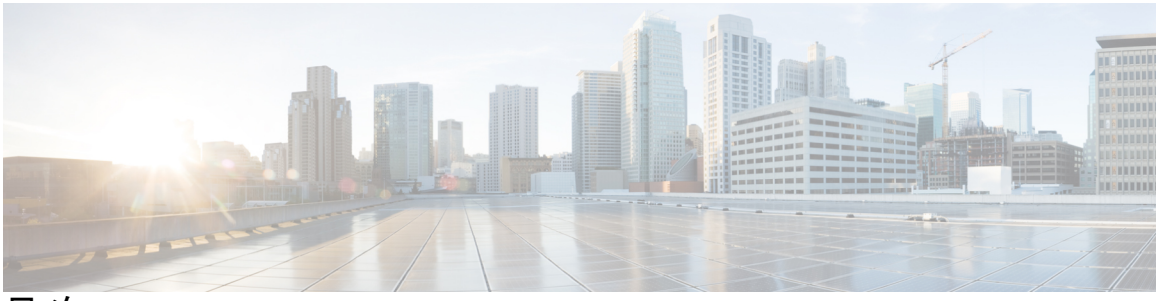
Amazon Web Services での Cisco Catalyst 8000V エッジソフトウェアの展開

初版：2021 年 1 月 30 日

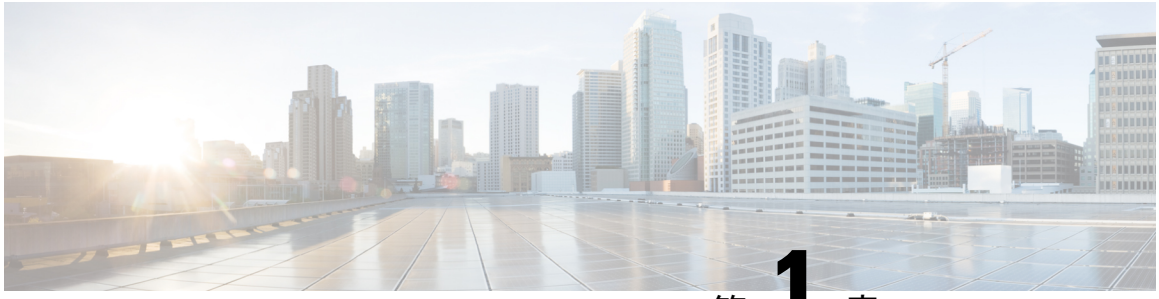
最終更新：2022 年 12 月 17 日

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/



目 次

【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。



第 1 章

はじめに

ここでは、このマニュアルの対象読者、構成、および表記法について説明します。また、他のマニュアルの入手方法についても説明します。

この前書きは、次の項で構成されています。

対象読者および適用範囲

このドキュメントは、Cisco Enterprise ルータの設定担当者を対象としています。このドキュメントの対象者は、主に次のとおりです。

- ネットワーキングに関する技術的な背景知識と経験を持つお客様。
- ルータベースのインターネットワーキングに関する基本的な知識に精通しているが、Cisco IOS ソフトウェアについては経験の浅いシステム管理者。
- インターネットワーキング装置のインストールと設定を担当しているシステム管理者、および Cisco IOS ソフトウェアに精通しているシステム管理者。

機能の互換性

コンフィギュレーション ガイドで説明されているデバイスで使用可能な機能などの Cisco IOS XE ソフトウェアの詳細については、それぞれのルータのドキュメントセットを参照してください。

特定の機能のサポートを確認するには、Cisco Feature Navigator ツールを使用します。これは、特定のソフトウェアリリース、フィーチャセット、またはプラットフォームをサポートする Cisco IOS XE のソフトウェアイメージを判別できるツールです。

表記法

このマニュアルでは、次の表記法を使用しています。

表記法	説明
^ または **Ctrl**	^ および **Ctrl** シンボルは、Ctrl キーを表します。たとえば、**^D** または **Ctrl+D** というキーの組み合わせは、**Ctrl** キーを押しながら **D** キーを押すことを意味します。キーは大文字で表記されていますが、大文字と小文字の区別はありません。
ストリング	ストリングは、イタリックで示される引用符を付けない一組の文字です。たとえば、SNMP コミュニティ ストリングとして public を設定する場合、ストリングの前後には引用符を使用しません。引用符を使用すると、その引用符も含めてストリングとみなされます。

コマンドシンタックスの説明には、次の表記法を使用しています。

表記法	説明
ボールド	ユーザが入力するコマンドおよびキーワードを示します。
イタリック体	イタリック体の文字は、ユーザが値を指定する引数です。
[x]	省略可能な要素（キーワードまたは引数）は、角かっこで囲んで示しています。
|	縦棒で区切られている場合、複数の任意または必須のキーワードまたは引数から、1 つを選択します。
[x | y]	角カッコで囲まれ、縦棒で区切られたキーワードまたは引数は、任意の選択肢です。
{x | y}	波カッコで囲まれ、縦棒で区切られたキーワードまたは引数は、必須の選択肢です。

省略可能または必須の要素内に、さらに省略可能または必須の選択肢を含める場合は、角カッコや波カッコを入れ子にして示しています。たとえば、次の表を参照してください。

表記法	説明
[x {y | z}]	角カッコ内の波カッコおよび縦棒は、任意の要素内の必須の選択肢です。

例では、次の表記法を使用しています。

表記法	説明
`screen`	画面に表示される情報の例は、Courier フォントで表します。
`bold screen`	ユーザの入力が必要なテキストの例は、太字の Courier フォントで表します。
< >	山カッコで囲まれたテキストは、パスワードなど、画面に出力されないテキストを表します。
!	行の先頭にある感嘆符（!）は、コメント行を表します。また、いくつかのプロセスでも、Cisco IOS XE ソフトウェアにより感嘆符が表示されることがあります。
[ ]	角カッコは、システムプロンプトに対するデフォルトの応答です。

注意 「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。

（注） 「注釈」です。役立つ情報や、このマニュアル以外の参照資料などを紹介しています。

通信、サービス、およびその他の情報

- シスコからタイムリーな関連情報を受け取るには、Cisco Profile Manager でサインアップしてください。
- 重要な技術によりビジネスに必要な影響を与えるには、Cisco Services [英語] にアクセスしてください。

- サービスリクエストを送信するには、Cisco Support [英語] にアクセスしてください。
- 安全で検証済みのエンタープライズクラスのアプリケーション、製品、ソリューション、およびサービスを探して参照するには、Cisco Marketplace にアクセスしてください。
- 一般的なネットワーク、トレーニング、認定関連の出版物を入手するには、Cisco Press [英語] にアクセスしてください。
- 特定の製品または製品ファミリの保証情報を探すには、Cisco Warranty Finder にアクセスしてください。

Cisco バグ検索ツール

Cisco バグ検索ツール（BST）は、シスコ製品とソフトウェアの障害と脆弱性の包括的なリストを管理する Cisco バグ追跡システムへのゲートウェイとして機能する、Web ベースのツールです。BST は、製品とソフトウェアに関する詳細な障害情報を提供します。

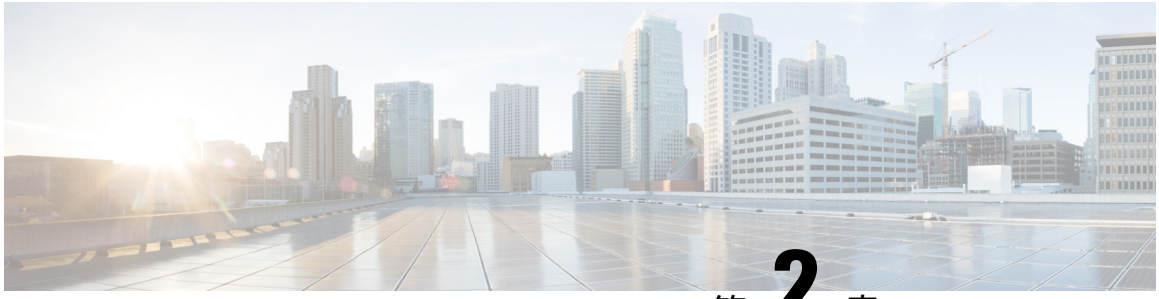
マニュアルに関するフィードバック

シスコのテクニカルドキュメントに関するフィードバックを提供するには、それぞれのオンラインドキュメントの右側のペインにあるフィードバックフォームを使用してください。

トラブルシューティング

トラブルシューティングの最新の詳細情報については、
https://www.cisco.com/c/ja_jp/support/index.html にある Cisco TAC Web サイトを参照してください。

製品カテゴリに移動し、リストから製品を選択するか、製品の名前を入力します。発生している問題に関する情報を見つけるには、**トラブルシュートおよびアラート**を参照してください。



第 2 章

Amazon Web Services での Cisco Catalyst 8000V エッジソフトウェアの概要

Cisco Catalyst 8000V エッジソフトウェア は、マルチテナントのクラウドサービスとしてルーティング、セキュリティ、ネットワーク管理の各機能を提供する仮想ルータです。

