docker Login (*Please Put in Quotes)
```

スクリプトを実行するには、次の引数を渡します。

表 *22 :* 引数と詳細

引数	特記事項
アプリケーション名（**Application Name**）	Terraform Template-1 の導入で作成した OCI アプリケーションの名前です。この値は、Template-1 で付与された「**autoscale_group_prefix**」とサフィックス「**_application**」を組み合わせたものです。
リージョン識別子（**Region Identifier**）	リージョン識別子は、さまざまな地域の OCI で固定された地域コードワードです。 例：フェニックスの場合は「us-phoenix-1」、メルボルンの場合は「ap-melbourne-1」。 すべてのリージョンとそのリージョン識別子のリストを取得するには、**[OCI] > [管理（Administration）] > [リージョン管理（Region Management）]**に移動します。
プロファイル名（**Profile Name**）	OCI のシンプルなユーザープロファイル名です。 例：*oracleidentitycloudservice/<user> @<mail> .com* 名前は、ユーザーのプロファイルセクションの下にあります。

引数	特記事項
コンパートメント OCID（Compartment OCID）	これは、コンパートメントの OCID（Oracle Cloud 識別子）です。ユーザーが OCI アプリケーションを格納しているコンパートメントの OCID。 **[OCI] > [アイデンティティ（Identity）] > [コンパートメント（Compartment）] > [コンパートメントの詳細（Compartment Details）]**に移動します。
オブジェクトストレージの名前空間（Object Storage Namespace）	テナントの作成時に作成される一意の識別子です。 **[OCI] > [管理（Administration）] > [テナントの詳細（Tenancy Details）]**に移動します。
認証トークン（Authorization Token）	これは、OCI コンテナレジストリに Oracle 関数をプッシュすることを許可する Docker ログイン用のパスワードとして使用されます。導入スクリプトでトークンを引用符で囲んで指定します。 **[OCI] > [アイデンティティ（Identity）] > [ユーザー（Users）] > [ユーザの詳細（User Details）] > [認証トークン（Auth Tokens）] > [トークンの生成（Generate Token）]**に移動します。 何らかの理由でユーザーの詳細が表示されない場合は、**[開発者サービス（Developer services）] > [機能（Functions）]**をクリックします。Terraform Template-1 で作成したアプリケーションに移動します。[利用を開始する（Getting Started）] をクリックし、[クラウドシェルの設定（Cloud Shell Setup）] を選択すると、手順を進めていく中で、以下に示すように認証トークンを生成するためのリンクが表示されます。 

ステップ 5 有効な入力引数を渡して、`python3 Deploy_Oracle_Functions.py` コマンドを実行します。すべての機能を展開するには時間がかかります。その後、ファイルを削除してクラウドシェルを閉じることができます。

Terraform Template-2 の展開

Template-2 は、アラーム、関数を呼び出すための ONS トピックなど、アラーム作成に関連するリソースを展開します。Template-2 の展開は、Terraform Template-1 の展開に似ています。

ステップ 1 OCI ポータルにログインします。

地域は、画面の右上隅に表示されます。目的の地域内に存在していることを確認してください。

ステップ 2 **[デベロッパーサービス（Developer Service）] > [リソースマネージャ（Resource Manager）] > [スタック（Stack）] > [スタックの作成（Create Stack）]**の順に選択します。

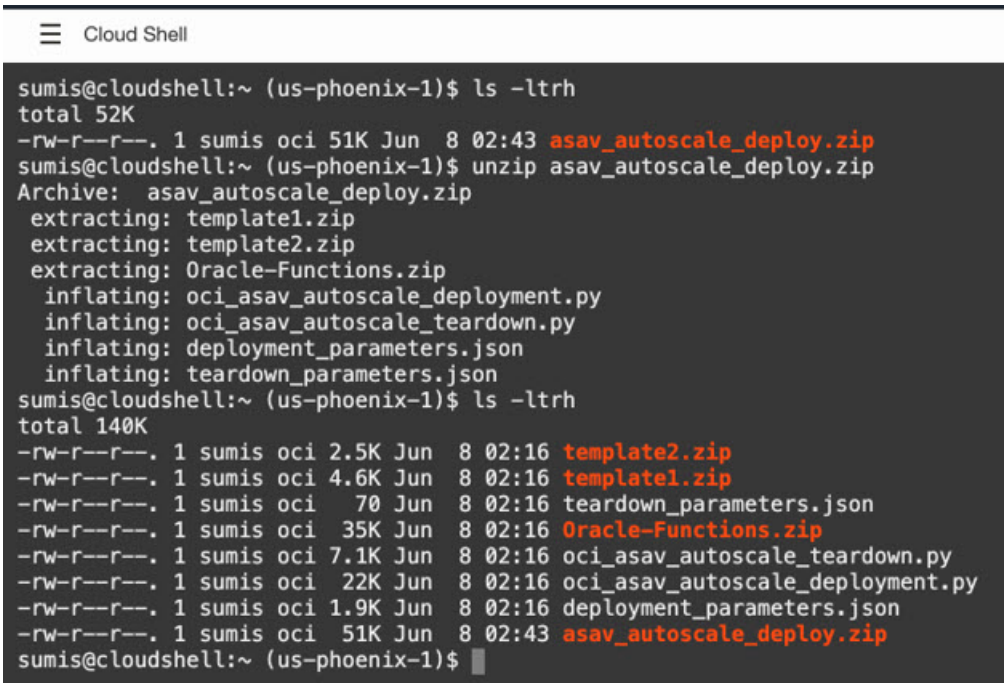
Terraform 設定のソースとして、ターゲットフォルダにある *Terraform template template2.zip* を選択します。

ステップ 3 次のステップで、**Terraformアクション（Terraform Actions）] > [適用（Apply）]**をクリックします。

クラウドシェルを使用した Auto Scale の導入

展開のオーバーヘッドを回避するために、簡単なエンドツーエンドの展開スクリプトを呼び出して、自動スケールソリューション（terraform template1、template2、および Oracle 関数）を展開できます。

ステップ 1 対象フォルダ内にある *asav_autoscale_deploy.zip* ファイルをクラウドシェルにアップロードして、ファイルを抽出します。



ステップ 2 `python3 make.py build` コマンドを実行する前に、*deployment_parameters.json* の入力パラメータが更新されていることを確認してください。

ステップ 3 Auto Scale ソリューションの導入を開始するには、クラウドシェルで `python3 oci_asav_autoscale_deployment.py` コマンドを実行します。

ソリューションの展開が完了するまでに約 10 ～ 15 分かかります。

ソリューションの展開中にエラーが発生した場合、エラーログが保存されます。

展開の検証

すべてのリソースが展開され、Oracle関数がアラームとイベントに接続されているかどうかを検証します。デフォルトでは、インスタンスプールのインスタンスの最小数と最大数はゼロです。OCI UI でインスタンスプールを編集して、必要な最小数と最大数に設定できます。これにより、新しい ASAv 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。この検証をポストすると、ASAv の実際の要件を展開できます。

（注） OCI 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 ASAv インスタンスを**スケールイン保護**として指定します。

Auto Scale のアップグレード

Auto Scale スタックのアップグレード

このリリースではアップグレードはサポートされていません。スタックを再導入する必要があります。

ASAv VM のアップグレード

このリリースでは、ASAv VM のアップグレードはサポートされていません。必要な ASAv イメージを使用してスタックを再導入する必要があります。

インスタンスプール

1. インスタンスプール内のインスタンスの最小数と最大数を変更するには、次の手順を実行します。

   [デベロッパーサービス（Developer Services）] > [機能（Function）] > [アプリケーション名（Terraform template-1で作成済み）（Application Name(created by Terraform Template 1)）] > [設定（Configuration）] をクリックします。

   min_instance_count と max_instance_count をそれぞれ変更します。

2. インスタンスの削除/終了は、スケールインと同等ではありません。インスタンスプール内のいずれかのインスタンスがスケールインアクションではなく外部アクションのために削除/終了された場合、インスタンスプールは自動的に新しいインスタンスを開始して回復します。

3. Max_instance _count では、スケールアウトアクションのしきい値制限を定義しますが、UI を介してインスタンスプールのインスタンス数を変更することでしきい値を上回ることが

できます。UIのインスタンス数が、OCIアプリケーションで設定されたmax_instance_count未満であることを確認します。それ以外の場合は、適切なしきい値に増やします。

4. アプリケーションから直接インスタンスプール内のインスタンスの数を減らしても、プログラムで設定されたクリーンアップアクションは実行されません。両方のロードバランサからバックエンドがドレインおよび削除されないため、ASAv に供与されているライセンスは失われます。

5. 何らかの理由で、ASAv インスタンスに異常があり応答せず、一定期間 SSH 経由で到達できない場合、インスタンスがインスタンスプールから強制的に削除され、ライセンスが失われる可能性があります。

Oracle 関数

- Oracle 関数は、実際には Docker イメージです。Docker イメージは、OCI コンテナレジストリのルートディレクトリに保存されます。Docker イメージは削除しないでください。Auto Scale ソリューションで使用される関数も削除されます。
- Terraform Template-1 によって作成された OCI アプリケーションには、Oracle 関数が正しく動作するために必要な重要な環境変数が含まれています。必須でない限り、これらの環境変数の値もフォーマットも変更しないでください。加えられた変更は、新しいインスタンスにのみ反映されます。

ロードバランサのバックエンドセット

OCI でインスタンスプールにロードバランサを関連付ける場合、ASAv で管理インターフェースとして設定されたプライマリインターフェースを使用した方法のみサポートされています。したがって、内部インターフェイスは内部ロードバランサのバックエンドセットに紐づけられます。外部インターフェイスは、外部ロードバランサのバックエンドセットに紐づけられます。これらのIPはバックエンドセットに自動的に追加されたり、削除されたりしません。Auto Scale ソリューションでは、これら両方のタスクをプログラムで処理します。ただし、外部アクション、メンテナンス、トラブルシューティングの場合は、手動で実行する必要性が生じることがあります。

要件に応じて、リスナーとバックエンドセットを使用して、ロードバランサーで追加のポートを開くことができます。今後のインスタンス IP はバックエンドセットに自動的に追加されますが、既存のインスタンス IP は手動で追加する必要があります。

ロードバランサでのリスナーの追加

ロードバランサでポートをリスナーとして追加するには、**[OCI]** > **[ネットワーキング（Networking）]** > **[ロードバランサ（Load Balancer）]** > **[リスナー（Listener）]** > **[リスナーの作成（Create Listener）]**に移動します。

バックエンドをバックエンドセットに登録

ASAvインスタンスをロードバランサに登録するには、ASAvインスタンスの外部インターフェイス IP を外部ロードバランサのバックエンドセットでバックエンドとして設定する必要があります。内部インターフェイス IP は、内部ロードバランサーのバックエンドセットでバック

エンドとして設定する必要があります。使用しているポートがリスナーに追加されていることを確認してください。

OCI のAuto Scale 設定の削除

Terraform を使用して導入されたスタックは、OCI の Resource Manager を使用して、同じ方法で削除できます。スタックを削除すると、そのスタックによって作成されたすべてのリソースが削除され、これらのリソースに関連付けられているすべての情報が完全に削除されます。

（注） スタックを削除する場合は、インスタンスプールのインスタンスの最小数を 0 にして、インスタンスが終了するまで待つことを推奨します。そうすることで、すべてのインスタンスの削除が容易になり、インスタンスが残りません。

手動による削除するか、クラウドシェルを使用した Auto Scale の削除 を使用できます。

手動による削除

エンドツーエンドの Auto Scale ソリューションの削除は、次の 3 つの手順で構成されます。Terraform Template-2 スタックの削除、Oracle 関数の削除、次いで Terraform Template-1 スタックの削除

Terraform Template-2 スタックの削除

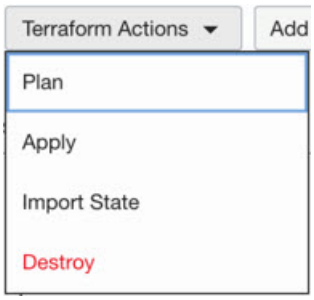
自動スケール設定を削除するには、最初に Terraform Template-2 スタックを削除する必要があります。

ステップ 1 OCI ポータルにログインします。

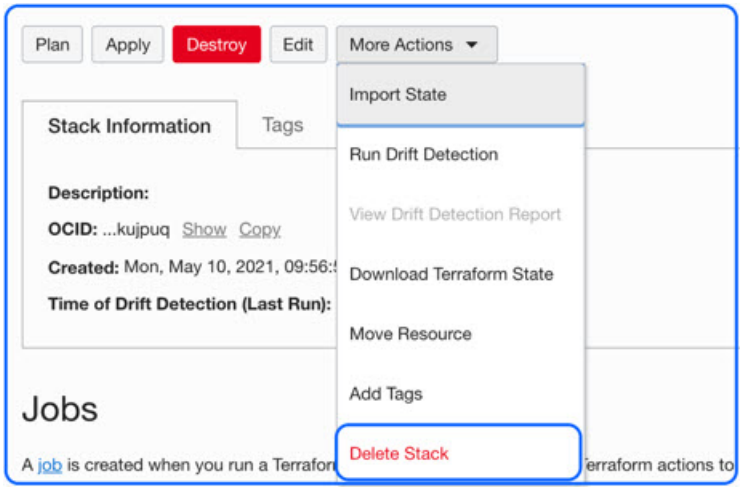
地域は、画面の右上隅に表示されます。目的の地域内に存在していることを確認してください。

ステップ 2 **[デベロッパーサービス（Developer Service）] > [リソースマネージャ（Resource Manager）] > [スタック（Stack）]**の順に選択します。

ステップ 3 Terraform Template-2 によって作成されたスタックを選択し、次の図に示すように [Terraformアクション（Terraform Actions）] ドロップダウンメニューで [破棄（Destroy）] を選択します。



破棄ジョブが作成されます。リソースが順次削除されるまで時間がかかります。破棄ジョブが完了したら、下の図に示すようにスタックを削除できます。



ステップ 4 Oracle 関数の削除に進みます。

Oracle 関数の削除

Oracle 関数の展開は Terraform Template スタック展開の一部としてではなく、クラウドシェルを使用して個別にアップロードします。したがって、削除も Terraform スタックの削除ではサポートされていません。Terraform Template-1 によって作成された OCI アプリケーション内のすべての Oracle 関数を削除する必要があります。

ステップ 1 OCI ポータルにログインします。

地域は、画面の右上隅に表示されます。目的の地域内に存在していることを確認してください。

ステップ 2 **[開発者サービス（Developer Services）]** > **[機能（Functions）]**の順に選択します。Template-1 スタックで作成されたアプリケーション名を選択します。

ステップ 3 このアプリケーション内で各機能にアクセスして削除します。

Terraform Template-1 スタックの削除

（注） Template-1 スタックの削除は、すべての Oracle 関数を削除した後にのみ成功します。

Terraform Template-2 の削除と同じです。

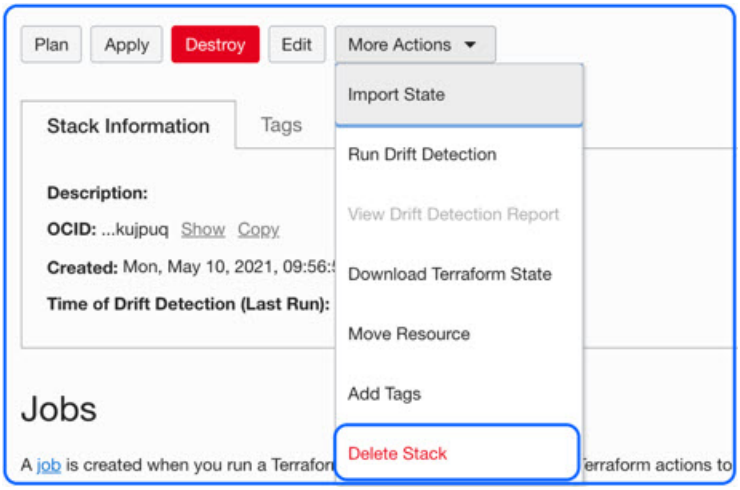
ステップ 1 OCI ポータルにログインします。

地域は、画面の右上隅に表示されます。目的の地域内に存在していることを確認してください。

ステップ 2 **[デベロッパーサービス（Developer Service）] > [リソースマネージャ（Resource Manager）] > [スタック（Stack）]**の順に選択します。

ステップ 3 Terraform Template-2 によって作成されたスタックを選択し、[Terraformアクション（Terraform Actions）] ドロップダウンメニューで [破棄（Destroy）] を選択します。破棄ジョブが作成されます。リソースが順次削除されるまで時間がかかります。

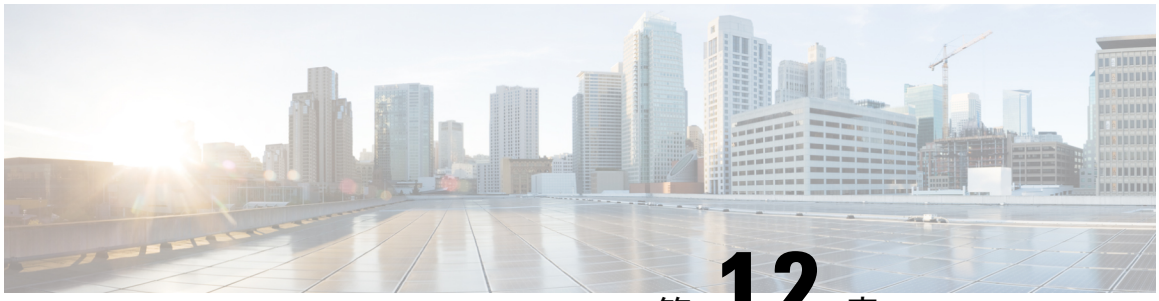
ステップ 4 破棄ジョブが完了したら、下の図に示すように、[その他の操作（More Actions）] ドロップダウンメニューからスタックを削除できます。



Terraform Template-1 スタックの削除が成功したら、すべてのリソースが削除され、残存しているリソースがないことを確認する必要があります。

クラウドシェルを使用した Auto Scale の削除

スクリプトを使用してスタックやオラクル関数を削除するには、 コマンドシェルで `python3 oci_asav_autoscale_teardown.py` コマンドを実行します。スタックが手動で展開されている場合は、stack1 と stack2 のスタック ID を更新し、*teardown_parameters.json* ファイルのアプリケーション ID を更新します。



第 12 章

Google Cloud Platform への ASAv の展開

Google Cloud Platform（GCP）に ASAv を導入できます。

GCP への ASAv の展開について

GCP を使用すると、Google と同じインフラストラクチャでアプリケーション、Web サイト、サービスを構築、展開、および拡張できます。

ASAv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。ASAv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。

GCP マシンタイプのサポート

ASAv 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。

ASAv は、次の汎用 *N1*、*N2*、およびコンピューティング最適化 *C2* GCP マシンタイプをサポートしています。

表 *23*：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	メモリ（GB）
c2-standard-4	4	16

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

表 *24 :* サポートされる汎用マシンタイプ

マシンタイプ	属性	
	vCPU	メモリ（**GB**）
n1-standard-4	4	15
n1-standard-8	8	30
n1-standard-16	16	60
n2-standard-4	4	16
n2-standard-8	8	32
n2-standard-16	16	64
n1-highcpu-8	8	7.2
n1-highcpu-16	16	14.4
n2-highcpu-8	8	8
n2-highcpu-16	16	16
n2-highmem-4	4	32
n2-highmem-8	8	64
n2-highmem-16	16	128

- ASAv には、少なくとも 3 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。
- メモリ最適化マシンタイプはサポートされていません。

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

C2 コンピューティング最適化マシンタイプの制限事項

コンピューティング最適化 C2 マシンタイプには、次の制約があります。

- コンピューティング最適化マシンタイプでは、リージョン永続ディスクを使用できません。詳細については、Google のドキュメント「Adding or resizing regional persistent disks」を参照してください。
- 汎用マシンタイプおよびメモリ最適化マシンタイプとは異なるディスク制限が適用されます。詳細については、Google のドキュメント「Block storage performance」を参照してください。
- 一部のゾーンとリージョンでのみ使用できます。詳細については、Google のドキュメント「Available regions and zones」を参照してください。
- 一部の CPU プラットフォームでのみ使用できます。詳細については、Google のドキュメント「CPU platforms」を参照してください。

ASAv と GCP の前提条件

- 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。
- ASAv へのライセンス付与。ASAv 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。「Licenses: Smart Software Licensing」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：ASDM に ASAv 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：内部ホストに ASAv を接続するために使用されます。
  - 外部インターフェイス：ASAv 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。
- 通信パス：
  - ASAv にアクセスするためのパブリック IP。
- ASAv システム要件については、Cisco ASA の互換性 [英語] を参照してください。

ASAv 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 上の ASAv は、次の機能をサポートしています。

- GCP 仮想プライベートクラウド（VPC）への展開
- 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

サポートされない機能

GCP 上の ASAv は、次の機能をサポートしていません。

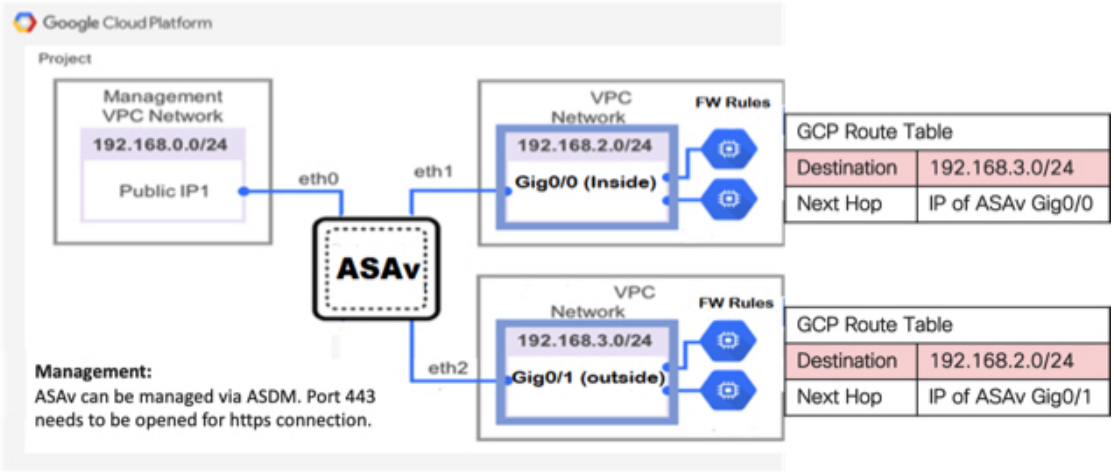
- IPv6
  - インスタンスレベルの IPv6 設定は GCP ではサポートされません
  - ロードバランサだけが IPv6 接続を受け入れて IPv4 経由で GCP インスタンスにプロキシできます
- ジャンボ フレーム
- ASAv ネイティブ HA
- 自動スケール
- トランスペアレント/インライン/パッシブ モード

GCP 上の ASAv のネットワークトポロジの例

次の図は、ASAv 用の 3 つのサブネット（管理、内部、外部）が GCP 内に設定されているルーテッド ファイアウォール モードの ASAv の推奨ネットワークトポロジを示しています。

Google Cloud Platform への ASAv の展開

図 *54 : GCP* 展開での *ASAv* の例



Google Cloud Platform への ASAv の展開

Google Cloud Platform（GCP）に ASAv を導入できます。

VPC ネットワークの作成

始める前に

ASAv の導入では、ASAv を導入する前に 3 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。
- 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、ASAv 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、ASAv 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。「GCP 上の ASAv のネットワークトポロジの例 （246 ページ）」を参照してください。

ステップ 1 GCP コンソールで、**[Networking]** > **[VPC network]** > **[VPC networks]** を選択し、[Create VPC Network] をクリックします。

ステップ 2 [Name] フィールドに、VPC ネットワークのわかりやすい名前を入力します（例：*vpc-asiasouth-mgmt*）。

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 [New subnet] の [Name] フィールドに、適切な名前を入力します（例：*vpc-asiasouth-mgmt*）。

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。3 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 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、VPC の残り 2 つの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

ASAvインスタンスの展開中に（SSHおよびHTTPS接続を許可するために）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。GCP 上の ASAv インスタンスの作成 （249 ページ）を参照してください。要件に応じて、内部および外部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。

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 [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：*asav-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。

GCP 上の ASAv インスタンスの作成

以下の手順を実行して、GCP Marketplace から提供される Cisco ASA 仮想ファイアウォール（ASAv）を使用して ASAv インスタンスを導入します。

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 ASA virtual firewall (ASAv)」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。

ステップ 5 [Deployment name] でインスタンスの一意の名前を指定します。

ステップ 6 [ゾーン（Zone）] で ASAv を導入するゾーンを選択します。

ステップ 7 [Machine type] で適切なマシンタイプを選択します。サポートされるマシンタイプの一覧については、GCP への ASAv の展開について （243 ページ）を参照してください。

ステップ 8 （オプション）[SSH key (optional)] で 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。

ステップ 9 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。

ステップ 10 （任意）[起動スクリプト（Startup script）] で ASAv の第 0 日用構成を指定します。day0 構成は、ASAv の初回起動時に適用されます。

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ことができる day0 構成の例を示します。

ASA コマンドの詳細については、『ASA 構成ガイド』および『ASA コマンドリファレンス』を参照してください。

重要 この例からテキストをコピーする場合は、サードパーティのテキストエディタまたは検証エンジンでスクリプトを検証して、形式エラーを防止し、無効な Unicode 文字を削除する必要があります。