このルータは Amazon Virtual Private Cloud（Amazon VPC）でサポートされており、AWS クラウドの論理的に分離されたセクションをプロビジョニングできます。これにより、定義した仮想ネットワークで AWS リソースを起動できます。

Cisco Catalyst 8000V は、自律モードまたはコントローラモードのいずれかで起動できます。デフォルトでは、Cisco Catalyst 8000V は自律モードで起動します。Cisco Catalyst 8000V を自律モードで展開し、使用する場合は、引き続きこのガイドを参照してください。

このガイドでは、パブリックおよびプライベートクラウドソリューションとして Amazon Web Services（AWS）で実行される Cisco Catalyst 8000V の展開オプション、展開手順、および設定について説明します。

Cisco SD-WAN の展開、または Cisco Catalyst 8000V をコントローラモードで展開する場合は、『Getting Started With the Cisco SD-WAN』を参照してください。

Amazon Web Services で Cisco Catalyst 8000V を実行するための展開オプション

Amazon Web Services（AWS）で Cisco Catalyst 8000V を使用するには、AWS Marketplace [英語] で Amazon マシンイメージ（AMI）として Cisco Catalyst 8000V インスタンスを購入して起動します。

Amazon マシンイメージ（AMI）は、インスタンスの起動に必要な情報を提供します。インスタンスを起動するときに AMI を指定する必要があります。AMI から必要な数のインスタンスを起動できることに留意してください。

AWS Marketplace から次の展開オプションのいずれかを選択します。

- Cisco Catalyst 8000V - Advantage PAYG
- Cisco Catalyst 8000V - Essentials PAYG
- Cisco Catalyst 8000V - BYOL
- Cisco Catalyst 8000V - BYOL For SDWAN

最初の 3 つのオプションのいずれかを選択した場合は、展開オプションを選択してからライセンス管理に進みます。Cisco SD-WAN オプションを選択した場合は、『Getting Started with Cisco SD-WAN』ガイドを参照してください。

（注） 以前のバージョンからアップグレードする場合は、新しい AMI から AWS EC2 インスタンスを再作成せずに、Cisco Catalyst 8000V .binファイルを使用して Cisco Catalyst 8000V インスタンスのバージョンをアップグレードします。

ライセンス

AWS Marketplace [英語] にアクセスしたら、AWS Marketplace で Cisco Catalyst 8000V デバイスを Amazon マシンイメージ（AMI）として購入し、起動します。

Cisco Catalyst 8000V デバイスを使用するには、最初にイメージまたはソリューションのリストを選択し、イメージを購入して AMI を展開します。次の手順として、シスコから Cisco Catalyst 8000V ソフトウェアライセンスを直接購入するか、すでにイメージに組み込まれているペイアズユーゴー（PAYG）ライセンスを使用します。

所有ライセンス持ち込み（BYOL）ライセンスモデルを使用している場合は、このセクションの続きを参照してください。それ以外の場合は、本ガイドの「ペイアズユーゴー」のセクションを参照してください。

所有ライセンス持ち込みモデル

所有ライセンス持ち込みは、シスコまたはパートナーからライセンスを購入して、そのライセンスを Cisco Catalyst 8000V デバイスにインストールするモデルです。BYOL ライセンスモデルを選択する場合は、AWS Marketplace から Cisco Catalyst 8000V AMI を展開してインスタンスを起動した後、Cisco Smart Licensing Usage Policy を使用してライセンスをインストールする必要があります。

Cisco Smart Licensing Usage Policy は、ネットワークの運用を中断させないライセンスソリューションを提供するという包括的な目的を持つ、既存のスマートライセンスモデルの進化版で

す。さらに言えば、このモデルは、お客様が購入して使用するハードウェアライセンスとソフトウェアライセンスの信頼性を示すコンプライアンス関係の構築を可能にします。

ライセンスを購入した後は、輸出規制または適用ライセンスを使用しない限り、キーの登録や生成などの準備手順は必要ありません。これらのライセンスは、使用前に承認が必要です。他のすべてのライセンスについては、製品機能をデバイスですぐに設定できます。

Cisco Catalyst 8000V ソフトウェアライセンスとライセンスの再ホストのプロセスの詳細については、『Cisco Catalyst 8000V Edge Software Configuration Guide』を参照してください。ライセンス SKU のリストについては、最新の Cisco Catalyst 8000V リリースノートを参照してください。

ペイアズユーゴー ライセンシング

AWS で Cisco Catalyst 8000V を使用するには、Cisco Catalyst 8000V を AWS Marketplace [英語] で Amazon マシンイメージ（AMI）として購入し、起動する必要があります。さらに、BYOL またはペイアズユーゴー（PAYG）ライセンスモデルを選択する必要があります。

BYOL モデルを選択した場合は、本ガイドの「ライセンス」のセクションを参照してください。PAYG ライセンスモデルを選択した場合は、引き続きこのセクションを参照してください。

Cisco Catalyst 8000V 時間課金 AMI またはペイアズユーゴー ライセンス モデルでは、指定された期間インスタンスを使用できます。このライセンスモデルでは、AWS Marketplace から直接インスタンスを起動して、インスタンスの使用を開始できます。ライセンスはイメージに組み込まれます。

このライセンスモデルでは、次の Cisco IOS XE テクノロジーパッケージを使用できます。**Cisco Catalyst 8000V - Essentials PAYG** および **Cisco Catalyst 8000V - Advantage PAYG**

PAYG は次の条件を前提としています。

- Cisco Catalyst 8000V AMI を使用すると、Amazon Web Services（AWS）によって時間単位で課金されます。この時間単位の使用料は、AWS から請求される VPC 使用料に追加されます。
- シスコから Cisco Catalyst 8000V のライセンスを直接購入することはできません。
- ルータにシスコのライセンスをインストールしないでください。
- 時間課金 AMI を再ホストすることはできません。

Cisco Catalyst 8000V テクノロジーパッケージに含まれる機能の詳細については、『Cisco Catalyst 8000V Edge Software Configuration Guide』を参照してください。

サポートされていない Cisco IOS XE テクノロジー

Cisco Catalyst 8000V インスタンスを AWS インスタンスで展開する場合、Cisco Catalyst 8000V でサポートされる Cisco IOS XE テクノロジーの数は他のハイパーバイザでサポートされる数よりも少なくなります。一部のテクノロジーは Amazon クラウドでサポートされていないために利用できないことがあります。

AWS インスタンスで Cisco Catalyst 8000V を展開する場合は、次の制約事項が適用されます。

- サポートされていない機能の CLI コマンドが Cisco Catalyst 8000V に表示される場合がありますが、シスコによるテストでは、これらのサポートされていない機能（本セクションの表に記載）は AWS 展開では機能しないことが判明しています。
- ルーティングプロトコルは、トンネル経由でのみサポートされます。
- Cisco Catalyst 8000V AMIは、Cisco Prime Network Services Controller を使用したルータのリモート管理をサポートしていません。

次の表に、AWS インスタンスで Cisco Catalyst 8000V を展開する場合にサポートされない Cisco IOS XE テクノロジーの一覧を示します。

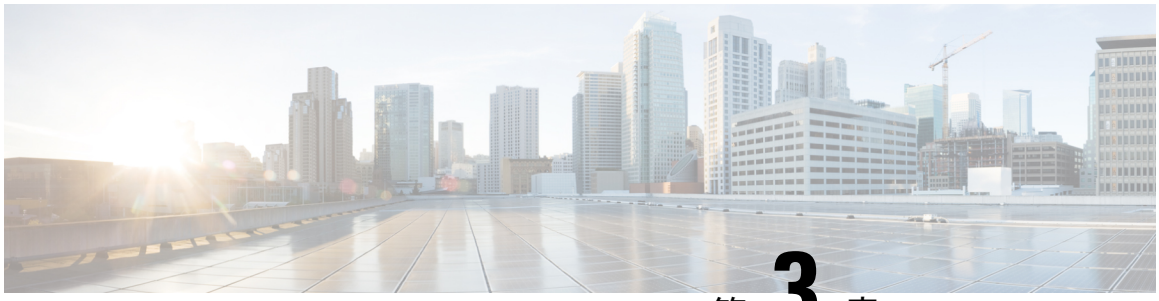
表 *1 : AWS* 展開でサポートされていない *Cisco IOS XE* テクノロジー

テクノロジー	サポートされていない機能
基本ルーティング	OSPF
IP マルチキャスト	IGMP と PIM
データセンター相互接続	OTV、VxLAN および WCCPv2
MPLS	MPLS、EoMPLS、VRF および VPLS
冗長性	HSRP
WAAS	統合された AppNav-XE

AWS 展開での Cisco IOS XE テクノロジーのサポートには次の警告が適用されます。

- 暗号マップが設定されているインターフェイスに NAT PAT を適用することはできません。解決策は、SVTI や DMVPN といった別の IP セキュリティ機能を使用することです。または、NAT 用に 1 台のルータを使用し、IP セキュリティ暗号マップ用に別のルータを使用する 2 ルータソリューションを設定することもできます。
- Amazon クラウド内の Cisco Catalyst 8000V ノード間で HSRP を設定することはできません。Amazon は、VPC 内のホストで HSRP を実行することを許可していません。Amazon AWS は、VPC 内のすべてのブロードキャストトラフィックとマルチキャストトラフィックをブロックします。

- Cisco Catalyst 8000V インターフェイスでの送信元/送信先チェックを無効にすることを推奨します。
- EtherChannel はサポートされていません。



第 3 章

AWS での Cisco Catalyst 8000V の展開

この章では、AWS で Cisco Catalyst 8000V インスタンスを展開する手順について説明します。Cisco Catalyst 8000V インスタンスを展開するには、AWS でサポートおよび管理されている Amazon マシンイメージ（AMI）が必要です。AMI によってインスタンスの起動に必要な情報が提供されます。

AWS Marketplace にログインしたら、適切なテンプレートまたは Marketplace オファーを選択します。さらに、この章で説明する手順に従い、暗号化された Elastic Block Storage（EBS）を使用して AMI を作成します。

（注） BYOL AMI を使用している場合は、ライセンス （6 ページ）を参照してください。

サポートされているインスタンスタイプ

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。Cisco Catalyst 8000V では、次のインスタンスタイプがサポートされています。

- c5.large
- c5.xlarge
- c5.2xlarge
- c5.4xlarge
- c5.9xlarge
- c5n.large
- c5n.xlarge

- c5n.2xlarge
- c5n.4xlarge
- c5n.9xlarge
- c5n.18xlarge
- t3.medium

（注） c5.4xlarge は、Cisco IOS XE 17.10.1 以降ではサポートされていません。

PMD マルチキューをサポートするインスタンスタイプを使用する場合のパフォーマンスの最適化については、PMD マルチキューのサポート （37 ページ） を参照してください。

インスタンスタイプの詳細については、Amazon EC2 インスタンスタイプを参照してください。

インスタンスごとにサポートされるネットワークインターフェイスの最大数を確認するには、「IP Addresses Per Network Interface Per Instance Type」[英語] を参照してください。

AWS で Cisco Catalyst 8000V を展開するための前提条件

AWS で Cisco Catalyst 8000V を起動する前に、次のことを行う必要があります。

- AWS アカウントを用意します。
- Cisco Catalyst 8000V コンソールにアクセスするための SSH クライアント（Windows の場合は Putty、Macintosh の場合は Terminal など）を用意します。
- Cisco Catalyst 8000V AMI のインスタンスタイプを決定します。
- ワンクリック起動を使用して AMI を起動する場合は、Amazon VPC を作成します。

AWS での Cisco Catalyst 8000V の展開に関する制約事項

VPC のジャンボフレームには制限があります。ジャンボフレームの詳細については、「Network Maximum Transmission Unit (MTU) for Your EC2 Instance」 [英語] を参照してください。

Cisco Catalyst 8000V インスタンスの展開

Cisco Catalyst 8000V AMI を展開するには、次のセクションに記載されている手順を実行します。

Cisco Catalyst 8000V Marketplace オファーの選択

ステップ 1 Amazon Web Services Marketplace にログインします。

ステップ 2 [Discover Products] をクリックします。

ステップ 3 検索バーで、Cisco Catalyst 8000V と検索します。次のオファーが表示されます。

- Cisco Catalyst 8000V - Advantage PAYG
- Cisco Catalyst 8000V - Essentials PAYG
- Cisco Catalyst 8000V - BYOL

ステップ 4 展開する予定の Cisco Catalyst 8000V AMI を選択します。

Marketplace には、サポートされるインスタンスタイプ、価格、サポートの詳細などの製品情報が表示されます。

AMI の起動

始める前に

ワンクリック起動で AMI を起動する場合は、まず仮想プライベートクラウド（VPC）を作成する必要があります。作成方法については、VPC に関する AWS のドキュメントを参照してください。

ステップ 1 AWS Marketplace から Cisco Catalyst 8000V オファーを選択したら、[Continue to Subscribe] をクリックします。

ステップ 2 オファーのサブスクリプションが完了したら、[Continue to Configuration] をクリックします。

ステップ 3 [Delivery Method] ドロップダウンリストから、履行オプションを選択します。選択可能な値には、[Amazon Machine Image] や [Cloudformation Template] が含まれます。いずれの場合も、ソフトウェアのバージョンと Cisco Catalyst 8000V を起動するリージョンを選択できます。

Amazon EC2 のゾーンとリージョンについては、「Regions and Availability Zones」 [英語] を参照してください。

ステップ 4 [Continue to Launch] をクリックします

ステップ 5 [Launch This Software] ウィンドウで、[Launch through EC2 Console] または [Launch from Website] を選択します。

ステップ 6 [Launch From Website] オプションを選択すると、一連の追加フィールドが表示されます。ドロップダウンリストから、[EC2 Instance Type]、[VPC]、[Subnet]、[Security Group]、[Key Pair] の適切な設定を選択します。

新しく起動したインスタンスを表示するには、[Launch] をクリックして https://console.aws.amazon.com/ec2/ に移動します。SSH を使用してインスタンスへの接続を試行する前に、[Status Check] に「`2/2 checks passed`」というメッセージが表示されていることを確認します。

ステップ 7 [Launch Through EC2 Console] オプションを選択した場合は、次の手順を実行します。

a) [Launch] をクリックして EC2 コンソールに移動します。

b) [Configure Instance Details] をクリックします。

c) インスタンスの詳細を設定します。ネットワークのドロップダウンリストから、適切なネットワークを選択します。

d) Cisco Catalyst 8000V インスタンスを展開する VPC サブネットを、ドロップダウンリストから選択します。この設定により、インスタンスの可用性ゾーンが決定されます。

（注） 最初に [Instance Details] ウィンドウで 2 つのインターフェイスを作成できます。さらにインターフェイスを追加するには、[Network Interfaces] をクリックします。サポートされるインターフェイスの最大数は、インスタンスタイプによって異なります。

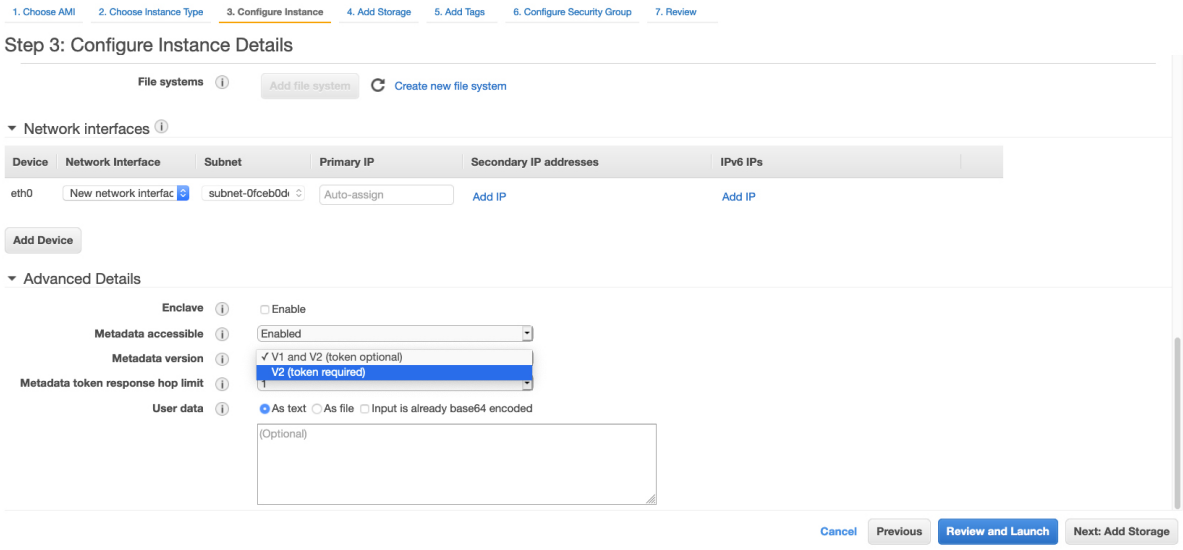
e) [Availability Zone] ドロップダウンリストで適切な可用性ゾーンを選択します。

f) [Metadata Accesible] フィールドにブートストラップオプションを指定して、ブートストラッププロパティを設定します。

（注） インスタンスのメタデータを有効にするには、[Metadata Accessible] フィールドを有効にします。このフィールドを有効にしないと、インスタンスはメタデータサービスにアクセスしてログイン情報やその他の情報を取得することができず、インスタンスにアクセスできなくなります。

g) [Metadata Version] ドロップダウンリストで、適切なメタデータバージョンを選択します。[V1 and V2 (token optional)] または [V2 (token required)] のいずれかを選択します。どちらのシナリオでも、インスタンスは、トークンを作成することでセッション指向の要求を使用します。トークンは、インスタンスに必要なすべてのメタデータを取得するために使用されます。

Cisco IOS XE 17.4.x では、バージョン 1（V1）のみになります。Cisco IOS XE17.6.1 以降では、メタデータバージョン V1 および V2 がサポートされます。