```
!ASA Version 9.15.1

interface management0/0

management-only
nameif management
security-level 100
ip address dhcp setroute
no shut
```

```
!
same-security-traffic permit inter-interface
same-security-traffic permit intra-interface
!
crypto key generate rsa modulus 2048
ssh 0 0 management
ssh timeout 60
ssh version 2
username admin password cisco123 privilege 15
username admin attributes
service-type admin
! required config end
dns domain-lookup management
dns server-group DefaultDNS
name-server 8.8.8.8
```

ステップ 11 プロビジョニングされるディスク容量についてデフォルトの [Boot disk type] と [Boot disk size in GB] を維持します。

ステップ 12 [Network interfaces] でインターフェイスを設定します。

- 管理
- inside
- outside

（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。

a) **[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。

b) **[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。

c) [完了（Done）] をクリックします。

ステップ 13 [Firewall] で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。
- [Allow HTTPS traffic from the Internet (ASDM access)] チェックボックスをオンにして、HTTPS 接続を許可します。

ステップ 14 **[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 15 [展開（Deploy）] をクリックします。

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。

GCP 上の ASAv インスタンスへのアクセス

展開中に SSH（ポート 22 経由の TCP 接続）を許可するファイアウォールルールがすでに有効化されていることを確認します。詳細については、GCP 上の ASAv インスタンスの作成 （249 ページ）を参照してください。

このファイアウォールルールにより、ASAv インスタンスへのアクセスが可能になり、次の方法を使用してインスタンスに接続できます。

- 外部 IP（External IP）
  - 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
- シリアル コンソール
- Gcloud コマンドライン

詳細については、Google ドキュメントの 『Connecting to instances』を参照してください。

（注） 第 0 日用構成で指定したログイン情報を使用するか、インスタンスの起動時に作成した SSH キーペアを使用して、ASAv インスタンスにログインできます。

外部 IP を使用した ASAv インスタンスへの接続

ASAv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ASAvインスタンスにアクセスできます。

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 ASAv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ASAv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。

（注） 第 0 日用構成で指定したログイン情報を使用するか、インスタンスの起動時に作成した SSH キーペアを使用して、ASAv インスタンスにログインできます。

SSH を使用した ASAv インスタンスへの接続

UNIX スタイルのシステムから ASAv 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。

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>
```

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。

<username> は、ASAv 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 ASAv インスタンスへの接続

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 ASAv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 ASAv インスタンスへの接続

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 ASAv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。

CPU 使用率とレポート

CPU使用率レポートには、指定された時間内に使用されたCPUの割合の要約が表示されます。通常、コアはピーク時以外には合計 CPU 容量の約 30 ～ 40% で動作し、ピーク時は約 60 ～ 70% の容量で動作します。

ASA Virtual の vCPU 使用率

ASA Virtual の vCPU 使用率には、データパス、制御ポイント、および外部プロセスで使用されている vCPU の量が表示されます。

GCP で報告される vCPU 使用率には、ASA Virtual 使用率に加えて、次のものが含まれます。

- ASA Virtual アイドル時間
- ASA Virtual マシンに使用された %SYS オーバーヘッド
- vSwitch、vNICおよびpNICの間を移動するパケットのオーバーヘッド。このオーバーヘッドは非常に大きくなる場合があります。

CPU 使用率の例

CPU 使用率の統計情報を表示するには、**show cpu usage** コマンドを使用します。

例

```
Ciscoasa#show cpu usage

CPU □□□□5□□□ 1%□1 □□□ 2%□5 □□□ 1%
```

報告された vCPU の使用率が大幅に異なる例を次に示します。

- ASA Virtual レポート：40%
- DP：35%
- 外部プロセス：5%
- ASA（ASA Virtual レポート）：40%
- ASA アイドル ポーリング：10%
- オーバーヘッド：45%

オーバーヘッドは、ハイパーバイザ機能の実行、および vSwitch を使用した NIC と vNIC の間のパケット転送に使用されています。

GCP CPU 使用率レポート

GCP コンソールでインスタンス名をクリックし、[モニタリング（Monitoring）] タブをクリックします。CPU 使用率が表示されます。

Compute Engine では、使用状況エクスポート機能を使用して、Compute Engine の使用状況の詳細レポートを Google Cloud Storage バケットにエクスポートできます。使用状況レポートには、リソースの有効期間に関する情報が表示されます。たとえば、プロジェクト内で `n2-standard-4` マシンタイプを実行している VM インスタンスの数と、各インスタンスの実行時間を確認できます。永続ディスクのストレージスペースや、Compute Engine の他の機能に関する情報も確認できます。

ASA Virtual と GCP のグラフ

ASA Virtual と GCP の間には CPU % の数値に違いがあります。

- GCP グラフの数値は ASA Virtual の数値よりも常に大きくなります。
- GCP ではこの値は「%CPU usage」と呼ばれ、ASA Virtual ではこの値は「%CPU utilization」と呼ばれます。

用語「%CPU utilization」と「%CPU usage」は別のものを意味しています。

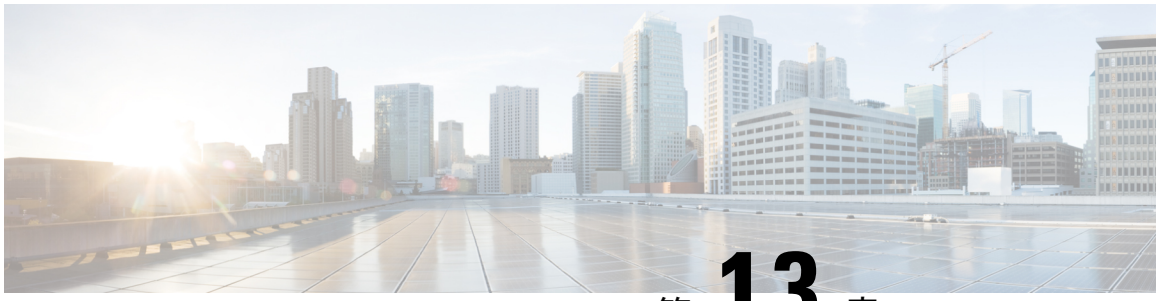
- CPU utilization は、物理 CPU の統計情報を提供します。
- CPU usage は CPU のハイパースレッディングに基づいた論理 CPU の統計情報を提供します。しかし、1 つの vCPU のみが使用されるため、ハイパースレッディングは動作しません。

GCP では「%CPU usage」は次のように計算されます。

アクティブに使用された仮想 CPU の量。使用可能な CPU の合計に対する割合として指定されます。

この計算は、ホストから見た CPU 使用率であり、ゲスト オペレーティング システムから見た CPU 使用率ではありません。また、これは仮想マシンで使用可能なすべての仮想 CPU の平均 CPU 使用率になります。

たとえば、1 個の仮想 CPU を搭載した 1 つの仮想マシンが、4 個の物理 CPU を搭載した 1 台のホストで実行されており、その CPU 使用率が 100% の場合、仮想マシンは、1 個の物理 CPU をすべて使用しています。仮想 CPU の使用率は、「MHz 単位の使用率 / 仮想 CPU の数 x コア周波数」として計算されます。



第 13 章

GCP への ASAv Auto Scale ソリューションの展開

GCP 上の ASAv 向けの Auto Scale ソリューション

以下のセクションでは、Auto Scale ソリューションのコンポーネントが GCP の ASAv 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

ASAv 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。

ASAv 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 に基づくスケーリングメトリックのサポート。
- ASAv 展開とマルチ可用性ゾーンのサポート。
- スケールアウトされた ASAv インスタンスに完全に自動化された構成を自動適用。

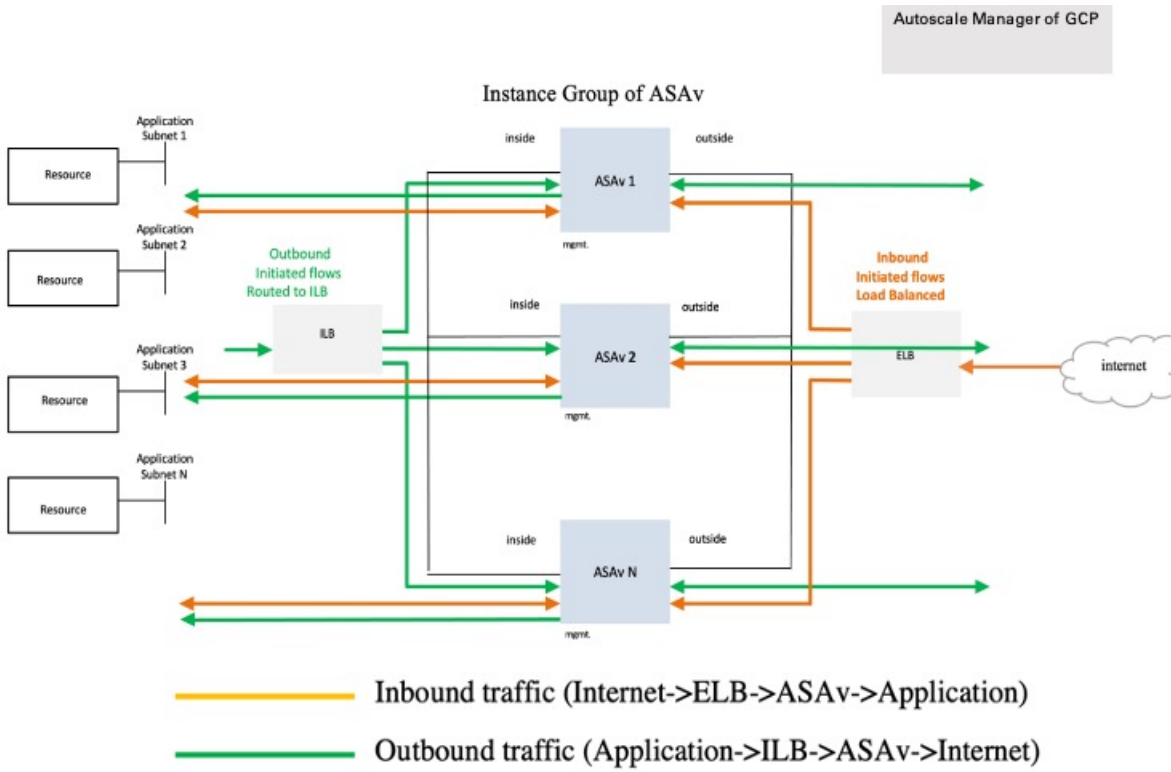
- ロードバランサとマルチ可用性ゾーンのサポート。
- シスコでは、導入を容易にするために、Auto Scale for GCP の導入パッケージを提供しています。

Auto Scale の導入例

ASAv Auto Scale for GCP は、ASAv 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 ASAv インスタンスに分散させます。その後、ファイアウォールからアプリケーションにトラフィックが転送されます。
- 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 ASAv インスタンスに分散させます。その後、ファイアウォールからインターネットにトラフィックが転送されます。
- 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 ASAv 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 *55 : ASAv* 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、ASAv 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

ASAv Auto Scale for GCP は GCP Deployment Manager のテンプレートをベースとした導入であり、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）を利用します。

ASAv 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 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。

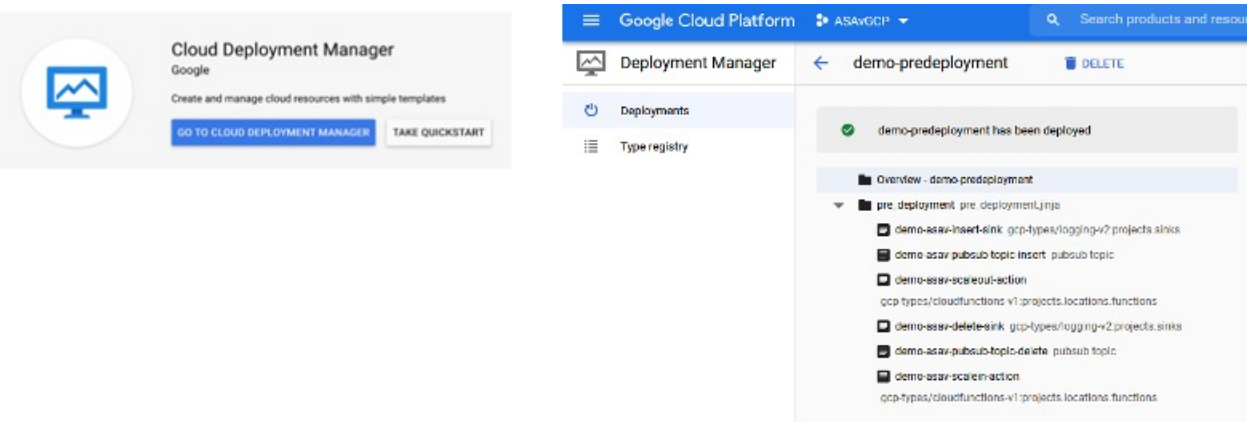
Auto Scale ソリューションのコンポーネント

ASAv 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、Python または 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。

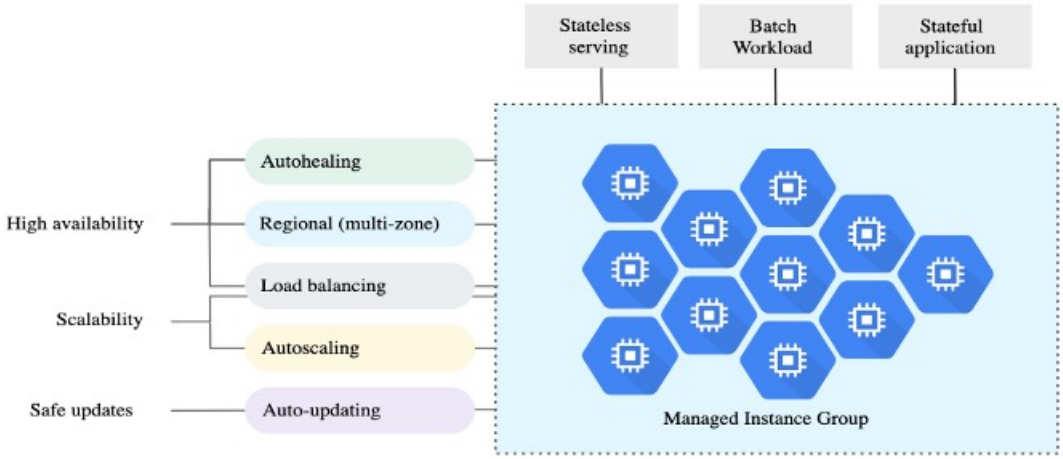
図 *56* : 導入マネージャビュー



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。

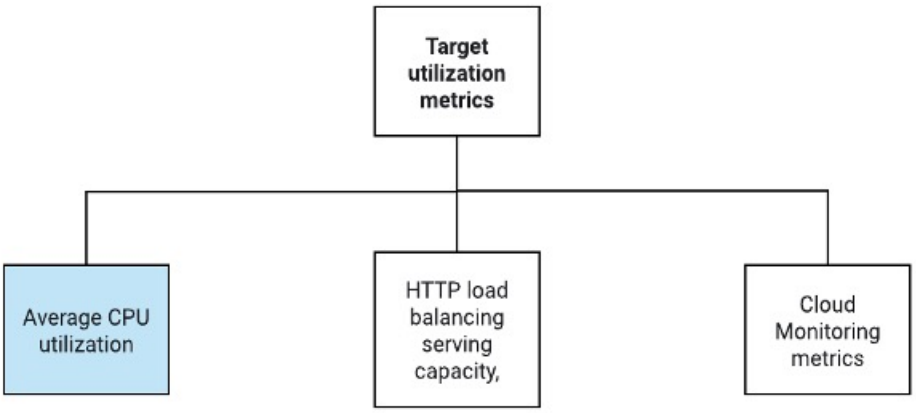
図 *57* : インスタンスグループの機能



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。

図 *58 :* ターゲット使用率メトリック



サーバーレスクラウド機能

Instance Group Manager でインスタンスが起動したときに、サーバーレスの Google Cloud 機能を使用して、SSH パスワードの設定、パスワードの有効化、ホスト名の変更を行います。

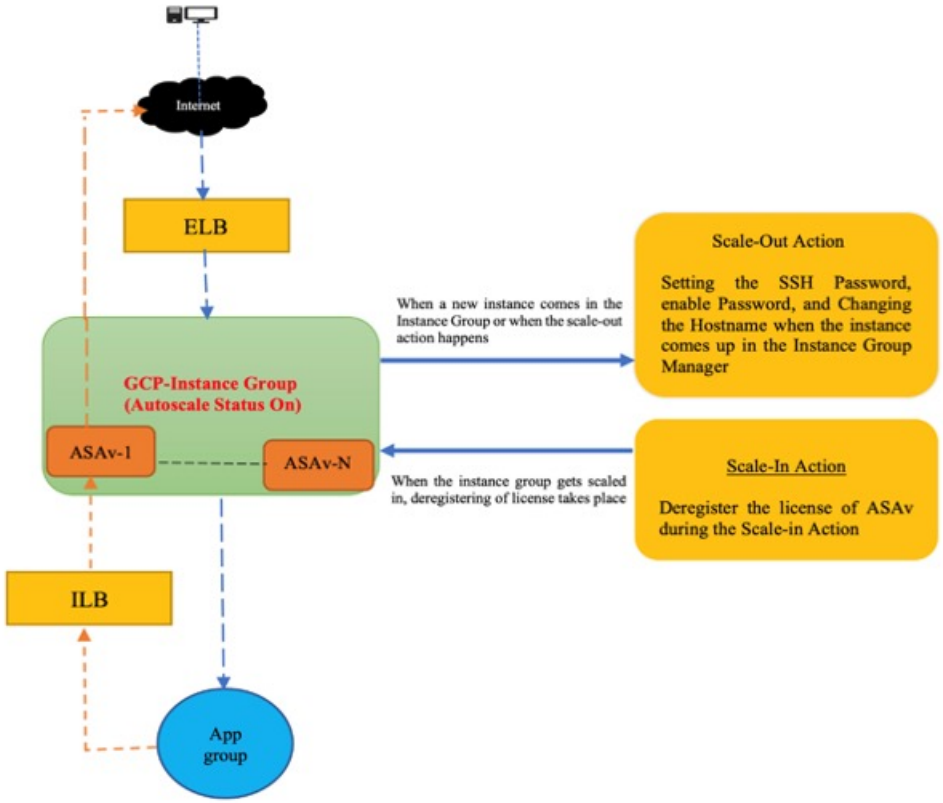
- スケールアウト中に新しい ASAv インスタンスがインスタンスグループに追加された場合、スケールアウトプロセスを常に監視できないため、SSH パスワードを設定し、パスワードを有効にして、ホスト名を変更する必要があります。
- クラウド機能は、スケールアウトプロセス中に クラウドのパブリック/サブトピックを介してトリガーされます。また、スケールアウト時のインスタンス追加専用のフィルタを備えたログシンクもあります。

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、ASAv インスタンスからライセンス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。
- クラウド機能は、トリガーされると削除対象の ASAv 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 ソリューションの大まかな概要

図 *59 : Auto Scale* ソリューションの概要



Auto Scale ソリューションの前提条件

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。

ネットワーキング

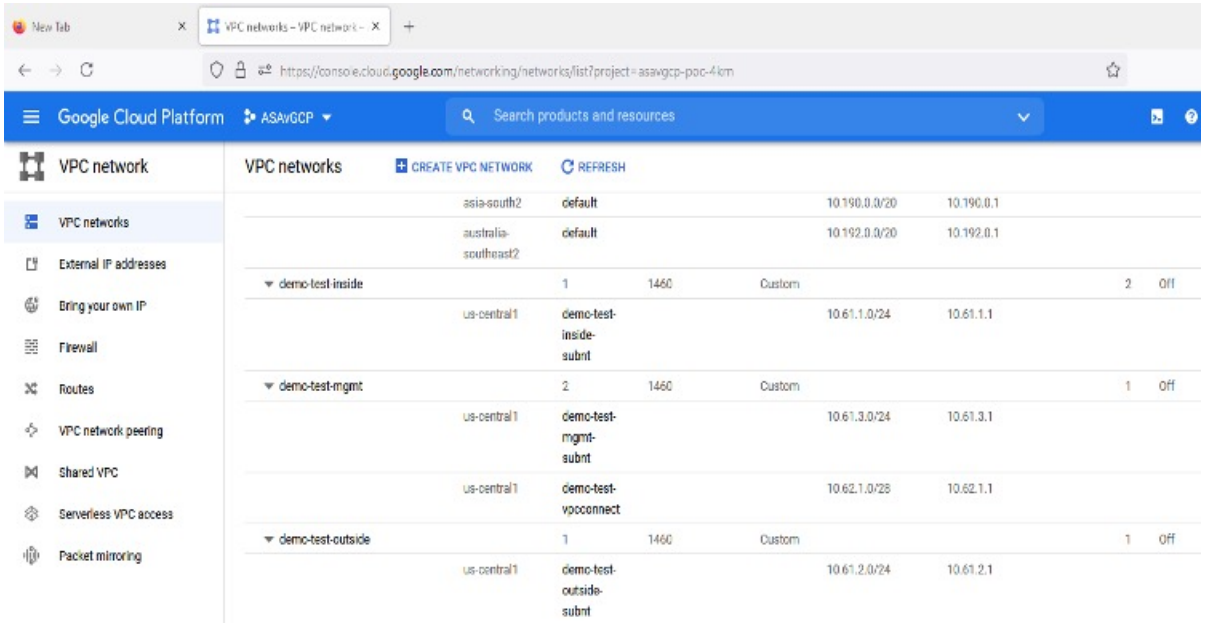
3 つの VPC が使用可能または作成されていることを確認してください。Auto Scale 展開では、ネットワークリソースの作成、変更、管理は行われません。

ASAv には 3 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 3 つのサブネットが必要です。

- 管理トラフィック

- 内部トラフィック
- 外部トラフィック

図 *60 : VPC* ネットワークビュー



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。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 ASAv 間の正常性プローブに必要です。サーバーレス機能と ASAv 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）。

ASA 構成ファイルの準備

Deployment Manager jinja 構成ファイルに含める ASAv 構成ファイルを準備します。この構成は、プロジェクト内の ASAv のインスタンステンプレートで起動スクリプトとして使用されます。

構成ファイルに最低限必要な内容は以下のとおりです。

- すべてのインターフェイスに DHCP IP 割り当てを設定します。

- GCP ロードバランサはトラフィックを Nic0 にのみ転送するため、Nic0 は「外部」としてマークする必要があります。
- Nic0 は IP 転送のみをサポートしているため、ASAv への SSH 接続に使用されます。
- ASA 設定の外部インターフェイスで SSH を有効にします。
- 外部インターフェイスから内部インターフェイスにトラフィックを転送するための NAT 構成を作成します。
- 目的のトラフィックを許可するアクセスポリシーを作成します。
- リソースの正常性ステータスについては、適切な NAT ルールを使用して、リソースの正常性プローブをメタデータサーバーにリダイレクトする必要があります。

参考用に ASA 構成ファイルの例を次に示します。

```
!ASA Version 9.15.1.10
!Interface Config
interface G0/0
nameif inside
security-level 100
ip address dhcp setroute
no shutdown


interface G0/1
nameif management
security-level 50
ip address dhcp setroute
no shutdown

interface M0/0
no management-only
nameif outside
security-level 0
ip address dhcp setroute
no shutdown
!
same-security-traffic permit inter-interface
!
!Due to some constraints in GCP,
!"GigabitEthernet0/0" will be used as a Management interface
!"Management0/0" will be used as a data interface
crypto key generate rsa modulus 2048
ssh 0.0.0.0 0.0.0.0 management
ssh version 2
ssh timeout 60
aaa authentication ssh console LOCAL
ssh authentication publickey {{ properties["publicKey"] }}
username admin privilege 15
username admin attributes
service-type admin

! required config end
dns domain-lookup management
dns server-group DefaultDNS
name-server 8.8.8.8
!
access-list all extended permit ip any any
access-list out standard permit any4
```

```
access-group all global
! Objects
object network metadata
host 169.254.169.254
object network ilb
host $(ref.{{ properties["resourceNamePrefix"] }}-ilb-ip.address)
object network hc1
subnet 35.191.0.0 255.255.0.0
object network hc2
subnet 130.211.0.0 255.255.63.0
object network elb
host $(ref.{{ properties["resourceNamePrefix"] }}-elb-ip.address)
object network appServer
host 10.61.2.3
object network defaultGateway
subnet 0.0.0.0 0.0.0.0
! Nat Rules
nat (inside,outside) source dynamic hc1 ilb destination static ilb metadata
nat (inside,outside) source dynamic hc2 ilb destination static ilb metadata
nat (inside,outside) source dynamic defaultGateway interface
 !
 object network appServer
nat (inside,outside) static $(ref.{{ properties["resourceNamePrefix"] }}-elb-ip.address)
object network defaultGateway
nat (outside,inside) dynamic interface
! Route Add
route inside 0.0.0.0 0.0.0.0 10.61.1.1 2
route management 0.0.0.0 0.0.0.0 10.61.3.1 3
license smart register idtoken <licenseIDToken>
```

GCP クラウド機能パッケージの構築

ASAv GCP Auto Scale ソリューションでは、圧縮された ZIP パッケージの形式でクラウド機能を提供する 2 つのアーカイブファイルを作成する必要があります。

- `scalein-action.zip`
- `scaleout-action.zip`

`scalein-action.zip` および `scaleout-action.zip` パッケージの作成方法については、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 ASAv デバイスを作成できます。

表 *25*：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	リソースの作成タイプ
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	新規作成（New）
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。	
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	
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	
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	