（注） [User Data] フィールドを使用して、カスタムデータ形式でブートストラッププロパティを指定することもできます。サポート対象のカスタムデータ形式については、「Day Zero Configuration」を参照してください。

h) [Next: Add Storage] をクリックして、デフォルトのハードドライブ設定を保持します。

（注） AWS で Cisco Catalyst 8000V インスタンスを使用する場合、仮想ハードドライブのサイズは変更できません。

i) [Next: Tag Instance] をクリックし、タグ情報を入力します。

j) [Next: Configure Security Groups] をクリックします。

k) 新しいセキュリティグループを作成するか、既存のセキュリティグループを選択します。Cisco Catalyst 8000V ではコンソールアクセスに SSH が必要です。さらに、Cisco Catalyst 8000V ではセキュリティグループが TCP/22 をブロックしないことも必要としています。これらの設定は、Cisco Catalyst 8000V インスタンスの管理に使用されます。

l) [Review and Launch] をクリックして Cisco Catalyst 8000V のインスタンス情報を確認し、[Launch] をクリックします。

m) 既存のキーペアを選択するか、独自の公開キーをアップロードして新しいキーを作成します。[Create Key Pair] をクリックして、AWS で新しいキーペアを作成することも可能です。キーペア名を入力して、[Create] をクリックします。

キー ペアが作成されたら、続行する前に Amazon から秘密キーをダウンロードしていることを確認します。新しく作成された秘密キーには一度しかアクセスできません。キーペアをダウンロードしたら、[Close] をクリックします。

（注） AWS セキュリティ ポリシーでは、秘密キーの権限レベルを 400 に設定する必要があります。この値を .pem ファイル用に設定するには、UNIX シェルターミナル画面を開き、**chmod 400** *pem-file-name* コマンドを実行します。

Cisco IOS XE 17.10.1 以降を搭載した Cisco Catalyst 8000V では、ED25519 SSH キーをサポートします。このキーは、既存の SSH-RSA キーに追加されます。キーの生成と検証をこれまで以上に迅速に行い、コリジョンからの復元性とセキュリティを向上させるために、ED25519 SSH キーを使用することを推奨します。

n) [Launch Instance] をクリックします。

AMI インスタンスが展開された後、メニューの [Instances] リンクをクリックしてステータスを表示できます。ステータスが [Running] から [Passed] に変わります。この時点で、Cisco Catalyst 8000V インスタンスが起動され、ソフトウェア設定の準備が完了します。

パブリック IP アドレスと Cisco Catalyst 8000V インスタンスの関連付け

SSH 接続を使用して管理コンソールにアクセスするには、まず Cisco Catalyst 8000V インスタンスのインターフェイスを VPC で作成されたパブリック IP アドレスに関連付ける必要があります。次の手順を実行して、パブリック IP アドレスを Cisco Catalyst 8000V インスタンスに関連付けます。

ステップ 1 [Services] > [EC2] > [Instances] の順に選択し、Cisco Catalyst 8000V インスタンスを選択します。

ステップ 2 [Network interfaces] ウィンドウで、[eth0] をクリックします。

ステップ 3 ダイアログボックスに、eth0 インターフェイスに関する詳細情報が表示されます。インターフェイスのプライベート IP アドレスを書き留めておきます。

ステップ 4 [Interface ID Value] をクリックします。

ステップ 5 [Actions] をクリックし、ドロップダウンリストから [Associate Address] を選択します。

ステップ 6 [Elastic IP address] リストから使用可能なパブリック IP アドレスを選択します。

ステップ 7 （オプション）別の Elastic Network Interface（ENI）にマッピングされている現在使用中のパブリック IP アドレスを再割り当てする場合は、[Allow Reassociation] をクリックします。

ステップ 8 選択したプライベート IP アドレスがステップ 3 でメモしたものと一致することを確認します。

ステップ 9 [Associate Address] をクリックします。

このアクションにより、パブリック IP アドレス（Amazon Elastic IP）がネットワーク インターフェイスのプライベート IP アドレスと関連付けられます。これで、このインターフェイスを使用して管理コンソールにアクセスできるようになります。

SSH を使用したインスタンスへの接続

AWS 上の Cisco Catalyst 8000V インスタンスへのコンソールアクセスには SSH が必要です。Cisco Catalyst 8000V AMI にアクセスするには、次の手順を実行します。

ステップ 1 Cisco Catalyst 8000V インスタンスを起動し、ステータスが [Running] と表示されたら、[Instances] ウィンドウでインスタンスを選択します。

ステップ 2 UNIX シェルコマンド **ssh -i** *pem-file-name* **ec2-user**@[*public-ipaddress* | *DNS-name*] を実行し、SSH を使用して Cisco Catalyst 8000V コンソールに接続します。

- 初めてインスタンスにアクセスするときは、AMI のデフォルトのユーザー名 **ec2-user** を使用します。
- .pem ファイルに保存されている秘密キーを使用して、インスタンスへのアクセスを認証します。

ステップ 3 Cisco Catalyst 8000V インスタンスを開始します。

BYOL AMI のライセンスのダウンロードとアクティブ化については、ライセンス （6 ページ） を参照してください。

SSH キーペアの作成

AWS で Cisco Catalyst 8000V インスタンスを展開するときに、お使いのインスタンスにアクセスするための認証方法として SSH キーを指定できます。この場合、キーペアを作成する必要があります。

キーペアを作成するには、Amazon EC2 を使用して RSA または ED25519 キーペアを作成します。また、他社製ツールを使用してキーペアを作成し、公開キーを Amazon EC2 インスタンスにインポートすることもできます。

キーペアを作成して設定すると、新しい VM が起動し、システムに「`status passes 2/2 check`」というメッセージが表示されます。新しい VM コンソールへは、.pem キーを使用してアクセスできます。また、秘密キーを使用して新しい VM コンソールへのアクセスを認証できます。

暗号化された Elastic Block Storage を使用した AMI の作成

Amazon Elastic Block Storage（EBS）の暗号化は、お使いの EC2 インスタンスに関連付けられた EBS リソースの暗号化ソリューションです。Amazon EBS の暗号化により、AWS KMS キーを使用してデータを確実に保護します。暗号化された Amazon EBS で Cisco Catalyst 8000V AMI を作成するには、次の手順を実行します。

ステップ 1 [Services] > [EC2] > [Instances] の順に選択します。

ステップ 2 暗号化された Amazon EBS で新しい AMI を作成するためのベースとして使用するインスタンスを選択します。ベースインスタンスのステータスが [Stopped] であることを確認してください。

ステップ 3 次の手順 a ～ f に従って、このインスタンスのスナップショットを作成します。

a) ルートデバイス（例：/dev/xvda/）をクリックします。
[Block Device] ダイアログボックスが表示されます。

b) [EBS ID] をクリックします。このスナップショットのボリュームが [ELASTIC BLOCK STORE] > [Volumes] に表示されます。

c) [Actions] > [Create Snapshot] を選択します。
[Create Snapshot] ダイアログボックスが表示されます。

d) [Create] をクリックします。

e) [EBS] ウィンドウの [Create Image] フィールドに、スナップショットの名前を入力します。

f) [Virtualization type] ドロップダウンリストから、[Hardware-assisted virtualization] オプションを選択します。

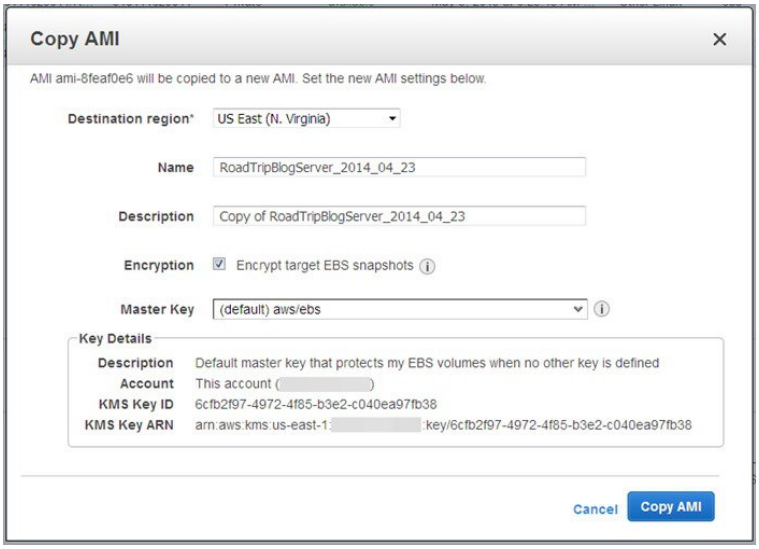
[Create Snapshot] ダイアログボックスに「Snapshot Creation Started」というメッセージが表示されます。スナップショットの作成が完了すると、[ELASTIC BLOCK STORE] > [Snapshots] に、ステータスが [Completed] の新しいスナップショットが表示されます。