パラメータ名	使用できる値/タイプ	説明	リソースの作成タイプ
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-healthallowall	既存
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	既存
insideVPCName	文字列	内部 VPC の名前。 例：demo-test-inside	既存
insideVPCSubnet	文字列	内部サブネットの名前。 例： demo-test-inside-subnt	既存
machineType	文字列	ASAv VM のマシンタイプ。 例：e2-standard-4	
maxASACount	整数	インスタンスグループで許可される ASAv インスタンスの最大数。 例：3	
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	
mgmtVPCName	文字列	管理 VPC の名前。 例：demo-test-mgmt	

パラメータ名	使用できる値/タイプ	説明	リソースの作成タイプ
mgmtVPCSubnet	文字列	管理サブネットの名前。 例： demo-test-mgmt-subnt	
minASACount	整数	任意の時点でインスタンスグループで使用可能な ASAv インスタンスの最小数。 例 1	
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	
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	
publicKey	文字列	ASAv VM の SSH キー。	
sourceImageURL	文字列	プロジェクトで使用する ASAv のイメージ。 例： https://www.googleapis.com/compute/v1/projects/cisco-public/global/images/cisco-asav-9-15-1-15	
アプリケーションサーバーの IP アドレス	文字列	内部 Linux マシンの内部 IP アドレス。 例：10.61.1.2	

パラメータ名	使用できる値/タイプ	説明	リソースの作成タイプ
内部 VPC ゲートウェイの IP アドレス	文字列	内部 VPC のゲートウェイ。 例：10.61.1.1	
管理 VPC ゲートウェイの IP アドレス	文字列	管理 VPC のゲートウェイ。 例：10.61.3.1	

Auto Scale ソリューションの展開

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。

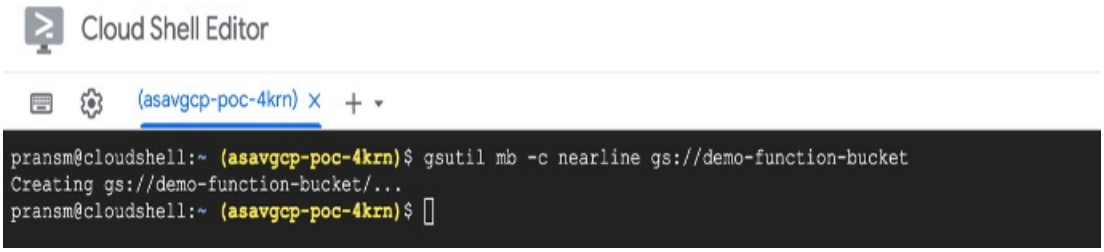
```
git clone git_url -b branch_name
```

例：

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
```

例：



ステップ 3 ZIP 形式の圧縮パッケージを作成します。

a) `scalein_action` および `scaleout_action` 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

requirements.txt

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

（注） フォルダー内を移動し、ファイルを選択して右クリックし、「圧縮||」を選択します。archive' を使用して、GCP が読み取れる .zip を作成します。

ステップ 4 Zip 形式の圧縮パッケージ（`scaleout-action.zip` および `scalein-action.zip`）を Cloud Editor 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 テンプレートから Cloud Editor ワークスペースにアップロードします。

- asav_autoscale.jinja
- asav_autoscale_params.yaml
- pre_deployment.jinja
- pre_deployment.yaml

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 scaleout-action.zip gs://bucket_name`
- `gsutil cp scalein-action.zip gs://bucket_name`

例：

```
pransm@cloudshell:~ (asavgcp-poc-4krn)$ gsutil cp scaleout-action.zip gs://demo-function-bucket
Copying file://scaleout-action.zip [Content-Type=application/zip]...
/ [1 files][  3.3 KiB/  3.3 KiB]
Operation completed over 1 objects/3.3 KiB.
pransm@cloudshell:~ (asavgcp-poc-4krn)$ gsutil cp scalein-action.zip gs://demo-function-bucket
Copying file://scalein-action.zip [Content-Type=application/zip]...
/ [1 files][  3.3 KiB/  3.3 KiB]
Operation completed over 1 objects/3.3 KiB.
pransm@cloudshell:~ (asavgcp-poc-4krn)$
```

ステップ 7 内部、外部、および管理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（10.8.2.0/28 など）が必要です。

ステップ 8 内部、外部、および管理インターフェイス用に 3 つのファイアウォールルールが必要です。また、ファイアウォールルールではヘルスチェックプローブを許可する必要があります。

ステップ 9 事前展開および ASAv Auto Scale 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) `asav_autoscale_params.yaml` ファイルを開き、以下のパラメータを更新します。

- **resourceNamePrefix**: <resourceNamePrefix>
- **region**: <region>
- **serviceAccountMailId**: <serviceAccountMailId>
- **publicKey**: <publicKey>
- **insideVPCName**: <Inside-VPC-Name>

- **insideVPCSubnet**: <Inside-VPC-Subnet>
- **outsideVPCName**: <Outside-VPC-Name>
- **outsideVPCSubnet**: <Outside-VPC-Subnet>
- **mgmtVPCName**: <Mgmt-VPC-Name>
- **mgmtVPCSubnet**: <Mgmt-VPC-Subnet>
- **insideFirewallRuleName**: <Inside-Network-Firewall-Tag>
- **outsideFirewallRuleName**: <Outside-Network-Firewall-Tag>
- **mgmtFirewallRuleName**: <Mgmt-Network-Firewall-Tag>
- healthCheckFirewallRuleName :<HealthCheck-IP-Firewall-Tag>
- **machineType**: <machineType>

（注） ASAv 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 ASAv 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。

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

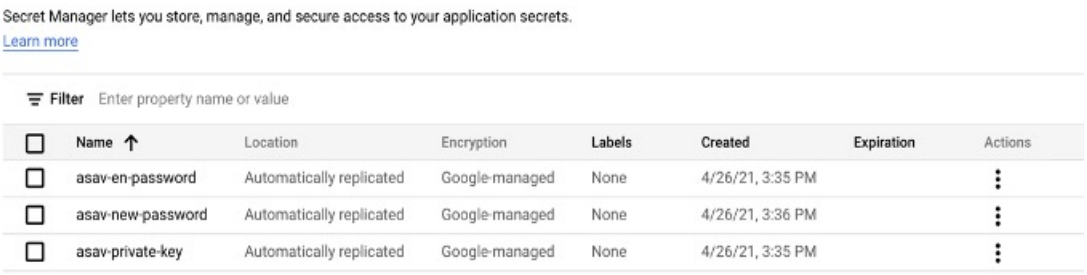
- **host**: <Application server IP address>
- **route inside 0.0.0.0 0.0.0.0**: <Inside VPC Gateway IP address> 2
- **route management 0.0.0.0 0.0.0.0**: <Management VPC Gateway IP address> 3
- **license smart register idtoken**: <licenseIDToken>

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

- **resourceNamePrefix**: <resourceNamePrefix>
- **region**: <region>
- **serviceAccountMailId**: <serviceAccountMailId>
- **vpcConnectorName**: <VPC-Connector-Name>
- **bucketName**: <bucketName>

ステップ 10 Secret Manager GUI を使用して、次の 3 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。

- asav-en-password
- asav-new-password
- asav-private-key


Secret Manager lets you store, manage, and secure access to your application secrets.
Learn more

Filter Enter property name or value

Name	Location	Encryption	Labels	Created	Expiration	Actions
asav-en-password	Automatically replicated	Google-managed	None	4/26/21, 3:35 PM		
asav-new-password	Automatically replicated	Google-managed	None	4/26/21, 3:36 PM		
asav-private-key	Automatically replicated	Google-managed	None	4/26/21, 3:35 PM		

ステップ 11 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 12 事前展開の YAML 構成を展開します。

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

例：

```
gcloud deployment-manager deployments create demo-predeployment
--config pre_deployment.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
```

NAME	TYPE	STATE
demo-asav-delete-sink	gcp-types/logging-v2:projects.sinks	COMPLETED
demo-asav-insert-sink	gcp-types/logging-v2:projects.sinks	COMPLETED
demo-asav-pubsub-topic-delete	pubsub.v1.topic	COMPLETED
demo-asav-pubsub-topic-insert	pubsub.v1.topic	COMPLETED
demo-asav-scalein-action	gcp-types/cloudfunctions-v1:projects.locations.functions	COMPLETED
demo-asav-scaleout-action	gcp-types/cloudfunctions-v1:projects.locations.functions	COMPLETED

ステップ 13 ASAv Auto Scale の展開を作成します。

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

例：

```
gcloud deployment-manager deployments create demo-asav-autoscale
--config asav_autoscale_params.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.
```

NAME	TYPE	STATE
demo-asav-autoscaler	compute.v1.regionAutoscaler	COMPLETED
demo-asav-backend-service-elb	compute.v1.regionBackendService	COMPLETED
demo-asav-backend-service-ilb	compute.v1.regionBackendService	COMPLETED
demo-asav-fr-elb	compute.v1.forwardingRule	COMPLETED
demo-asav-fr-ilb	compute.v1.forwardingRule	COMPLETED
demo-asav-hc-elb	compute.v1.regionHealthChecks	COMPLETED
demo-asav-hc-ilb	compute.v1.healthCheck	COMPLETED
demo-asav-health-check	compute.v1.healthCheck	COMPLETED
demo-asav-instance-group	compute.v1.regionInstanceGroupManager	COMPLETED
demo-asav-instance-template	compute.v1.instanceTemplate	COMPLETED
demo-elb-ip	compute.v1.address	COMPLETED

ステップ 14 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].
```

NAME	NETWORK	DEST_RANGE	NEXT_HOP	PRIORITY
demo-ilb	sdt-test-asav-inside	0.0.0.0/0	10.7.1.60	1000

ステップ 15 Cloud Router と Cloud NAT を作成します。

```
gcloud compute routers create <cloud-router-name>
--project=<project-name> --region <region> --network=<outside-vpc-name>
--advertisement-mode=custom
```

```
gcloud compute routers nats create <cloud-nat-name>
--router=<cloud-router-name> --nat-all-subnet-ip-ranges --auto-allocate-nat-external-ips
--region=<region>
```

例：

```
gcloud compute routers create demo-cloud-router --project=asavgcp-poc-4krn
--region us-central1 --network=sdt-test-asav-outside --advertisement-mode=custom
Creating router [demo-cloud-router]...done.
```

NAME	REGION	NETWORK
demo-cloud-router	us-central1	sdt-test-asav-outside

```
gcloud compute routers nats create demo-cloud-nat
--router=demo-cloud-router --nat-all-subnet-ip-ranges
--auto-allocate nat-external-ips --region=us-central1
Creating NAT [demo-cloud-nat] in router [demo-cloud-router]...done.
```

Auto Scale ロジック

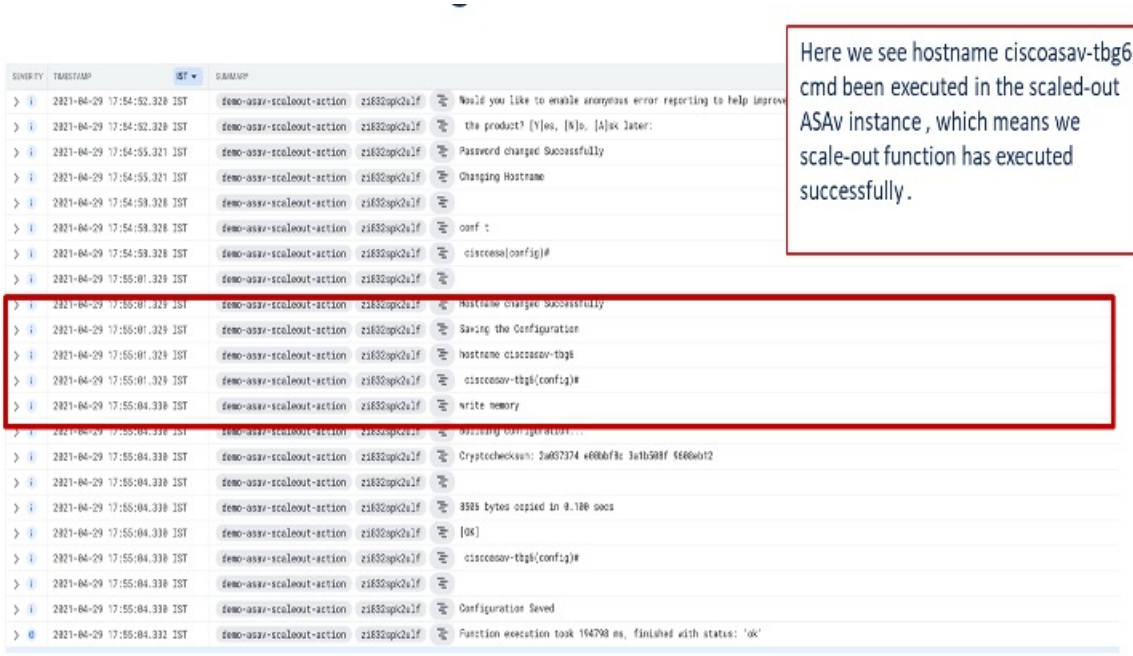
- オートスケーラは、ターゲット CPU 使用率レベルを、インスタンスグループ内の一定期間にわたるすべての vCPU の平均使用量の一部として扱います。
- 合計 vCPU の平均使用率がターゲット使用率を超えると、オートスケーラによって VM インスタンスが追加されます。合計 vCPU の平均使用率がターゲット使用率よりも低い場合、オートスケーラはインスタンスを削除します。
- たとえば、0.75 のターゲット使用率を設定すると、オートスケーラはインスタンスグループ内のすべての vCPU の平均使用率を 75% に維持するように指示されます。
- スケーリングの決定では、CPU 使用率メトリ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

Auto Scale の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。

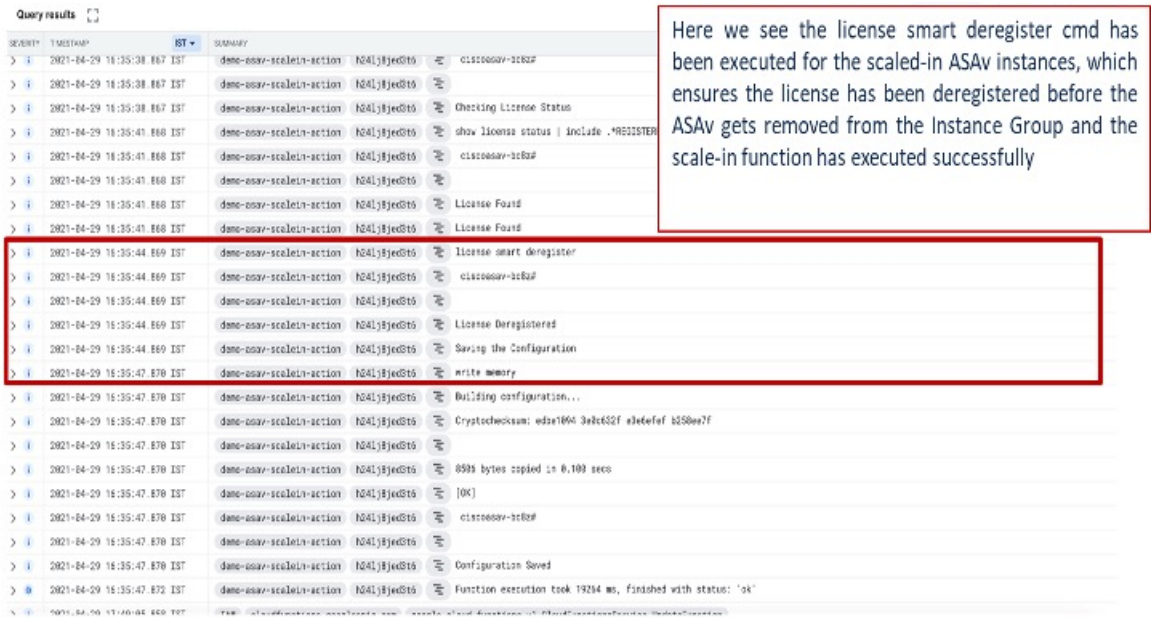
- スケールアウト機能のログ

図 *61* : スケールアウト機能のログ



- スケールイン機能のログ

図 *62 :* スケールイン機能のログ



Auto Scale のガイドラインと制約事項

- IPv4 だけがサポートされます。
- サポートされているライセンスは BYOL のみです。PAYG は GCP 上の ASAv では利用できません。
- 外部ロードバランサはテンプレートによって作成されるため、ロードバランサのパブリック IP に関する特定の DNS 要件は範囲外です。
- アプリケーションはユーザーが作成したロードバランサの背後にあると想定され、ASAv は（トラフィックを特定のアプリケーション IP に直接送信する代わりに）すべてのトラフィックをこのロードバランサにルーティングします。
- TAG、冗長性、およびロードバランサアフィニティ構成の必要性に関する詳細は考慮されていません。
- ASAv ログイン情報は次の状況で表示されます。
  - サーバーレスコードのクリアテキスト。
  - インスタンスグループ内のすべてのインスタンス。
  - インスタンステンプレート（共有 GCP アカウントを使用している場合）。

  このような機密データは、GCP の公開キーサービスを使用して保護できます。

重要 シスコでは、ライセンスサーバーへの ASAv の登録を定期的に追跡して、スケールアウトされた ASA が期待どおりにライセンスサーバーに登録されているか、スケールインされた ASAv インスタンスがライセンスサーバーから削除されているか確認することを推奨しています。