ステップ 4 [EC2] > [IMAGES] > [AMIs] を選択して、プライベート AMI を作成します。

以前に作成したスナップショット インスタンスの名前が AMI のリストに表示されます。

ステップ 5 作成したスナップショット インスタンスを選び、[Actions] > [Copy AMI] の順に選択します。

[Copy AMI] ダイアログボックスに、[Destination region]、[Name]、[Description]、[Encryption]、[Master Key]、[Key details] の各フィールドが表示されます。



ステップ 6 [Destination region] ドロップダウンリストで、宛先（[US East] など）を選択します。

ステップ 7 [Name] に「**encrypted-C8000V-1**」といった名前を入力します。

ステップ 8 [Description] を指定します。

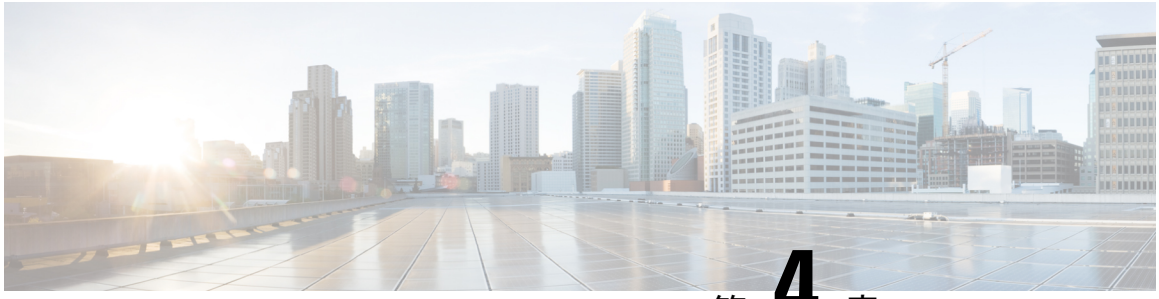
ステップ 9 [Encrypt target EBS snapshots] チェックボックスをオンにします。

ステップ 10 [Master Key] ドロップダウンリストでデフォルト値を選択します。

ステップ 11 [Copy AMI] をクリックします。

暗号化された EBS を使用した新しい AMI が数分後に作成されます。

ステップ 12 新しい AMI のステータスを確認するには、[EC2] > [IMAGES] > [AMIs] に移動します。新しい AMI がリストされていることがわかります。



第 4 章

ゲストシェルの有効化

ゲストシェルの有効化

AWS で実行中の Cisco Catalyst 8000V でゲストシェルを有効にするには、IAM インスタンスロールを作成し、EC2 サービスとの信頼関係を確立します。IAMインスタンスロールを既存の Cisco Catalyst 8000V インスタンスに割り当てるか、それとも IAM インスタンスロールを新しい Cisco Catalyst 8000V インスタンスに割り当てるかを選択できます。

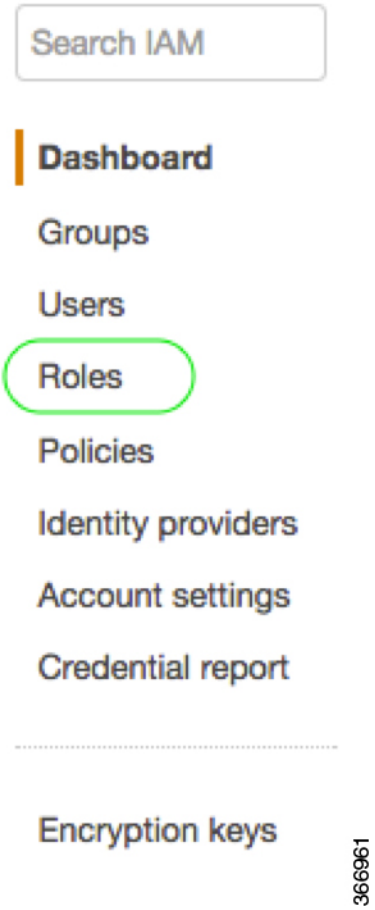
これらのタスクを実行する方法の詳細については、「Cisco Catalyst 8000V への IAM インスタンスロールの割り当て」と「新しい Cisco Catalyst 8000V への IAM インスタンスロールの割り当て」を参照してください。

次に、Cisco Catalyst 8000V でその後の設定手順を実行し、ゲストシェルを開きます。

IAM インスタンスロールの作成

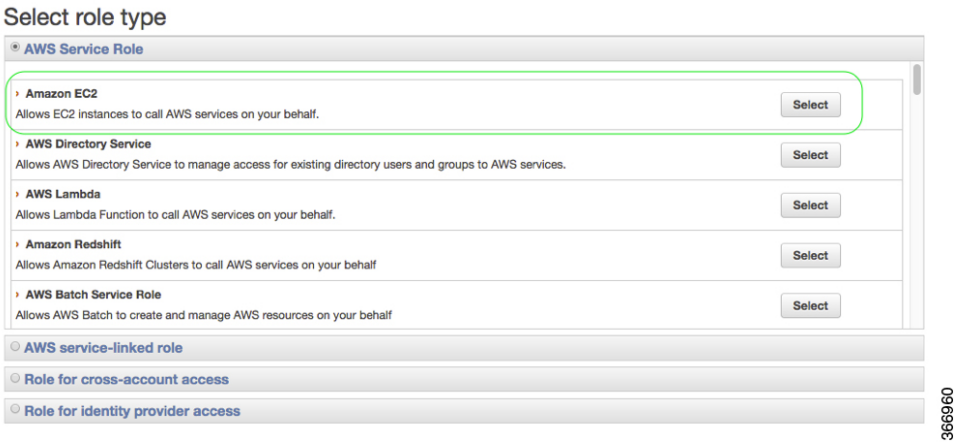
1. IAM ロールを作成する権限を持つ管理者として、AWS にサインインします。
2. [EC2] をクリックして、EC2 コンソールを開始します。
3. [IAM] をクリックして、IAM コンソールを開始します。
4. [Roles] をクリックします。

図 *1 : IAM* インスタンスロール



5. [Create New Role] をクリックします。
6. アプリのロールの名前を入力します。
7. [Continue] をクリックします。
8. ロールタイプを選択します。

図 *2 : IAM* インスタンスロールタイプ



9. Amazon EC2 ロールタイプに関して、[Select] をクリックします。

   この操作により、EC2 サービスとの信頼関係が確立されます。

10. [Set Permissions] で、[Select Policy Template] をクリックします。

11. [Select] をクリックして、テンプレート（[Amazon S3 Full Access] など）を選択します。複数のサービスを選択できます。アクセスをさらに詳細に指定するには、このオプションを使用します。たとえば、IAM インスタンスロールに S3 バケットからの読み取りを許可する一方で、S3 バケットへの書き込みは許可しない設定にすることができます。

12. ロール名を入力します。

13. [Create Role] をクリックします。

Cisco Catalyst 8000V インスタンスへの IAM インスタンスロールの割り当て

IAMインスタンスロールの指定は、ゲストシェルへのアクセスに必須ではありません。とはいえ、指定しておくと、キーまたはパスワードを使用してAWSアカウントの特定のエンティティにアクセスできるようになるため、Cisco Catalyst 8000Vインスタンスのアカウント情報を保存する必要がなくなります。

ステップ 1 [EC2] をクリックして、EC2 ダッシュボードを開きます。

ステップ 2 一覧表示された Cisco Catalyst 8000V インスタンスのいずれかを選択し、右クリックして [Instance Setup] を選択します。次に、[Attach/Replace IAM Role] を選択します。

ステップ 3 ドロップダウンリストから、以前に作成した IAM インスタンスロールを選択します。

ステップ 4 Cisco Catalyst 8000V で次の CLI 設定コマンドを入力し、Cisco Catalyst 8000V を再起動します。

```
Router(config)# interface GigabitEthernet1
Router(config-if)# ip address dhcp
Router(config-if)# ip nat outside
Router(config-if)# exit
Router(config)# interface VirtualPortGroup0
Router(config-if)# ip address 192.168.35.1 255.255.255.0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 overload
Router(config)# ip access-list standard GS_NAT_ACL
Router(config)# permit 192.168.0.0 0.0.255.255
Router(config)# app-hosting appid guestshell
Router(config-app-hosting)# vnic gateway1 virtualportgroup 0 guest-interface 0 guest-ipaddress
192.168.35.2 netmask 255.255.255.0 gateway 192.168.35.1 name-server 8.8.8.8 default
Router(config-app-hosting)# resource profile custom cpu 1500 memory 512
Router(config-app-hosting)# exit
Router(config)# exit
Router# guestshell enable
Router# guestshell run python
```

新しいインスタンスへの IAM インスタンスロールの割り当て

次の手順は、新しい Cisco Catalyst 8000V インスタンスの作成中に IAM インスタンスロールを Cisco Catalyst 8000V に割り当てる方法を示しています。

ステップ 1 EC2 インスタンスとして新しい Cisco Catalyst 8000V を起動し、インスタンスタイプを選択します。

ステップ 2 [Next: Configure Instance Details] をクリックします。

ステップ 3 次の 2 つのうちいずれかの手順を実行します。

a) [IAM role] テキストボックスをクリックして、ドロップダウンリストから既存の IAM インスタンスロールを選択します。

b) [Create new IAM role] をクリックして、新しい IAM インスタンスロールを作成します。

ステップ 4 Cisco Catalyst 8000V インスタンスで次の CLI 設定コマンドを入力し、インスタンスを再起動します。

```
Router(config)# interface GigabitEthernet1
Router(config-if)# ip address dhcp
Router(config-if)# ip nat outside
Router(config-if)# exit
Router(config)# interface VirtualPortGroup0
Router(config-if)# ip address 192.168.35.1 255.255.255.0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 overload
Router(config)# ip access-list standard GS_NAT_ACL
Router(config)# permit 192.168.0.0 0.0.255.255
Router(config)# app-hosting appid guestshell
Router(config-app-hosting)# vnic gateway1 virtualportgroup 0 guest-interface 0 guest-ipaddress
192.168.35.2 netmask 255.255.255.0 gateway 192.168.35.1 name-server 8.8.8.8 default
Router(config-app-hosting)# resource profile custom cpu 1500 memory 512
```

```
Router(config-app-hosting)# exit
Router(config)# exit
Router# guestshell enable
Router# guestshell run python
```

ゲストシェルの例

次の例は、Cisco Catalyst 8000V インスタンスのゲストシェルでパッケージをダウンロードする方法と、他の便利なゲストシェルコマンドのいくつかを示しています。

1. `yum` コマンドまたは `pip3` コマンドを使用してパッケージをインストールします。たとえば、`[guestshell@guestshell ~] sudo pip3 install awscli` コマンドを入力して、AWS CLI と Amazon SDK をインストールします。

```
[guestshell@guestshell ~]$ sudo pip3 install awscli
WARNING: Running pip install with root privileges is generally not a good idea. Try
 `pip3 install --user` instead.
Collecting awscli
  Downloading
https://files.pythonhosted.org/packages/ce/38/6f2b06f0f00e6b381ad4741d0bf97eb2e3faa2328a2dfe31577154a207c/awscli-1.18.157-py2.py3-none-any.whl
 (3.4MB)
    100% |################################| 3.4MB 369kB/s
Collecting colorama<0.4.4,>=0.2.5; python_version != "3.4" (from awscli)
  Downloading
https://files.pythonhosted.org/packages/c9/dc/45cdef1b4d119eb96316b3117e6d5708a08029992b2fee2c143c7a0a5cc5/colorama-0.4.3-py2.py3-none-any.whl
Collecting s3transfer<0.4.0,>=0.3.0 (from awscli)
  Downloading
https://files.pythonhosted.org/packages/69/79/e6afb3d8b0b4e96cefbdb690f741d7dd24547ff1f94240c997a26fa908d3/s3transfer-0.3.3-py2.py3-none-any.whl
 (69kB)
    100% |################################| 71kB 7.3MB/s
Collecting docutils<0.16,>=0.10 (from awscli)
  Downloading
https://files.pythonhosted.org/packages/22/cd/a6aa959dca619918ccb55023b4cb151949c64d4d5d55b3f4ffd7eee0c6e8/docutils-0.15.2-py3-none-any.whl
 (547kB)
    100% |################################| 552kB 2.1MB/s
Collecting PyYAML<5.4,>=3.10; python_version != "3.4" (from awscli)
  Downloading
https://files.pythonhosted.org/packages/64/c2/b80047c7ac2478f9501676c988a5411ed5572f35d1beff9cae07d321512c/PyYAML-5.3.1.tar.gz
 (269kB)
    100% |################################| 276kB 3.6MB/s
Collecting rsa<=4.5.0,>=3.1.2; python_version != "3.4" (from awscli)
  Downloading
https://files.pythonhosted.org/packages/26/f8/8127fcda0294f044121d20aac7785feb810e159098447967a6103dedf96/rsa-4.5-py2.py3-none-any.whl
Collecting botocore==1.18.16 (from awscli)
  Downloading
https://files.pythonhosted.org/packages/2d/9e/afa41db0cc9118693f5db783b9b021be67ea23c8b7b317caa4632dbf3cf/botocore-1.18.16-py2.py3-none-any.whl
 (6.7MB)
    100% |################################| 6.7MB 173kB/s
Collecting pyasn1>=0.1.3 (from rsa<=4.5.0,>=3.1.2; python_version != "3.4"->awscli)
  Downloading
https://files.pythonhosted.org/packages/62/1e/a94a8d635fa3ce4cfc7f506003548d0a2447ae76fd5ca53932970fe3053f/pyasn1-0.4.8-py2.py3-none-any.whl
 (77kB)
    100% |################################| 81kB 7.5MB/s
Collecting urllib3<1.26,>=1.20; python_version != "3.4" (from
botocore==1.18.16->awscli)
  Downloading
https://files.pythonhosted.org/packages/9f/f0/a391d1463ebb1b233795cabfc0ef38d3db4442339de68f847026199e69d7/urllib3-1.25.10-py2.py3-none-any.whl
 (127kB)
```

```
    100% |################################| 133kB 6.1MB/s
Collecting python-dateutil<3.0.0,>=2.1 (from botocore==1.18.16->awscli)
  Downloading
https://files.pythonhosted.org/packages/d4/70/d60450c3dd48ef87586924207ae8907090de0b306af2bce5d134d78615cb/python_dateutil-2.8.1-py2.py3-none-any.whl
 (227kB)
    100% |################################| 235kB 4.0MB/s
Collecting jmespath<1.0.0,>=0.7.1 (from botocore==1.18.16->awscli)
  Downloading
https://files.pythonhosted.org/packages/07/cb/5f001272b6faeb23c1c9e0acc04d48eaaf5c862c17709d20e3469c6e0139/jmespath-0.10.0-py2.py3-none-any.whl
Collecting six>=1.5 (from python-dateutil<3.0.0,>=2.1->botocore==1.18.16->awscli)
  Downloading
https://files.pythonhosted.org/packages/ee/ff/48bde5c0f013094d729fe4b0316ba2a24774b3ff1c52d924a8a4cb04078a/six-1.15.0-py2.py3-none-any.whl
Installing collected packages: colorama, urllib3, six, python-dateutil, jmespath,
botocore, s3transfer, docutils, PyYAML, pyasn1, rsa, awscli
  Running setup.py install for PyYAML ... done
Successfully installed PyYAML-5.3.1 awscli-1.18.157 botocore-1.18.16 colorama-0.4.3
 docutils-0.15.2 jmespath-0.10.0 pyasn1-0.4.8 python-dateutil-2.8.1 rsa-4.5
s3transfer-0.3.3 six-1.15.0 urllib3-1.25.10
[guestshell@guestshell ~]$ aws s3 ls c8kv
Unable to locate credentials. You can configure credentials by running "aws configure"
```

2. AWS CLI をインストールしたら、`aws s3 ls` などの `aws s3` コマンドを入力します。

```
[guestshell@guestshell ~]$ aws s3 ls c8kv
2020-10-14 19:44:08  433546509 upgrade.bin
[guestshell@guestshell ~]$
```

3. `sudo pip3 install csr_aws_guestshell` コマンドを使用して、サンプルスクリプトを含む Cisco Catalyst 8000V AWS パッケージをダウンロードできます。