Auto Scale のトラブルシューティング

次に、ASAv 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からのみ構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。
- テンプレートの導入中のエラー：「<>」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、および同じ導入名がすでに存在することを確認します。
- Google 関数が ASAv 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。
- ASAv 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。
- ライセンスの登録に失敗：ライセンス ID トークンが正しいことを確認します。また、Cloud NAT が作成されており、ASAv が tools.cisco.com にアクセスできることを確認します。

第 14 章

OpenStack への ASAv の展開

OpenStack に ASAv を導入できます。

OpenStack への ASAv の展開について

OpenStack 環境に ASAv を展開できます。OpenStack は、パブリック クラウドとプライベート クラウドの、クラウド コンピューティング プラットフォームを構築および管理するための一連のソフトウェア ツールで、KVM ハイパーバイザと緊密に統合されています。

ASAv に対する OpenStack プラットフォームのサポートを有効にすると、オープン ソース クラウド プラットフォームで ASAv を実行できます。OpenStack は、KVM ハイパーバイザを使用して仮想リソースを管理します。ASAv デバイスは、KVM ハイパーバイザですでにサポートされています。したがって、OpenStack のサポートを有効にするためにカーネルパッケージやドライバを追加する必要はありません。

ASAv と OpenStack の前提条件

- software.cisco.com から ASAv qcow2 ファイルをダウンロードし、Linux ホストに格納します。

  http://www.cisco.com/go/asa-software

- ASAv は、オープンソースの OpenStack 環境と Cisco VIM 管理対象 OpenStack 環境での展開をサポートします。

  OpenStack のガイドラインに従って OpenStack 環境をセットアップします。

- オープンソースの OpenStack ドキュメントを参照してください。

  Stein リリース：https://docs.openstack.org/project-deploy-guide/openstack-ansible/stein/overview.html

  Queens リリース：https://docs.openstack.org/project-deploy-guide/openstack-ansible/queens/overview.html

- Cisco Virtualized Infrastructure Manager（VIM）OpenStack のドキュメント（Cisco Virtualized Infrastructure Manager のマニュアル、3.4.3 ～ 3.4.5）を参照してください。

- ASAv へのライセンス付与。ASAv 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。「Licenses: Smart Software Licensing」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
  - 内部および外部インターフェイス
- 通信パス：
  - 管理インターフェイス：ASDM に ASAv を接続するために使用され、トラフィック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 ASAv 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）：ASAv 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。
- 通信パス：
  - ASAv にアクセスするためのフローティング IP。
- サポートされている ASAv の最小バージョン：
  - ASA 9.16.1
- OpenStack の要件については、「OpenStack の要件」を参照してください。
- ASAv システム要件については、Cisco ASA の互換性 [英語] を参照してください。

ASAv および OpenStack のガイドラインと制限事項

サポートされる機能

OpenStack 上の ASAv は次の機能をサポートします。

- OpenStack 環境のコンピューティングノードで実行されている KVM ハイパーバイザへの ASAv の展開
- OpenStack CLI
- Heat テンプレートベースの展開
- OpenStack Horizon ダッシュボード
- 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
- CLI および ASDM を使用した ASAv の管理
- ドライバ：VIRTIO、VPP、および SRIOV

サポートされない機能

OpenStack 上の ASAv は以下をサポートしません。

- 自動スケール
- OpenStack Stein リリースと Queens リリース以外の OpenStack リリース
- Ubuntu 18.04 バージョンと Red Hat Enterprise Linux（RHEL）7.6 以外のオペレーティングシステム

OpenStack の要件

OpenStack 環境は、サポートされているハードウェアとソフトウェアの次の要件に準拠している必要があります。

表 *26 :* ハードウェアおよびソフトウェアの要件

カテゴリ	サポートされるバージョン	注記
サーバー	UCS C240 M5	2 台の UCS サーバーを推奨します。os-controller ノードと os-compute ノードに 1 台ずつです。
要因	VIRTIO、IXGBE、I40E	サポートされているドライバは次のとおりです。
オペレーティングシステム	Ubuntu Server 18.04	これは、UCS サーバーで推奨されている OS です。

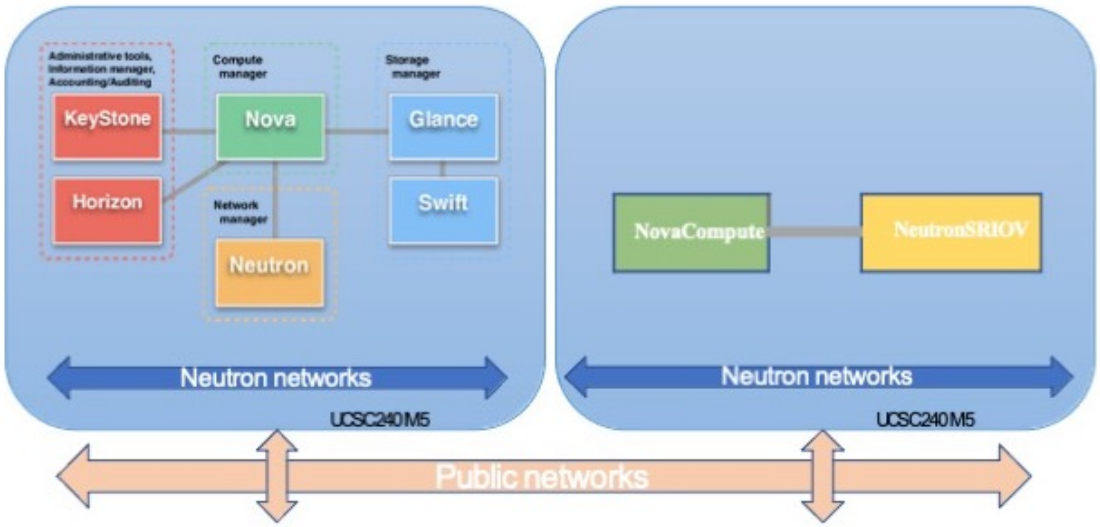
カテゴリ	サポートされるバージョン	注記
OpenStack バージョン	Stein リリース	さまざまな OpenStack リリースの詳細については、次の URL を参照してください。 https://releases.openstack.org/

表 *27 : Cisco VIM Managed OpenStack* のハードウェアとソフトウェアの要件

カテゴリ	サポートされるバージョン	注記
サーバ ハードウェア	UCS C220-M5/UCS C240-M4	os-controller ノードごとに 3 台、os-compute ノードに 2 台以上で、5 台の UCS サーバーを推奨します。
ドライバ	VIRTIO、SRIOV、および VPP	サポートされているドライバは次のとおりです。
Cisco VIM バージョン	Cisco VIM 3.4.4 サポート対象： • オペレーティングシステム - Red Hat Enterprise Linux 7.6 • OpenStack バージョン - OpenStack 13.0（Queens リリース）	詳細については、シスコ仮想インフラストラクチャ マネージャのドキュメント 3.4.3 ～ 3.4.5 を参照してください。 さまざまな OpenStack リリースの詳細については、https://releases.openstack.org/ を参照してください。
	Cisco VIM 4.2.1 サポート対象： • オペレーティングシステム - Red Hat Enterprise Linux 8.2 • OpenStack バージョン - OpenStack 16.1（トレイン リリース）	詳細については、シスコ仮想インフラストラクチャ マネージャのドキュメント 4.2.1 を参照してください。 さまざまな OpenStack リリースの詳細については、https://releases.openstack.org/ を参照してください。

図 *63 : OpenStack* プラットフォームトポロジ

OpenStack プラットフォームトポロジは、2 台の UCS サーバーでの一般的な OpenStack セットアップを示しています。



OpenStack 上の ASAv のネットワークトポロジの例

次の図は、ASAv 用の 3 つのサブネット（管理、内部、外部）が OpenStack 内に設定されているルーテッドファイアウォールモードの ASAv の推奨ネットワークトポロジを示しています。

図 *64 : OpenStack* への *ASAv* の導入例

OpenStack への ASAv の展開

シスコでは、ASAv を展開するためのサンプルの Heat テンプレートを提供しています。OpenStack インフラストラクチャのリソースを作成する手順は、ネットワーク、サブネット、およびルータインターフェイスを作成するために、Heat テンプレート（`deploy_os_infra.yaml`）ファイルで結合されます。ASAv の展開手順は大まかに次の部分に分類されます。

- ASAv qcow2 イメージを OpenStack Glance サービスにアップロードします。
- ネットワーク インフラストラクチャを作成します。

- ネットワーク
- サブネット
- ルータ インターフェイス

- ASAv インスタンスを作成します。
  - フレーバ
  - セキュリティ グループ
  - フローティング IP
  - インスタンス

次の手順を使用して、OpenStack に ASAv を展開できます。

OpenStack への ASAv イメージのアップロード

qcow2 イメージ（`asav-<version>.qcow2`）を OpenStack コントローラノードにコピーし、イメージを OpenStack Glance サービスにアップロードします。

始める前に

Cisco.com から ASAv qcow2 ファイルをダウンロードし、Linux ホストに格納します。

http://www.cisco.com/go/asa-software

（注） Cisco.com のログインおよびシスコ サービス契約が必要です。

ステップ 1 qcow2 イメージファイルを OpenStack コントローラノードにコピーします。

ステップ 2 ASAv イメージを OpenStack Glance サービスにアップロードします。

```
root@ucs-os-controller:$ openstack image create <image_name> --public --disk-
format qcow2 --container-format bare --file ./<asav_qcow2_file>
```

ステップ 3 ASAv イメージが正常にアップロードされたことを確認します。

root@ucs-os-controller:$ openstack image list

例：

```
root@ucs-os-controller:$ openstack image
list+--------------------------------------+-------------------+--------+
| ID                                   | Name              | Status
|+--------------------------------------+-------------------+--------+
| 06dd7975-0b6e-45b8-810a-4ff98546a39d | asav-<version>-image | active
|+--------------------------------------+-------------------+--------+
```

アップロードしたイメージとそのステータスが表示されます。

次のタスク

deploy_os_infra.yaml テンプレートを使用してネットワーク インフラストラクチャを作成します。

OpenStack と ASAv のネットワーク インフラストラクチャの作成

始める前に

Heatテンプレートファイルは、フレーバ、ネットワーク、サブネット、ルータインターフェイス、セキュリティグループルールなど、ネットワークインフラストラクチャとASAvに必要なコンポーネントを作成するために必要です。

- deploy_os_infra.yaml
- env.yaml

ASAv バージョンのテンプレートは次の GitHub リポジトリから入手できます。

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

重要 シスコが提供するテンプレートはオープンソースの例として提供しているものであり、通常の Cisco TAC サポートの範囲内では扱われていません。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。

ステップ 1 インフラストラクチャ Heat テンプレートファイルを展開します。

root@ucs-os-controller:$ openstack stack create <stack-name> -e <environment files name> -t <deployment file name>

例：

```
root@ucs-os-controller:$ openstack stack create infra-stack -e env.yaml -t deploy_os_infra.yaml
```

ステップ 2 インフラストラクチャ スタックが正常に作成されたかどうかを確認します。

root@ucs-os-controller:$ openstack stack list

次のタスク

OpenStack で ASAv インスタンスを作成します。

OpenStack での ASAv インスタンスの作成

ASAv Heat テンプレートのサンプルを使用して、OpenStack に ASAv を導入します。

始める前に

OpenStack で ASAv を展開するには、次の Heat テンプレートが必要です。

- `deploy_asav.yaml`

ASAv バージョンのテンプレートは次の GitHub リポジトリから入手できます。

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

重要 シスコが提供するテンプレートはオープンソースの例として提供しているものであり、通常の Cisco TAC サポートの範囲内では扱われていません。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。

ステップ 1 ASAv Heat テンプレートファイル（`deploy_asav.yaml`）を展開して、ASAv インスタンスを作成します。

root@ucs-os-controller:$ openstack stack create asav-stack -e env.yaml-t deploy_asav.yaml

例：

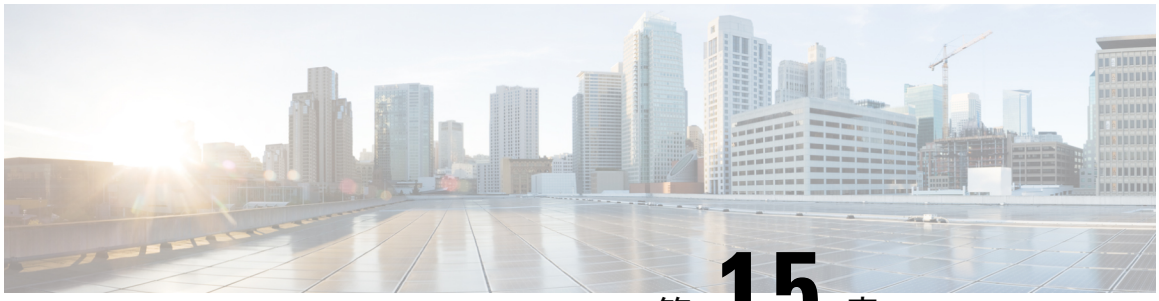
```
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| id                  | 14624af1-e5fa-4096-bd86-c453bc2928ae |
| stack_name          | asav-stack                           |
| description         | ASAvtemplate                         |
| creation_time       | 2020-12-07T14:55:05Z                 |
| updated_time        | None                                 |
| stack_status        | CREATE_IN_PROGRESS                   |
| stack_status_reason | Stack CREATE started                 |
+---------------------+--------------------------------------+
```

ステップ 2 ASAv スタックが正常に作成されたことを確認します。

root@ucs-os-controller:$ openstack stack list

例：

```
+--------------------------------------+----------+----------------------------------+--------------+----------------------+--------------+
| ID                                   | Stack Name  | Project                          | Stack
Status    | Creation Time    | Updated Time |
+--------------------------------------+----------+----------------------------------+--------------+----------------------+--------------+
| 14624af1-e5fa-4096-bd86-c453bc2928ae | asav-stack  | 13206e49b48740fdafca83796c6f4ad5 |
CREATE_COMPLETE | 2020-12-07T14:55:05Z | None         |
| 198336cb-1186-45ab-858f-15ccd3b909c8 | infra-stack | 13206e49b48740fdafca83796c6f4ad5 |
CREATE_COMPLETE | 2020-12-03T10:46:50Z | None         |
+--------------------------------------+----------+----------------------------------+--------------+----------------------+--------------+
```



第 15 章

Nutanix 上で ASAv を展開する

この章では、ASAv を Nutanix 環境に展開する手順について説明します。

Nutanix で ASAv を使い始める

Cisco 適応型セキュリティ仮想アプライアンス（ASAv）は、仮想化環境に包括的なファイアウォール機能を提供し、データセンタートラフィックとマルチテナント環境のセキュリティを強化します。

Nutanix 上で ASAv を展開します。

Nutanix での ASAv のガイドラインと制限

重要 ASAv は、8 GB のディスクストレージサイズで展開されます。ディスク容量のリソース割り当てを変更することはできません。

ASAv を展開する前に、次のガイドラインと制限事項を確認します。

推奨される vNIC

最適なパフォーマンスを得るには、次の vNIC をお勧めします。

VirtIO：10 Gbps の動作をサポートしますが、CPU サイクルも必要な準仮想化ネットワークドライバです。

CPU ピニング

Nutanix 環境で ASAv を機能させるには、CPU ピニングが必要です。「CPU ピンニングの有効化 （60 ページ）」を参照します。

ハイ アベイラビリティのためのフェールオーバー

フェールオーバー配置の場合は、スタンバイ装置が同じライセンス権限付与を備えていることを確認してください（たとえば、両方の装置が 2 Gbps の権限付与であることなど）。

重要 ハイアベイラビリティペアを作成するときは、同じ順序で各 ASAv にデータインターフェイスを追加する必要があります。完全に同じインターフェイスが各 ASAv に追加されているが、順序が異なる場合、ASAv コンソールにエラーが表示され、フェールオーバー機能に影響を与える可能性があります。

一般的な注意事項

- サポートされるインターフェイスの最大数は 10 です。10 を超える数のインターフェイスを追加しようとすると、エラーメッセージが表示されます。

（注）
- デフォルトでは、ASAv は同じサブネット上に管理インターフェイスと内部インターフェイスを設定します。
- ネットワーク インターフェイスを変更するときは、ASAv デバイスをオフにする必要があります。

- デフォルトでは、ASAv は、**異なるサブネット**上に管理インターフェイスと内部インターフェイスの両方を設定したことを前提としています。管理インターフェースには「IP address DHCP setroute」があり、デフォルトゲートウェイは DHCP によって提供されます。
- ASAv は、3 つ以上のインターフェイスを使用して最初の起動時にパワーアップする必要があります。システムは、3 つのインターフェースなしでは展開されません。
- ASAv は、データトラフィック用に 1 つの管理インターフェイス（nic0）と最大 9 つのネットワーク インターフェイス（nic1-9）の合計 10 のインターフェイスをサポートします。データトラフィックのネットワーク インターフェイスは、任意の順序に従うことができます。