   例：

```
[guestshell@guestshell ~]$ sudo pip3 install csr_aws_guestshell
WARNING: Running pip install with root privileges is generally not a good idea. Try
 `pip3 install --user` instead.
Collecting csr_aws_guestshell
Downloading
https://files.pythonhosted.org/packages/42/a7/c772726166f809b823ef448f5d7fa2cfb8a809525a11991612d81cdb080a/csr_aws_guestshell-0.0.17.dev.tar.gz
Collecting awscli (from csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/23/1b/2b65ddb4e18ab8b232d413b7591ccccd2202123ffb3c3fd1dbb099d0170ce/awscli-1.18.162-py2.py3-none-any.whl
 (3.4MB)
100% |################################| 3.4MB 352kB/s
Collecting boto (from csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/23/10/c0b78c27298029e4454a472a1919bde20cb182dab1662cec7f2ca1dcc523/boto-2.49.0-py2.py3-none-any.whl
 (1.4MB)
100% |################################| 1.4MB 794kB/s
Collecting boto3 (from csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/30/3c/c9b65c398de6589c93b3c65aa27a695a6cdf7588c50a7934ac3ac6599a8a/boto3-1.16.2-py2.py3-none-any.whl
 (129kB)
100% |################################| 133kB 7.2MB/s
Collecting rsa<=4.5.0,>=3.1.2; python_version != "3.4" (from
awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/26/f8/8127fdda0294f044121d20aac7785feb810e15909844796a6103dedfc96/rsa-4.5-py2.py3-none-any.whl
Collecting botocore==1.19.2 (from awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/1f/96/35fa364675cf17e3a190ae0816ac4b078ca86a62ef071d32c0f86d52bc/botocore-1.19.2-py2.py3-none-any.whl
 (6.7MB)
100% |################################| 6.7MB 164kB/s
Collecting PyYAML<5.4,>=3.10; python_version != "3.4" (from awscli->csr_aws_guestshell)
```

```
Downloading
https://files.pythonhosted.org/packages/64/c2/b80047c7ac2478f9501676c988a5411ed5572f35d1beff9cae07d321512c/PyYAML-5.3.1.tar.gz
 (269kB)
100% |################################| 276kB 3.6MB/s
Collecting s3transfer<0.4.0,>=0.3.0 (from awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/69/79/e6afb3d8b0b4e96cefbdc690f741d7dd24547ff1f94240c997a26fa908d3/s3transfer-0.3.3-py2.py3-none-any.whl
 (69kB)
100% |################################| 71kB 7.6MB/s
Collecting docutils<0.16,>=0.10 (from awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/22/cd/a6aa959dca619918ccb55023b4cb151949c64d4d5d55b3f4ffd7eee0c6e8/docutils-0.15.2-py3-none-any.whl
 (547kB)
100% |################################| 552kB 1.9MB/s
Collecting colorama<0.4.4,>=0.2.5; python_version != "3.4" (from
awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/c9/dc/45cdef1b4d119eb96316b3117e6d5708a08029992b2fee2c143c7a0a5cc5/colorama-0.4.3-py2.py3-none-any.whl
Collecting jmespath<1.0.0,>=0.7.1 (from boto3->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/07/cb/5f001272b6faeb23c1c9e0acc04d48eaaf5c862c17709d20e3469c6e0139/jmespath-0.10.0-py2.py3-none-any.whl
Collecting pyasn1>=0.1.3 (from rsa<=4.5.0,>=3.1.2; python_version !=
"3.4"->awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/62/1e/a94a8d635fa3ce4cfc7f506003548d0a2447ae76fd5ca53932970fe3053f/pyasn1-0.4.8-py2.py3-none-any.whl
 (77kB)
100% |################################| 81kB 9.4MB/s
Collecting urllib3<1.26,>=1.25.4; python_version != "3.4" (from
botocore==1.19.2->awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/56/aa/4ef5aa67a9a6255db124a5db5262332b1d4153462eb8fd89c9fa41e5d92/urllib3-1.25.11-py2.py3-none-any.whl
 (127kB)
100% |################################| 133kB 6.5MB/s
Collecting python-dateutil<3.0.0,>=2.1 (from
botocore==1.19.2->awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/d4/70/d60450c3dd48ef87586924207ae8907090de0b306af2bce5d134d78615cb/python_dateutil-2.8.1-py2.py3-none-any.whl
 (227kB)
100% |################################| 235kB 4.6MB/s
Collecting six>=1.5 (from
python-dateutil<3.0.0,>=2.1->botocore==1.19.2->awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/ee/ff/48bde5c0f013094d729fe4b0316ba2a24774b3ff1c52d924a8a4cb04078a/six-1.15.0-py2.py3-none-any.whl
Installing collected packages: pyasn1, rsa, urllib3, six, python-dateutil, jmespath,
 botocore, PyYAML, s3transfer, docutils, colorama, awscli, boto, boto3,
csr-aws-guestshell
Running setup.py install for PyYAML ... done
Running setup.py install for csr-aws-guestshell ... done
Successfully installed PyYAML-5.3.1 awscli-1.18.162 boto-2.49.0 boto3-1.16.2
botocore-1.19.2 colorama-0.4.3 csr-aws-guestshell-0.0.17.dev0 docutils-0.15.2
jmespath-0.10.0 pyasn1-0.4.8 python-dateutil-2.8.1 rsa-4.5 s3transfer-0.3.3 six-1.15.0
 urllib3-1.25.11
```

次のスクリプトが csr_aws_guestshell パッケージに含まれています。

get-metadata.py：AWS からインスタンスメタデータを取得して出力します。

get-route-table.py：VPC のインスタンスをルート、ルートテーブル、関連付けを含めて取得します。

save-config-to-s3.py：Cisco IOS XE CLI コマンドを S3 バケットに保存します。

save-tech-support-to-s3.py：テクニカルサポートの出力を S3 バケットに保存します。

load-bin-from-s3.py：Cisco Catalyst 8000V 用の .binファイルをダウンロードしてリロードします。

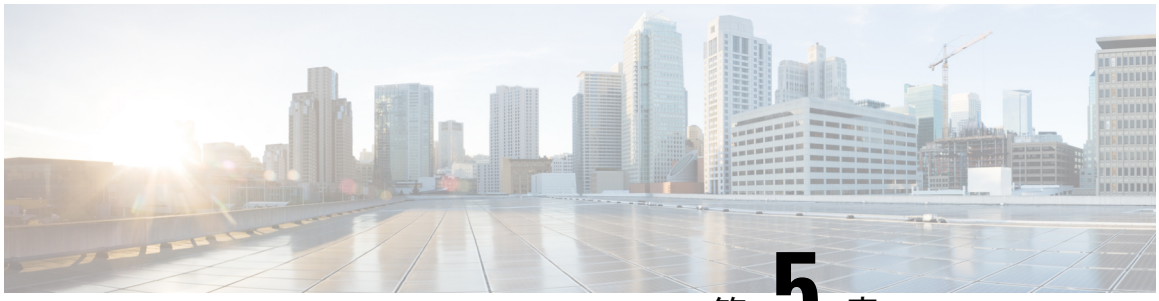
get-stat-drop.py：CLI 統計情報を取得して、CloudWatch にプッシュします。

capture-interface.py：監視に使用する Cisco IOS XE CLI コマンドを設定して、一定期間パケットをキャプチャした後、ファイルを S3 にアップロードします。

4. 次の例では、load-bin-from-s3.py スクリプトが S3 からバイナリをロードし、Cisco Catalyst 8000V イメージを起動します。

```
[guestshell@guestshell ~]$ load-bin-from-s3.py csr1kv ultra_167.bin
/bootflash/ultra_167.bin  446866343 / 446866343  (100.00%)
Download Complete
```

（注） `csr_aws_guestshell` パッケージは、引き続き Cisco Catalyst 8000V と連動します。



第 5 章

パブリッククラウド用 L2 拡張の設定

この章では、企業とクラウドプロバイダーが LISP を使用して Cisco Catalyst 8000V インスタンスを含むパブリッククラウドの L2 拡張を設定できるようにする方法について説明します。コマンドライン インターフェイスを使用して、パブリック クラウド ネットワークとエンタープライズ ネットワーク間のレイヤ 2 ドメインを拡張します。

LISP レイヤ 2 拡張を設定する前に理解しておく必要がある用語と概念の一部を次に示します。

- **Locator/ID Separation Protocol（LISP）**：LISP は、単一 IP アドレスではなく 2 つの名前空間を使用するネットワークアーキテクチャおよびプロトコルです。
  - エンドポイント識別子（EID）：エンドホストに割り当てられます。
  - ルーティングロケータ（RLOC）：グローバル ルーティング システムを構成するデバイス（主にルータ）に割り当てられます。
- **LISP 対応仮想化ルータ**：ルーティング機能と LISP 機能（ホストモビリティを含む）をサポートする仮想マシンまたはアプライアンス。
- **エンドポイント ID（EID）**：EID は、パケットの最初の（最も内側の）LISP ヘッダーに含まれる送信元および宛先アドレスフィールドで使用される IPv4 または IPv6 アドレスです。
- **ルーティングロケータ（RLOC）**：LISP ノード間のフローをカプセル化および転送するために使用される IPv4 または IPv6 アドレス。RLOC は、EID-to-RLOC マッピングルックアップの出力です。
- **出力トンネルルータ（ETR）**：ETR はトンネルエンドポイントであるデバイスで、LISP 機能のあるコアネットワークの部分（インターネットなど）にサイトを接続し、サイトの EID-to-RLOC マッピングを公開し、Map-Request メッセージに応答し、サイトのエンドシステムに LISP でカプセル化されたユーザーデータをカプセル化解除して配信します。運用中、ETR は設定済みのすべての Map Server に定期的に Map-Register メッセージを送信します。送信される Map-Register メッセージには、ETR のサイトに接続されている EID 番号付きネットワークのすべての EID-to-RLOC エントリが含まれます。
- **入力トンネルルータ（ITR）**：ITR はトンネルの開始点となるデバイスです。ITR は、LISP 機能のあるサイトに向かうすべてのトラフィックの EID-to-RLOC マッピングを検索しま

す。ITR が EID 宛てのパケットを受信すると、まずマッピングキャッシュの EID を調べます。ITR が一致を見つけると、LISP ヘッダー内でパケットをカプセル化し、RLOC の 1 つを送信元 IP アドレスとし、マッピング キャッシュ エントリからの RLOC の 1 つを IP 接続先とします。ITR はその後、パケットを通常どおりルーティングします。