（注） ASAv のネットワーク インターフェイスの最小数は、3 つのデータインターフェイスです。

- コンソールアクセスの場合、ターミナルサーバーは telnet を介してサポートされます。
- サポートされている vCPU とメモリのパラメータは次のとおりです。

CPU	メモリ	ASAv プラットフォームのサイズ	ライセンスのタイプ
1	2 GB	1vCPU/2 GB（デフォルト）	1G（ASAv10）
4	8 GB	4 vCPU/8 GB	2G（ASAv30）
8	16 GB	8 vCPU/16 GB	10G（ASAv50）
16	32 GB	16vCPU/32GB	20G（ASAv100）

サポートされる機能

- ルーテッドモード（デフォルト）
- トランスペアレント モード

（注） マルチノードクラスタのサービスチェーンは、トランスペアレントモードではサポートされていません。

インターフェイスのネットワークアダプタ、送信元ネットワーク、宛先ネットワークに関する以下の用語索引を参照してください。

ネットワーク アダプタ	送信元ネットワーク	宛先ネットワーク	機能
vnic0	Management0-0	Management0/0	管理
vnic1	GigabitEthernet0-1	GigabitEthernet 0/1	Outside
vnic2	GigabitEthernet0-2	GigabitEthernet 0/2	内側
vnic3-9	データ	データ	データ

Proxmox VE 上の ASAv

Proxmox Virtual Environment（VE）は、Nutanix 仮想マシンを管理できるオープンソースのサーバー仮想化プラットフォームです。Proxmox VE は、Web ベースの管理インターフェイスも提供します。

Proxmox VE に ASAv を導入する場合は、エミュレートされたシリアルポートを持つように VM を設定する必要があります。シリアルポートがないと、スタートアッププロセス中に ASAv がループ状態になります。すべての管理タスクは、Proxmox VE Web ベース管理インターフェイスを使用して実行できます。

（注） Unix シェルまたは Windows Powershell に慣れている上級ユーザー向けに、Proxmox VE は仮想環境のすべてのコンポーネントを管理するコマンド ライン インターフェイスを提供します。このコマンド ライン インターフェイスには、インテリジェントなタブ補完機能と UNIX の man ページ形式の完全なドキュメントがあります。

ASAv を正しく開始するには、VM にシリアルデバイスを設定する必要があります。

1. メイン Management Center の左側のナビゲーションツリーで ASAv VM を選択します。
2. 仮想マシンの電源をオフにします。
3. **Hardware** > **Add** > **Network Device** を選択して、シリアルポートを追加します。
4. 仮想マシンの電源をオンにします。
5. Xterm.js を使用して ASAv VM にアクセスします。

ゲスト/サーバーで端末をセットアップしてアクティブ化する方法については、Proxmox シリアル端末のページを参照してください。

サポートされない機能

- Nutanix AHV 上の ASAv は、インターフェイスのホットプラグをサポートしていません。ASAv の電源がオンになっているときに、インターフェイスを追加または削除しないでください。
- Nutanix AHV は、Single Root I/O Virtualization（SR-IOV）または Data Plane Development Kit-Open vSwitch（DPDK-OVS）をサポートしていません。

（注） Nutanix AHV は、VirtIO を使用したゲスト内 DPDK をサポートします。詳細については、AHV での DPDK サポートを参照してください。

関連資料

- Nutanix Release Notes
- Nutanix Field Installation Guide
- Nutanix でのハードウェアのサポート
- Nutanix AHV での Virtio-Net Multi-Queue サポート

ASAv と Nutanix のシステム要件

ASA のバージョン

9.16.2

ASAv メモリ、vCPU、およびディスクのサイジング

ASAv の導入に使用される特定のハードウェアは、導入されるインスタンスの数と使用要件によって異なります。ASAv の各インスタンスには、サーバー上での最小リソース割り当て（メモリ容量、CPU 数、およびディスク容量）が必要です。

ASAv ライセンス

- ASAv CLI 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 ASA コンフィギュレーション ガイド』の「*ASAv* のスマート ソフトウェア ライセンシングの設定」を参照してください。

Nutanix のコンポーネントとバージョン

コンポーネント	バージョン
Nutanix Acropolis OS（AOS）	5.15.5 LTS 以降
Nutanix クラスタチェック（NCC）	4.0.0.1
Nutanix AHV	20201105.12 以降

Nutanix にASAv を展開する方法

ステップ	タスク	詳細情報
1	前提条件を確認します。	ASAv と Nutanix を展開するための前提条件（292 ページ）
2	ASAv qcow2 ファイルを Nutanix 環境にアップロードします。	QCOW2 ファイルを Nutanix にアップロード（292 ページ）
3	仮想マシンの展開時に適用される初期構成データを使用して、第 0 日の構成ファイルを準備します。	第 0 日のコンフィギュレーション ファイルの準備 （293 ページ）
4	Nutanix にASAv を展開します。	ASAv を Nutanix に展開する （295 ページ）
5	ASAv を起動します。	ASAv の起動 （296 ページ）

ASAv と Nutanix を展開するための前提条件

- Cisco.com から ASAv qcow2 ファイルをダウンロードし、Linux ホストに格納します。

  http://www.cisco.com/go/asa-software

（注） Cisco.com のログインおよびシスコ サービス契約が必要です。

- ASA ソフトウェアおよび HyperFlex ハイパーバイザの互換性については、「Cisco Asa Compatibility」を参照してください。

QCOW2 ファイルを Nutanix にアップロード

ASAv を Nutanix 環境に展開するには、Prism Web コンソールで ASAv qcow2 ディスクファイルからイメージを作成する必要があります。

始める前に

Cisco.com から qcow2 ディスクファイルをダウンロードします：https://software.cisco.com/download/navigator.html）

ステップ 1 Nutanix Prism Web コンソールにログインします。

ステップ 2 歯車アイコンをクリックして [設定（Settings）] ページを開きます。

ステップ 3 左側のペインで [イメージの設定（Image Configuration）] をクリックします。

ステップ 4 [Upload Image] をクリックします。

ステップ 5 イメージを作成します。

1. イメージの名前を入力します。
2. [イメージタイプ（Image Type）] ドロップダウンリストから、[ディスク（DISK）] を選択します。
3. [ストレージコンテナ（Storage Container）] ドロップダウンリストから、目的のコンテナを選択します。
4. ASAv qcow2 ディスクファイルの場所を指定します。

   URL を指定して Web サーバーからファイルをインポートすることも、ワークステーションからファイルをアップロードすることもできます。
5. [保存（Save）] をクリックします。

ステップ 6 [イメージの設定（Image Configuration）] ページに新しいイメージが表示されるまで待ちます。

第 0 日のコンフィギュレーション ファイルの準備

ASAv を展開する前に、第 0 日の構成ファイルを準備できます。このファイルは、仮想マシンの導入時に適用される初期設定データを含むテキスト ファイルです。

第 0 日のコンフィギュレーション ファイルを使用して展開する場合、プロセスで、ASAv アプライアンスの初期設定全体を実行できます。

ファイルでは、以下を指定できます。

- システムのホスト名。
- 管理者アカウントの新しい管理者ユーザー名とパスワード。
- 最初のファイアウォール モード。最初のファイアウォール モード（**ルーテッド**または**トランスペアレント**）を設定します。

  ローカルを使用して展開を管理する予定の場合は、ファイアウォールモードに**ルーテッド**のみ入力できます。ASAvデバイスマネージャを使用して透過ファイアウォールモードインターフェイスを設定することはできません。

- 有効にする ASDM：
  - **http server enable**
  - **access-group all global**
  - **http 0.0.0.0 0.0.0.0 management**
- アクセス リスト（Access List）
- Name-Server
- アプライアンスが管理ネットワークで通信することを許可するネットワーク設定。

（注） 第 0 日の構成ファイルをアップロードするか、表示されたテキストボックスにコンテンツをコピーして貼り付けることができます。

ステップ 1 任意のテキストエディタを使用して、新しいテキストファイルを作成します。

ステップ 2 次の例に示すように、テキストファイルに構成の詳細を入力します。

例：

```
ASA Version 9.16.2
!
console serial
interface management0/0
nameif management
security-level 100
ip address 192.168.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/0
```

```
nameif inside
security-level 100
ip address 10.1.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/1
nameif outside
security-level 0
ip address 198.51.100.2 255.255.255.0
no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
```

最初の行は ASA のバージョンで始める必要があります。day0-config は、有効な ASA 構成である必要があります。day0-config を生成する最適な方法は、既存の ASA または ASAv から実行コンフィギュレーションの関連部分をコピーすることです。day0-config 内の行の順序は重要で、既存の **show running-config** コマンド出力の順序と一致している必要があります。

Day0-config 可能な構成：

- ホスト名
- ドメイン名
- Administrative Password
- インターフェイス
- IP アドレス
- スタティック ルート
- DHCP サーバー
- ネットワーク アドレス変換規則

（注） 第 0 日の構成ファイルの内容は、JSON 形式である必要があります。JSON 検証ツールを使用してテキストを検証する必要があります。

ステップ 3 ファイルを `day0-config.txt` として保存します。

ステップ 4 [Custom Script] オプションを選択します。

ステップ 5 `day0-config.txt` ファイルをアップロードするか、表示されたテキストボックスにファイルをコピーして貼り付けます。

ステップ 6 ステップ 1 ～ 3 を繰り返して、展開する ASAv ごとに一意のデフォルト構成ファイルを作成します。

ASAv を Nutanix に展開する

始める前に

展開する FMCv のイメージが [Image Configuration] ページに表示されていることを確認します。

ステップ 1 Nutanix Prism Web コンソールにログインします。

ステップ 2 メインメニューバーで、[View] ドロップダウンリストをクリックし、[VM] を選択します。

ステップ 3 VM ダッシュボードで、[VMの作成（Create VM）] をクリックします。

ステップ 4 次の手順を実行します。

1. ASAv インスタンスの名前を入力します。
2. （オプション）ASAv インスタンスの説明を入力します。
3. ASAv インスタンスで使用するタイムゾーンを選択します。

ステップ 5 コンピューティングの詳細を入力します。

1. ASAv インスタンスに割り当てる仮想 CPU の数を入力します。
2. 各仮想 CPU に割り当てる必要があるコアの数を入力します。
3. ASAv インスタンスに割り当てるメモリの量（GB）を入力します。

ステップ 6 ASAv インスタンスにディスクを接続します。

1. [Disks] で、[Add New Disk] をクリックします。
2. [タイプ（Type）] ドロップダウンリストから、[ディスク（DISK）] を選択します。
3. [操作（Operation）] ドロップダウンリストから、[イメージサービスから複製（Clone from Image Service）] を選択します。
4. [Bus Type] ドロップダウンリストから [SATA] を選択します。
5. [イメージ（Image）] ドロップダウンリストから、使用するイメージを選択します。
6. [追加（Add）] をクリックします。

ステップ 7 3 つ以上の仮想ネットワーク インターフェイスを設定します。

[ネットワークアダプタ（NIC）（Network Adapters (NIC)）] で、[新しいNIC の追加（Add New NIC）] をクリックし、ネットワークを選択して、[追加（Add）] をクリックします。

このプロセスを繰り返して、ネットワーク インターフェイスをさらに追加します。

Nutanix 上の ASAv は、データトラフィック用に 1 つの管理インターフェイスと最大 9 つのネットワーク インターフェイスの合計 10 のインターフェイスをサポートします。ネットワークへのインターフェイスの割り当ては、次の順番であることが必要です。

- vnic0：管理インターフェイス（必須）
- vnic1：外部インターフェイス（必須）
- vnic2：内部インターフェイス（必須）
- vnic3-9：データインターフェイス（オプション）

ステップ 8 ASAv のアフィニティポリシーを設定します。

[VM Host Affinity] で、[Set Affinity] をクリックし、ホストを選択して、[Save] をクリックします。

ノードに障害が発生した場合でも VM を実行できるようにするには、1 つ以上のホストを選択します。

ステップ 9 第 0 日の構成ファイルを準備済みの場合は、次の手順を実行します。

1. [カスタムスクリプト（Custom Script）] を選択します。
2. [Upload A File] をクリックし、第 0 日の構成ファイル（`day0-config.txt`）を選択するか、もしくはコンテンツをコピーしてペーストします。

（注） 他のすべてのカスタム スクリプト オプションは、リリースではサポートされていません。

ステップ 10 [Save] をクリックして、ASAv インスタンスを展開します。VM テーブルビューにインスタンスが表示されます。

ステップ 11 VM テーブルビューで、新しく作成したインスタンスを選択し、[Power On] をクリックします。

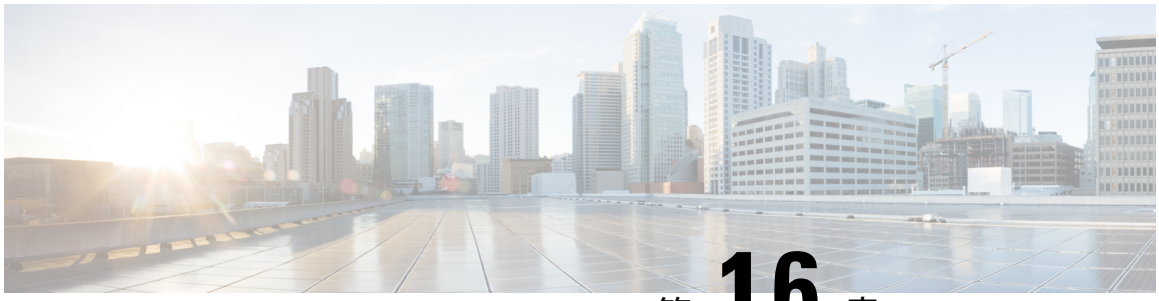
ASAv の起動

VM の電源がオンになったら、`day0-config` ファイルを使用して事前定義されたユーザー名とパスワードで **[ASAv-VM]** > **[Launch Console]**を選択してアクセスします。

（注） 初期設定の完了後の仮想デバイスのこれらの設定を変更するには、CLI を使用します。

ステップ 1 [Launch Console] をクリックして、展開された ASAv にアクセスします。

ステップ 2 **asav ログイン**プロンプトで、`day0-config` **ユーザー名**と**パスワード**を使用してログインします。



第 16 章

Cisco HyperFlex への ASAv の導入

HyperFlex システムは、あらゆる場所であらゆるアプリケーションにハイパーコンバージェンスを提供します。Cisco Unified Computing System（Cisco UCS）テクノロジーを備える HyperFlex は、Cisco Intersight クラウド運用プラットフォームを通じて管理され、場所を問わずアプリケーションとデータを強力にサポートし、コアデータセンターからエッジ、そしてパブリッククラウドまでの運用を最適化し、DevOps 手法を推進して俊敏性を高めることができます。

この章では、Cisco HyperFlex 環境内における Firepower Threat Defense Virtual の機能（機能のサポート、システム要件、ガイドライン、制限事項など）について説明します。

重要 ASAv の最小メモリ要件は 2 GB です。現在の ASAv が 2 GB 未満のメモリで動作している場合、ASAv VM のメモリを増やさずに、以前のバージョンから 9.13(1)+ にアップグレードすることはできません。また、最新バージョンを使用して新しい ASAv VM を再導入することもできます。

Cisco HyperFlex での ASAv のガイドラインと制限事項

ASAv Cisco HyperFlex の複数のインスタンスを作成して、VMware vCenter Server に導入できます。ASAv の導入に使用される特定のハードウェアは、導入されるインスタンスの数と使用要件によって異なります。作成する各仮想アプライアンスには、ホストマシン上での最小リソース割り当て（メモリ、CPU 数、およびディスク容量）が必要です。

重要 ASAv は、8 GB のディスクストレージサイズで展開されます。ディスク容量のリソース割り当てを変更することはできません。

ASAv を展開する前に、次のガイドラインと制限事項を確認します。

推奨される vNIC

最適なパフォーマンスを得るには、vmxnet3 vNIC を使用することを推奨します。この vNIC は、10Gbps の動作をサポートしますが、CPU サイクルも必要な準仮想化ネットワークドライバです。さらに、vmxnet3 を使用する場合は、TCP パフォーマンスの低下を避けるために大量受信オフロード（LRO）を無効にする必要があります。

OVF ファイルのガイドライン

- asav-vi.ovf：vCenter に導入する場合
- ASAv OVF の導入では、ローカリゼーション（非英語モードでのコンポーネントのインストール）はサポートされません。ご自身の環境の VMware vCenter と LDAP サーバーが ASCII 互換モードでインストールされていることを確認してください。
- ASAv をインストールして VM コンソールを使用する前に、キーボードを [United States English] に設定する必要があります。

ハイ アベイラビリティ ガイドラインのためのフェールオーバー

フェールオーバー配置の場合は、スタンバイ装置が同じライセンス権限付与を備えていることを確認してください（両方の装置が 2 Gbps の権限付与であることなど）。

重要 ASAv を使用して高可用性ペアを作成する場合は、データインターフェイスを各 ASAv に同じ順序で追加する必要があります。完全に同じインターフェイスを異なる順序で各 ASAv に追加すると、ASAv コンソールにエラーが表示される場合があります。また、フェールオーバー機能にも影響が出ることがあります。

IPv6 のガイドライン

VMware vSphere Web Client を使用して ASAv OVF ファイルを最初に導入する場合は、管理インターフェイスに IPv6 アドレスを指定できません。ASDM または CLI を使用して、IPv6 アドレッシングを後で追加できます。

vMotion に関するガイドライン

- VMware では、vMotion を使用する場合は共有ストレージのみを使用する必要があります。ASAv の導入時に、ホストクラスタがある場合は、ストレージをローカルに（特定のホスト上）、または共有ホスト上でプロビジョニングできます。ただし、ASAv を vMotion を使用して別のホストに移行する場合、ローカルストレージを使用するとエラーが発生します。

スループット用のメモリと vCPU の割り当てとライセンス

- ASAv に割り当てられたメモリのサイズは、スループットレベルに合わせたものです。異なるスループットレベルのライセンスを要求する場合を除いて、[設定の編集（Edit Settings）] ダイアログボックスのメモリ設定または vCPU ハードウェアの設定は変更しないでください。アンダープロビジョニングは、パフォーマンスに影響を与える可能性があります。

（注） メモリまたはvCPUハードウェア設定を変更する必要がある場合は、ASAv のライセンス （1 ページ）に記載されている値のみを使用してください。VMware が推奨するメモリ構成の最小値、デフォルト値、および最大値は使用しないでください。

CPU 予約

- デフォルトで、ASAv の CPU 予約は 1000 MHz です。共有、予約、および制限の設定を使用することで、ASAv に割り当てられた CPU リソースの量を変更できます。**[設定の編集（Edit Settings）] > [リソース（Resources）] > [CPU]**。より低い設定で必要なトラフィック負荷が課されている状況で ASAv が目的を達成できる場合は、CPU 予約の設定を 1000 Mhz 未満にできます。ASAv によって使用される CPU の量は、それが動作しているハードウェアプラットフォームだけでなく、それが行っている作業のタイプと量によっても異なります。

  仮想マシンの [Performance] タブの [Home] ビューに配置された [CPU Usage (MHz)] チャートから、すべての仮想マシンに関する CPU 使用率をホストの視点で確認できます。ASAv が標準的なトラフィック量を処理しているときの CPU 使用率のベンチマークを設定すれば、その情報を CPU 予約の調整時の入力として使用できます。

  詳細については、CPU Performance Enhancement Advice のリンクを参照してください。
- ASDM で ASAv**show vm > show cpu** コマンドを使用して、リソース割り当て、およびオーバープロビジョニングまたはアンダープロビジョニングされたリソースを

  表示できます。

  [ホーム（Home）] > [デバイスダッシュボード（Device Dashboard）] > [デバイス情報（Device Information）] > [仮想リソース（Virtual Resources）] タブ

  または

  [モニタリング（Monitoring）] > [プロパティ（Properties）] > [システムリソースグラフ（System Resources Graphs）] > [CPU] ペイン

UCS B および C シリーズ ハードウェアにおけるトランスペアレントモードに関するガイドライン

MAC フラップが、Cisco UCS B（コンピューティングノード）および C（コンバージドノード）シリーズハードウェアのトランスペアレントモードで動作する一部の ASAv 設定で発生することがあります。MAC アドレスがさまざまな場所で出現した場合、パケットはドロップされます。

VMware 環境にトランスペアレントモードで ASAv を導入する場合に MAC フラップを回避するには、次のガイドラインを参考にしてください。

- VMware NIC チーミング：UCS B または C シリーズにトランスペアレントモードで ASAv を導入する場合、内部および外部インターフェイスに使用するポートグループにはアクティブアップリンクを 1 つだけ設定し、アップリンクは同じである必要があります。vCenter で VMware NIC チーミングを設定します。
- ARP インスペクション：ASAv で ARP インスペクションを有効にし、受信インターフェイスで MAC および ARP エントリを静的に設定します。ARP インスペクションと有効化の詳細については、Cisco ASA シリーズ コンフィギュレーション ガイド（一般的な操作）[英語] を参照してください。

ASAv および HyperFlex のシステム要件

HyperFlex HX シリーズの設定とクラスタ

設定	クラスタ
HX220c コンバージドノード	• フラッシュクラスタ • 最小 3 ノードクラスタ（データベース、VDI、VSI）
HX240c コンバージドノード	• フラッシュクラスタ • 最小 3 ノードクラスタ（VSI：IT/Biz アプリケーション、テスト/開発）
HX220C とエッジ（VDI、VSI、ROBO） HX240C（VDI、VSI、テスト/開発）	• ハイブリッドクラスタ • 最小 3 ノードクラスタ
B200 + C240/C220	コンピューティング バウンド アプリ/VDI

HyperFlex HX シリーズの導入オプション：

- ハイブリッドクラスタ
- フラッシュクラスタ

- HyperFlex HX エッジ
- SED ドライブ
- NVME キャッシュ
- GPU

HyperFlex HX クラウドを利用した管理オプションについては、『Cisco HyperFlex システム設置ガイド』の「*HyperFlex* ファブリック インターコネクトに接続されたクラスタの展開」のセクションを参照してください。

HyperFlex コンポーネントとバージョン

コンポーネント	バージョン
VMware vSphere	7.0.2-18426014
HyperFlex Data Platform	4.5.2a-39429

サポートされる機能

- 展開モード：ルーテッド（スタンドアロン）、ルーテッド（HA）、およびトランスペアレント
- ASAv のネイティブ HA
- ジャンボフレーム
- VirtIO
- HyperFlex データセンタークラスタ（ストレッチ クラスタを除く）
- HyperFlex Edge クラスタ
- HyperFlex すべての NVMe、オールフラッシュ、およびハイブリッド コンバージド ノード
- HyperFlex コンピューティング専用ノード

サポートされない機能

SR-IOV を使用した ASAv の実行は、HyperFlex で認定されていません。

（注） HyperFlex は SR-IOV をサポートしていますが、MLOM VIC に加えて PCI-e NIC も必要です。

Cisco HyperFlex への ASAv の導入方法

ステップ	タスク	詳細情報
1	ガイドラインと制限事項を確認します。	Cisco HyperFlex での ASAv のガイドラインと制限事項 （297 ページ）
2	前提条件を確認します。	ASAv および Cisco HyperFlex の前提条件 （302 ページ）
3	Cisco.com から OVF ファイルをダウンロードします。	ASAv ソフトウェアのダウンロードと解凍 （303 ページ）
4	Cisco HyperFlex に ASAv を導入します。	vSphere vCenter への Cisco HyperFlex 上の ASAv の導入 （303 ページ）
5	ASAv コンソールにアクセスします。	ASAv コンソールへのアクセス （306 ページ）

ASAv および Cisco HyperFlex の前提条件

VMware vSphere Web Client、vSphere スタンドアロンクライアント、または OVF ツールを使用して Cisco HyperFlex に ASAv を導入できます。システム要件については、『Cisco ASA Compatibility』を参照してください。

vSphere 標準スイッチのセキュリティ ポリシー

vSphere スイッチについては、レイヤ 2 セキュリティ ポリシーを編集して、ASAv インターフェイスによって使用されるポート グループに対してセキュリティ ポリシーの例外を適用できます。次のデフォルト設定を参照してください。

- 無差別モード：**拒否**
- MAC アドレスの変更：**許可**
- 不正送信：**許可**

次の ASAv 設定については、これらの設定の変更が必要な場合があります。詳細については、vSphere のマニュアルを参照してください。

表 *28* : ポート グループのセキュリティ ポリシーの例外

セキュリティの例外	ルーテッド ファイアウォール モード		トランスペアレント ファイアウォール モード	
	フェールオーバーなし	フェールオーバー	フェールオーバーなし	フェールオーバー
無差別モード	<任意>	<任意>	承認	承認
MAC アドレスの変更	<任意>	承認	<任意>	承認
不正送信	<任意>	承認	承認	承認

ASAv ソフトウェアのダウンロードと解凍

はじめる前に

ASAv を導入する前に、vSphere（管理用）に少なくとも 1 つのネットワークを設定しておく必要があります。

ステップ 1 ZIP ファイルを Cisco.com からダウンロードし、ローカル ディスクに保存します。

https://www.cisco.com/go/asa-software

（注） Cisco.com のログインおよびシスコ サービス契約が必要です。

ステップ 2 ファイルを作業ディレクトリに解凍します。ディレクトリからファイルを削除しないでください。次のファイルが含まれています。

- asav-vi.ovf：vCenter への導入用。
- boot.vmdk：ブート ディスク イメージ。
- disk0.vmdk：ASAv のディスク イメージ。
- day0.iso：day0-config ファイルおよびオプションの idtoken ファイルを含む ISO。
- asav-vi.mf：vCenter への導入用のマニフェスト ファイル。

vSphere vCenter への Cisco HyperFlex 上の ASAv の導入

この手順を使用して、HyperFlex から VMware vSphere vCenter に ASAv を導入します。vSphere Web Client（または vSphere Client）を使用して、仮想マシンを導入して設定できます。

始める前に

HyperFlex に ASAv を導入する前に、vSphere（管理用）に少なくとも 1 つのネットワークを設定しておく必要があります。

ASAv を HyperFlex クラスタにインストールする前に、HyperFlex クラスタと共有データストアを作成する必要があります。詳細については、HyperFlex コンフィギュレーション ガイド [英語] を参照してください。

ステップ 1 vSphere Web クライアントにログインします。

ステップ 2 vSphere Web Client（または vSphere Client）を使用し、**[アクション（ACTIONS）]** > **[OVFテンプレートの導入（Deploy OVF Template）]** をクリックして、以前ダウンロードした OVF テンプレートファイルを導入します。

[Deploy OVF Template] ウィザードが表示されます。

ステップ 3 ファイルシステムで OVF テンプレートソースの場所を参照し、[次へ（NEXT）] をクリックします。

ステップ 4 [OVFテンプレートの詳細（OVF Template Details）] ページを確認し、OVF テンプレートの情報（製品名、バージョン、ベンダー、ダウンロードサイズ、ディスク上のサイズ、説明）を確認して、[次へ（NEXT）] をクリックします。

ステップ 5 [エンドユーザーライセンス契約書（End User License Agreement）] ページが表示されます。OVF テンプレート（VI テンプレートのみ）でパッケージ化されたライセンス契約書を確認し、[承認（Accept）] をクリックしてライセンスの条件に同意し、[次へ（NEXT）] をクリックします。

ステップ 6 [名前と場所（Name and Location）] ページで、この導入の名前を入力し、HyperFlex を導入するインベントリ内の場所（共有データストアまたはクラスタ）を選択して、[次へ（NEXT）]をクリックします。名前はインベントリフォルダ内で一意である必要があり、最大 80 文字を使用できます。

VSphere Web Client では、インベントリビューに管理対象オブジェクトの組織階層が表示されます。インベントリは、vCenter Server またはホストが管理対象オブジェクトを整理する目的で使用する階層構造です。この階層には、vCenter Server にあるすべての監視対象オブジェクトが含まれています。

ステップ 7 ASAv HyperFlex を実行するリソースプールに移動して選択し、[次へ（NEXT）] をクリックします。

（注） このページは、クラスタにリソースプールが含まれている場合にのみ表示されます。コンピューティングリソースプールの場合、最高のパフォーマンスを得るためにはクラスタのみを推奨します

ステップ 8 [導入設定（Deployment Configuration）] を選択します。[設定（Configuration）] ドロップダウンリストから、サポートされている 3 つの vCPU/メモリ値のいずれかを選択し、[次へ（NEXT）] をクリックします。

ステップ 9 仮想マシンファイルを保存する [ストレージ（Storage）] の場所を選択し、[次へ（NEXT）] をクリックします。

このページで、宛先クラスタですでに構成されているデータストア（HX 接続を使用して作成された HX クラスタ共有データストア）を選択します。仮想マシン コンフィギュレーション ファイルおよび仮想ディスク ファイルが、このデータストアに保存されます。仮想マシンとそのすべての仮想ディスクファイルを保存できる十分なサイズのデータストアを選択してください。

ステップ 10 [ネットワークマッピング（Network Mapping）]ページで、OVF テンプレートで指定されたネットワークをインベントリ内のネットワークにマッピングし、[次へ（NEXT）] をクリックします。

Management 0-0 インターフェイスが、インターネットから到達可能な VM ネットワークと関連付けられていることを確認します。非管理インターフェイスは、管理モードに応じて ASAv Mangement Centre または ASAv Device Manager から設定できます。

重要 HyperFlex 上の ASAv では、仮想デバイスを作成するときのデフォルトが vmxnet3 インターフェイスになりました。以前は、デフォルトはe1000でした。e1000インターフェイスを使用している場合は、切り替えることを**強く推奨します**。Vmxnet3 のデバイスドライバとネットワーク処理はHyperFlex と統合されているため、使用するリソースが少なくなり、ネットワークパフォーマンスが向上します。

ネットワークはアルファベット順になっていない可能性があります。ネットワークを見つけることが非常に困難な場合は、後で [設定の編集（Edit Settings）] ダイアログボックスからネットワークを変更できます。導入後、インスタンスを右クリックして [設定の編集（Edit Settings）] を選択します。ただし、[ネットワークマッピング（Network Mapping）] ページには ID は表示されません（ネットワークアダプタ ID のみ）。

以下に示す、インターフェイス（vmxnet3 のデフォルトインターフェイス）のネットワークアダプタ、送信元ネットワーク、宛先ネットワークの対応を参照してください。

表 *29 :* 送信元から宛先ネットワークへのマッピング：*vmxnet3*

ネットワーク アダプタ ID	ASAv インターフェイス ID
ネットワーク アダプタ 1	Management 0/0
ネットワーク アダプタ 2	GigabitEthernet 0/0
ネットワーク アダプタ 3	GigabitEthernet 0/1
ネットワーク アダプタ 4	GigabitEthernet 0/2
ネットワーク アダプタ 5	GigabitEthernet 0/3
ネットワーク アダプタ 6	GigabitEthernet 0/4
ネットワーク アダプタ 7	GigabitEthernet 0/5
ネットワーク アダプタ 8	GigabitEthernet 0/6
ネットワーク アダプタ 9	GigabitEthernet 0/7
ネットワーク アダプタ 10	GigabitEthernet 0/8

ASAv を導入する際には、合計 10 個のインターフェイスを指定できます。データインターフェイスについて、送信元ネットワークが正しい宛先ネットワークにマッピングされ、各データインターフェイスが一意のサブネットまたは VLAN にマッピングされていることを確認します。すべてのインターフェイスを使用する必要はなく、使用する予定がないインターフェイスは、設定内で無効のままにできます。

ステップ 11 [プロパティ（Properties）] ページで、OVF テンプレート（VI テンプレートのみ）でパッケージ化された、ユーザー設定可能なプロパティを設定します。

- [パスワード（Password）]：管理アクセス用のパスワードを設定します。
- [ネットワーク（Network）]：完全修飾ドメイン名（FQDN）、DNS、検索ドメイン、ネットワークプロトコル（IPv4 または IPv6）などのネットワーク情報を設定します。
- [管理インターフェイス（Management Interface）]：管理構成を設定し、ドロップダウンをクリックして [DHCP/手動（DHCP/Manual）] を選択し、管理インターフェイスの IP 構成を設定します。
- [ファイアウォールモード（Firewall Mode）]：初期ファイアウォールモードを設定します。[ファイアウォールモード（Firewall Mode）] のドロップダウン矢印をクリックし、サポートされている 2 つのモードのいずれか（[ルーテッド（Routed）] または [トランスペアレント（Transparent）]）を選択します。

ステップ 12 [次へ（NEXT）] をクリックします。[準備完了（Ready To Complete）] セクションで、表示された情報を確認します。これらの設定を使用して展開を開始するには、[終了（Finish）] をクリックします。変更を加えるには、[戻る（Back）] をクリックして前のダイアログボックスに戻ります。

（任意）[導入後に電源をオン（Power on after deployment）] オプションにチェックマークを付けて、VM の電源をオンにし、[終了（Finish）] をクリックします。

ウィザードが完了すると、vSphere Web Client によって仮想マシンが処理されます。[グローバル情報（Global Information）] 領域の [最近のタスク（Recent Tasks）] ペインで [OVF展開の初期設定（Initialize OVF deployment）] ステータスを確認できます。

この手順が終了すると、[OVFテンプレートの導入（Deploy OVF Template）] 完了ステータスが表示されます。

ASAv インスタンスがインベントリ内の指定されたデータセンターの下に表示されます。新しい VM の起動には、最大 30 分かかることがあります。

（注） Cisco Licensing Authority に ASAv HyperFlex を正常に登録するには、インターネットアクセスが必要です。インターネットに接続してライセンス登録を完了させるには、導入後に追加の設定が必要になることがあります。

ASAv コンソールへのアクセス

ASDM を使用する場合、トラブルシューティングに CLI を使用する必要がある場合があります。デフォルトでは、組み込みの VMware vSphere コンソールにアクセスできます。または、コピーアンドペーストなどのより優れた機能を持つネットワーク シリアル コンソールを設定できます。

- VMware vSphere コンソールの使用
- ネットワーク シリアル コンソール ポートの設定

VMware vSphere コンソールの使用

初期設定またはトラブルシューティングを行うには、VMware vSphere Web Client により提供される仮想コンソールから CLI にアクセスします。後で Telnet または SSH の CLI リモート アクセスを設定できます。

始める前に

vSphere Web Client では、ASAv コンソール アクセスに必要なクライアント統合プラグインをインストールします。

ステップ 1 VMware vSphere Web Client で、インベントリの ASAv インスタンスを右クリックし、[Open Console] を選択します。または、[Summary] タブの [Launch Console] をクリックします。

ステップ 2 コンソールでクリックして Enter を押します。注：Ctrl + Alt を押すと、カーソルが解放されます。

ASAv がまだ起動中の場合は、起動メッセージが表示されます。

ASAv が初めて起動すると、OVF ファイルから提供されたパラメータを読み込み、それらを ASAv システム構成に追加します。その後、起動プロセスが自動的に再開され、稼働を開始します。この二重起動プロセスは、初めて ASAv を導入した場合にのみ発生します。

（注） ライセンスをインストールするまで、スループットは 100 Kbps に制限されるため、予備接続テストを実行できます。ライセンスは、通常の操作に必要です。ライセンスをインストールするまで、次のメッセージがコンソールで繰り返し表示されます。

```
Warning: ASAv platform license state is Unlicensed.
Install ASAv platform license for full functionality.
```

次のプロンプトが表示されます。

```
ciscoasa>
```

このプロンプトは、ユーザー EXEC モードで作業していることを示します。ユーザー EXEC モードでは、基本コマンドのみを使用できます。

ステップ 3 特権 EXEC モードにアクセスします。

例：

```
ciscoasa> enable
```

次のプロンプトが表示されます。

```
Password:
```

ステップ 4 Enterキーを押して、次に進みます。デフォルトでは、パスワードは空白です。以前にイネーブル パスワードを設定した場合は、Enter を押す代わりにこれを入力します。

プロンプトが次のように変化します。

```
ciscoasa#
```

設定以外のすべてのコマンドは、特権 EXEC モードで使用できます。特権 EXEC モードからコンフィギュレーション モードに入ることもできます。

特権モードを終了するには、**disable** コマンド、**exit** コマンド、または **quit** コマンドを入力します。

ステップ 5 グローバル コンフィギュレーション モードにアクセスします。

```
ciscoasa# configure terminal
```

プロンプトが次のように変化します。

```
ciscoasa(config)#
```

グローバル コンフィギュレーション モードから ASAv の設定を開始できます。グローバル コンフィギュレーション モードを終了するには、**exit** コマンド、**quit** コマンド、または **end** コマンドを入力します。

ネットワーク シリアル コンソール ポートの設定

コンソール エクスペリエンスの向上のために、コンソール アクセスについて、ネットワーク シリアル ポートを単独で設定するか、または仮想シリアル ポート コンセントレータ（vSPC）に接続するように設定できます。各方法の詳細については、VMware vSphere のマニュアルを参照してください。ASAv では、仮想コンソールの代わりにシリアル ポートにコンソール出力を送信する必要があります。この手順では、シリアル ポート コンソールを有効にする方法について説明します。

ステップ 1 VMware vSphere でネットワーク シリアル ポートを設定します。VMware vSphere のマニュアルを参照してください。

ステップ 2 ASAv で、「use_ttyS0」という名前のファイルを disk0 のルート ディレクトリに作成します。このファイルには内容が含まれている必要はありません。この場所に存在することのみが必要です。

disk0:/use_ttyS0

- ASDM から **[ツール（Tools）]** > **[ファイル管理（File Management）]** ダイアログボックスを使用して、この名前で空のテキストファイルをアップロードできます。
- vSphere コンソールで、ファイル システム内の既存のファイル（任意のファイル）を新しい名前にコピーできます。次に例を示します。

```
ciscoasa(config)# cd coredumpinfo
ciscoasa(config)# copy coredump.cfg disk0:/use_ttyS0
```

ステップ 3 ASAvをリロードします。

- ASDM から **[Tools]** > **[System Reload]** を選択します。
- vSphere コンソールで **reload** を入力します。

  ASAv は vSphere コンソールへの送信を停止し、代わりにシリアル コンソールに送信します。

ステップ 4 シリアル ポートの追加時に指定した vSphere のホスト IP アドレスとポート番号に Telnet 接続するか、または vSPC の IP アドレスとポートに Telnet 接続します。

vCPU またはスループット ライセンスのアップグレード

ASAv は、使用できる vCPU の数に影響するスループット ライセンスを使用します。

ASAv の vCPU の数を増やす（または減らす）場合は、新しいライセンスを要求して、その新しいライセンスを適用し、新しい値と一致するように VMware の VM プロパティを変更します。

（注） 割り当てられた vCPU は、ASAv 仮想 CPU ライセンスまたはスループットライセンスと一致している必要があります。RAM は、vCPU 用に正しくサイズ調整されている必要があります。アップグレードまたはダウングレード時には、この手順に従って、ライセンスと vCPU を迅速に調整するようにします。永続的な不一致がある場合、ASAv は適切に動作しません。

ステップ 1 新しい ASAv 仮想 CPU ライセンスまたはスループットライセンスを要求します。

ステップ 2 新しいライセンスを適用します。フェールオーバー ペアの場合、両方の装置に新しいライセンスを適用します。

ステップ 3 フェールオーバーを使用するかどうかに応じて、次のいずれかを実行します。

- フェールオーバーあり：vSphere Web Client で、スタンバイ ASAv の電源を切断します。たとえば、ASAv をクリックしてから [Power Off the virtual machine] をクリックするか、または ASAv を右クリックして [Shut Down Guest OS] を選択します。
- フェールオーバーなし：vSphere Web Client で、ASAv の電源を切断します。たとえば、ASAv をクリックしてから [Power Off the virtual machine] をクリックするか、または ASAv を右クリックして [Shut Down Guest OS] を選択します。

ステップ 4 ASAv をクリックしてから [仮想マシンの設定の編集（Edit Virtual machine settings）] をクリックします（または ASAv を右クリックして [設定の編集（Edit Settings）] を選択します）。

[Edit Settings] ダイアログボックスが表示されます。

ステップ 5 新しい vCPU ライセンスの正しい値を確認するには、ASAv のライセンス （1 ページ）にある CPU 要件とメモリ要件を参照してください。

ステップ 6 [Virtual Hardware] タブの [CPU] で、ドロップダウン リストから新しい値を選択します。

ステップ 7 [Memory] には、新しい RAM の値を入力します。

ステップ 8 [OK] をクリックします。

ステップ 9 ASAv の電源をオンにします。たとえば、[Power On the Virtual Machine] をクリックします。

ステップ 10 フェールオーバー ペアの場合：

1. アクティブ装置へのコンソールを開くか、またはアクティブ装置で ASDM を起動します。
2. スタンバイ装置の起動が終了した後、スタンバイ装置にフェールオーバーします。

- ASDM：**[Monitoring]** > **[Properties]** > **[Failover]** > **[Status]** を選択し、[Make Standby] をクリックします。
- CLI： **failover active**

3. アクティブ装置に対して、ステップ 3 ～ 9 を繰り返します。

次のタスク

詳細については、ASAv のライセンス （1 ページ）を参照してください。

Cisco HyperFlex での ASAv のパフォーマンス調整

ASAv は高性能のアプライアンスですが、最適な結果を得るには Cisco HyperFlex の調整が必要な場合があります。

以下は、HyperFlex 環境で ASAv の最高のパフォーマンスを促進するためのベストプラクティスと推奨事項です。

ジャンボ フレームの有効化

MTU が大きいほど、大きいパケットを送信できます。パケットが大きいほど、ネットワークの効率が良くなる可能性があります。次のガイドラインを参照してください。

- トラフィック パスの MTU の一致：すべての ASAv インターフェイスとトラフィック パス内のその他のデバイスのインターフェイスでは、MTU が同じになるように設定することを推奨します。MTU の一致により、中間デバイスでのパケットのフラグメント化が回避できます。
- ジャンボ フレームへの対応：MTU を最大 9198 バイトに設定できます。ASAv の最大値は 9000 です。

この手順では、次の環境でジャンボフレームを有効にする方法について説明します。

vSphere 7.0.1 上の HyperFlex クラスタ > VMware vSphere vSwitch> Cisco UCS ファブリック インターコネクト（FI）

ステップ 1 ASAv を展開した ASAv ホストの MTU 設定を変更します。

1. vSphere Web クライアントを使用して vCenter サーバーに接続します。
2. HyperFlex ホストの [詳細システム設定（Advanced System Settings）] で、[Net.Vmxnet3NonTsoPacketGtMtuAllowed] の設定パラメータの値を `1` にします。
3. 変更を保存してホストを再起動します。

詳細については、「https://kb.vmware.com/s/article/1038578」を参照してください。

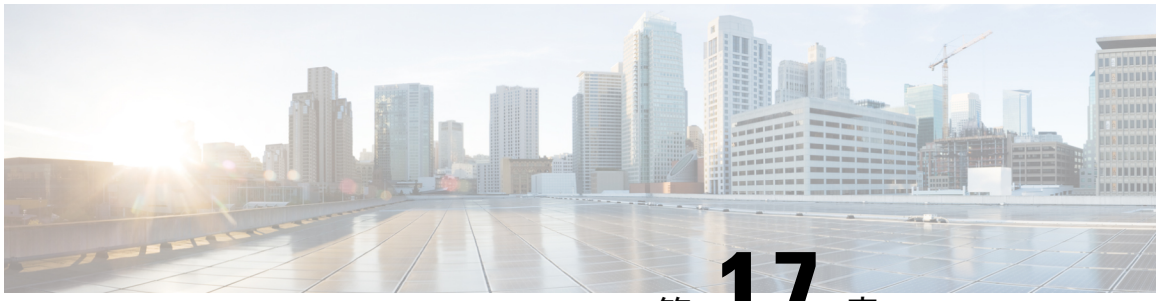
ステップ 2 VMware vSphere vSwitch の MTU 設定を変更します。

1. vSphere Web クライアントを使用して vCenter サーバーに接続します。
2. VMware vSphere vSwitch のプロパティを編集し、[MTU] の値を `9000` に設定します。

ステップ 3 Cisco UCS ファブリック インターコネクト（FI）の MTU 設定を変更します。

1. Cisco UCS Management コンソールにログインします。
2. QoS システムクラスを編集するには、**[LAN]** > **[LANクラウド（LAN Cloud）]** > **QoS システム クラス（QoS System Class）** の順に選択します。[全般（General）] タブで、[MTU] の値を 9216 に設定します。
3. vNIC を編集するには、**[LAN]** > **[ポリシー（Policies）]** > **[ルート（root）]** > **[サブ組織（Sub-Organizations）]**

   <your-hyperflex-org>**vNIC テンプレート** <your-vnic> の順に選択します。[全般（General）] タブで、[MTU] の値を 9000 に設定します。



第 17 章

ASAv の設定

ASAv の導入では、ASDM アクセスを事前設定します。導入時に指定したクライアント IP アドレスから、Web ブラウザで ASAv 管理 IP アドレスに接続できます。この章では、他のクライアントが ASDM にアクセスできるようにする方法と CLI アクセスを許可する方法（SSH または Telnet）についても説明します。この章で取り上げるその他の必須の設定作業には、ASDM でウィザードが提供するライセンスのインストールおよび一般的な設定作業が含まれます。

ASDM の起動

ステップ 1 ASDM クライアントとして指定した PC で次の URL を入力します。

https://*asa_ip_address*/**admin**

次のボタンを持つ ASDM 起動ウィンドウが表示されます。

- **Install ASDM Launcher and Run ASDM**
- **Run ASDM**
- **Run Startup Wizard**

ステップ 2 ランチャをダウンロードするには、次の手順を実行します。

a) [Install ASDM Launcher and Run ASDM] をクリックします。

b) ユーザー名とパスワードのフィールドを空のままにし（新規インストールの場合）、[OK] をクリックします。HTTPS 認証が設定されていない場合は、ユーザー名および**イネーブル**パスワード（デフォルトで空白）を入力しないで ASDM にアクセスできます。HTTPS 認証を有効にした場合、ユーザー名と関連付けられたパスワードを入力します。

c) インストーラを PC に保存して、インストーラを起動します。インストールが完了すると、ASDM-IDM ランチャが自動的に開きます。

d) 管理 IP アドレスを入力し、ユーザー名とパスワードを空白のままにし（新規インストールの場合）、[OK] をクリックします。HTTPS 認証を有効にした場合、ユーザー名と関連付けられたパスワードを入力します。

ステップ 3 Java Web Start を使用するには、次の手順を実行します。

a) [Run ASDM] または [Run Startup Wizard] をクリックします。
b) プロンプトが表示されたら、ショートカットをコンピュータに保存します。オプションで、アプリケーションを保存せずに開くこともできます。
c) ショートカットから Java Web Start を起動します。
d) 表示されたダイアログボックスに従って、任意の証明書を受け入れます。Cisco ASDM-IDM Launcher が表示されます。
e) ユーザー名とパスワードを空白のままにし（新規インストールの場合）、[OK] をクリックします。HTTPS 認証を有効にした場合、ユーザー名と関連付けられたパスワードを入力します。

ASDM を使用した初期設定の実行

次の ASDM ウィザードおよび手順を使用して初期設定を行うことができます。

- Startup Wizard の実行
- （任意）ASAv の内側にあるパブリックサーバーへのアクセス許可
- （オプション）VPN ウィザードの実行
- （オプション）ASDM の他のウィザードの実行

CLI の設定については、Cisco ASA シリーズ CLI コンフィギュレーション ガイド [英語] を参照してください。

Startup Wizard の実行

セキュリティ ポリシーをカスタマイズして導入方法に最適化するには、[Startup Wizard] を実行します。

ステップ 1 **[Wizards]** > **[Startup Wizard]** を選択します。

ステップ 2 セキュリティ ポリシーをカスタマイズして、導入方法に最適化します。次を設定できます。

- ホスト名
- ドメイン名
- 管理パスワード
- インターフェイス

- IP アドレス
- スタティック ルート
- DHCP サーバー
- ネットワーク アドレス変換規則
- その他の項目

（任意）ASAv の内側にあるパブリックサーバーへのアクセス許可

[設定（Configuration）] > [ファイアウォール（Firewall）] > [パブリックサーバー（Public Servers）] ペインで、セキュリティポリシーが自動的に設定され、インターネットから内部サーバーにアクセスできるようになります。ビジネス オーナーとして、内部ネットワーク サービス（Web サーバーや FTP サーバーなど）に外部ユーザーがアクセスできるようにする必要がある場合があります。これらのサービスは、ASAvの背後にある、Demilitarized Zone（DMZ; 非武装地帯）と呼ばれる別のネットワーク上に配置できます。DMZ にパブリック サーバーを配置すると、パブリック サーバーに対する攻撃は内部ネットワークには影響しません。

（オプション）VPN ウィザードの実行

次のウィザード（**[Wizards] > [VPN Wizards]**）を使用して、VPN を設定できます。

- サイト間 VPN ウィザード：ASAv と別の VPN 対応デバイス間で IPsec サイト間トンネルを作成します。
- AnyConnect VPN ウィザード：Cisco AnyConnect VPN Client の SSL VPN リモートアクセスを設定します。AnyConnect クライアント では ASA へのセキュアな SSL 接続が提供されるため、リモートユーザーによる企業リソースへのフル VPN トンネリングが可能になります。ASA ポリシーを設定すると、リモートユーザーが最初にブラウザを使用して接続するときに、AnyConnect クライアントをダウンロードできます。AnyConnect クライアント 3.0 以降を使用する場合、クライアントは、SSL または IPsec IKEv2 VPN プロトコルを実行できます。
- Clientless SSL VPN Wizard：ブラウザにクライアントレス SSL VPN リモート アクセスを設定します。クライアントレス ブラウザベース SSL VPN によって、ユーザーは Web ブラウザを使用して ASA へのセキュアなリモートアクセス VPN トンネルを確立できます。認証されると、ユーザーにはポータルページが表示され、サポートされる特定の内部リソースにアクセスできるようになります。ネットワーク管理者は、グループ単位でユーザーにリソースへのアクセス権限を付与します。ACL は、特定の企業リソースへのアクセスを制限したり、許可するために適用できます。
- IPsec (IKEv1 or IKEv2) Remote Access VPN Wizard：Cisco IPsec クライアント用の IPsec VPN リモート アクセスを設定します。

Azure への ASAv IPsec 仮想トンネルインターフェイス（VTI）接続の構成方法については、『Azure への ASA IPsec VTI 接続の構成』を参照してください。

（オプション）ASDM の他のウィザードの実行

高可用性を備えたフェールオーバー、VPN クラスタ ロード バランシング、およびパケット キャプチャを設定するには、ASDM でその他のウィザードを実行します。

- High Availability and Scalability Wizard：フェールオーバーまたは VPN ロード バランシングを設定します。
- Packet Capture Wizard：パケット キャプチャを設定し、実行します。このウィザードは、入出力インターフェイスのそれぞれでパケット キャプチャを 1 回実行します。パケットをキャプチャすると、PC にパケット キャプチャを保存し、パケット アナライザでチェックおよびリプレイできます。

詳細設定

ASAv の設定を続行するには、Cisco ASA シリーズ ドキュント一覧 [英語] を参照してください。

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。