- **xTR**：入力トンネルルータ（ITR）機能と出力トンネルルータ（ETR）機能の両方を実行するデバイスの総称。
- **PxTR**：IP ネットワークと LISP ネットワーク間の相互接続ポイント。このピアリングポイントで ITR と ETR の役割を果たします。
- **マップサーバー（MS）**：MS は、LISP サイト ETR がその EID プレフィクスを登録する LISP インフラストラクチャ デバイスです。MS は、クライアント出力トンネルルータ（ETR）からの登録要求を承認し、正常に登録されたそれらの ETR の EID プレフィックスを集約し、Border Gateway Protocol（BGP）を用いて集約されたプレフィックスを 代替論理トポロジ（ALT）にアドバタイズすることで、分散 LISP マッピングデータベースの一部を実行します。

  小規模なプライベート マッピング システム展開では、すべての ETR がそれぞれの MS に登録されるように設定した状態で、MS はスタンドアローンとして設定できます（または複数の MS があってもよい）。複数の場合、すべての MS はプライベート マッピング システム展開内のマッピングシステムの完全な情報を有します。

  より大規模なマッピングシステム展開またはパブリック マッピング システム展開では、MS は、Generic Routing Encapsulation（GRE）トンネルと BGP セッションの部分メッシュを用いて、他のマップサーバーシステムに対して設定されます。
- **マップリゾルバ（MR）**：MR は LISP インフラストラクチャ デバイスです。ITR は、EID-to-RLOC マッピングを解決する際に、LISP Map-Request クエリを MR に送信します。MR は要求を受信し、適切なマップサーバーを選択します。

LISP と用語の詳細な概要については、「Locator ID Separation Protocol Overview」を参照してください。

LISP レイヤ 2 拡張の設定

Cisco Catalyst 8000V は、パブリッククラウド、プライベートクラウド、およびハイブリッドクラウドに展開できます。企業がハイブリッドクラウドに移行する場合、サーバーに対して一切変更を加えずに、サーバーをクラウドに移行する必要があります。企業は、同じサーバー IP

アドレス、サブネットマスク、およびデフォルトゲートウェイ設定を使用することを望むかもしれません。クラウド内で独自の IP アドレス方式を使用し、クラウドプロバイダーのインフラストラクチャのアドレス方式によって制限されないことを望む可能性があります。

この要件を満たすために、シスコは Amazon Web Services（AWS）上で動作する LISP レイヤ 2 拡張を Cisco Catalyst 8000V に提供します。この場合、Cisco Catalyst 8000V インスタンスはエンタープライズ データセンターとパブリッククラウド間のブリッジとして機能します。LISP レイヤ 2 拡張を設定すると、プライベートデータセンター内のレイヤ 2 ネットワークをパブリッククラウドに拡張して、お客様のサイトとパブリッククラウド間でのホスト到達可能性を実現できるようになります。また、データセンターとパブリッククラウド間のアプリケーションワークロードの移行を有効にすることもできます。

利点

- データ移行が容易になり、ネットワークのワークロード IP アドレスやファイアウォールルールが最適化されます。これにより、ブロードキャストドメインを拡張せずにサブネットの連続性を確保できます。
- プロバイダーサイトで VM を仮想的に追加し、VM がプロバイダーサイトで実行されている間に、クラウドバーストを活用して、仮想的に VM をエンタープライズサーバーに挿入できるようにします。
- 部分的な障害回復と障害回避のためのバックアップサービスを提供します。

LISP レイヤ 2 拡張の設定の前提条件

各 Cisco Catalyst 8000V ルータに 1 つの外部 IP アドレスを設定する必要があります。この場合、IPsec トンネルは 2 つの Cisco Catalyst 8000V インスタンスの IP アドレス間に構築され、IPsec トンネルにはプライベートアドレスがあります。

LISP レイヤ 2 拡張の設定の制約事項

- AWS ECS サブネットでは、企業 VRF 数と VM アドレス数が制限されます。
- IPv6 アドレス形式は、Cisco Catalyst 8000V Amazon マシンイメージ（AMI）ではサポートされていません。

LISP レイヤ 2 拡張の設定

L2 拡張機能を設定するには、まず AWS に Cisco Catalyst 8000V インスタンスを展開し、インスタンスを xTR として設定する必要があります。その後、展開を完了するためにマッピングシステムを設定する必要があります。

LISP サイトは、アップストリーム プロバイダーへの 2 系統の接続を持つ、ITR と ETR の両方として設定された（xTR とも呼ばれる）Cisco Catalyst 8000V インスタンスを使用します。次に LISP サイトは、ネットワークコアのマップリゾルバ/マップサーバー（MR/MS）として設定されたスタンドアロンデバイスに登録されます。マッピングシステムは、移行済みのパブリック IP に送信されるパケットの LISP カプセル化およびカプセル化解除を実行します。AWS からのトラフィックについては、必要に応じて（接続先へのルートがルーティングテーブルで見つからない場合は常に）、Cisco Catalyst 8000V インスタンスがエンタープライズ データセンターの PxTR を介してルーティングします。

LISP マップサーバーおよびマップリゾルバをマッピングサービスに使用する際、LISP xETR 機能を設定して有効化するには、次の手順を実行します。

AWS での Cisco Catalyst 8000V インスタンスの作成

ステップ 1 Amazon Web Services にログインします。左側のナビゲーションウィンドウで、[VPC] をクリックします。

ステップ 2 [Start VPC Wizard] をクリックし、左側のペインから [VPC with Single Public Subnet] を選択します。

ステップ 3 [Select] をクリックします。

ステップ 4 仮想プライベートクラウドにサブネットを作成します。次のプロパティを使用します。

a) Default Subnet：10.0.0.0/24（パブリック IP にマッピングされる）。

b) Additional subnets：0.0.1.0/24 および 1.0.0.2.0/24。これらはプライベート IP アドレスであり、Cisco Catalyst 8000V インスタンスから見て内部である可能性があります。

ステップ 5 [Create VPC] を選択します。

ステップ 6 [Security] > [Network ACLs] を選択します。

ステップ 7 [Create Security Group] をクリックして、Cisco Catalyst 8000V インスタンスのセキュリティグループを作成します。次のプロパティを設定します。

a) Name：SSH アクセス

b) TCP Port 22 traffic：インバウンド許可

c) SSH access to C8000V for management：有効

ステップ 8 追加のセキュリティグループを作成するには、ステップ 6 を実行します。

ステップ 9 Cisco Catalyst 8000V の製品ページに移動し、[Continue] をクリックします。

ステップ 10 [Launch with E2 Console] をクリックして、地理的地域に応じた Cisco Catalyst 8000V を起動します。

ステップ 11 適切なインスタンスタイプを選択します。サポートされているインスタンスタイプについては、表 2-1 および 2-2 を参照してください。

中規模インスタンスタイプ（m1.medium）の最小メモリ要件は 10Mbps です。大規模インスタンスタイプ（m1.large）の場合は 50Mbps です。

ECU は Elastic Compute Unit の略です。ECU は、CPU 容量を測定する Amazon 独自の方法です。

すべての EC2 インスタンスはハイパースレッド化されています。

ステップ 12 作成した VPC で Cisco Catalyst 8000V インスタンスを起動します。次のプロパティを使用します。

a) [Shutdown] 動作を [Stop] に設定します。
b) [Tenancy] を [Shared] に設定します。共有ハードウェアインスタンスを実行するには、[Shared] オプションを選択します。

ステップ 13 インスタンスをセキュリティグループ（SSH-ACCESS）に関連付けます。セキュリティルールを使用すると、Cisco Catalyst 8000V インスタンスのトラフィックを制御するファイアウォールルールを設定できます。

ステップ 14 秘密キーを Cisco Catalyst 8000V インスタンスに関連付けます。キーペアは、秘密キーと公開キーで構成されます。Cisco Catalyst 8000V インスタンスを認証して接続するには、秘密キーを指定する必要があります。公開キーは AWS に保存されます。必要に応じて、新しいキーペアを作成できます。

ステップ 15 [Launch Instance] をクリックします。

ステップ 16 Cisco Catalyst 8000V インスタンスが AWS に展開されているかどうかを確認します。

展開に成功すると、ステータスが *2/2/ checks passed* に変わります。

サブネットの設定

ステップ 1 Cisco Catalyst 8000V インスタンスを選択します。

ステップ 2 [Actions] > [Networking] > [Manage IP Addresses] の順に選択します。

ステップ 3 エンタープライズ ホスト アドレスを指定します。この IP アドレスは、eth1 のセカンダリアドレスです。

ステップ 4 [Yes, Update] をクリックします。

AWS 上の Cisco Catalyst 8000V とエンタープライズシステム上の Cisco Catalyst 8000V 間におけるトンネルの設定

エンタープライズデータセンター内に展開された Cisco Catalyst 8000V インスタンスとパブリッククラウド内に展開された Cisco Catalyst 8000V インスタンス間の通信は、両者の間に確立された IP セキュリティ（IPsec）トンネルによって保護されます。LISP カプセル化トラフィックは、パブリッククラウドと企業間のデータ発信元認証、完全性保護、アンチリプライ保護、および機密性を実現する IPsec トンネルで保護されます。

ステップ 1 AWS で Cisco Catalyst 8000V インスタンスを設定します。

```
interface Loopback1
 ip address 33.33.33.33 255.255.255.255
!
interface Tunnel2
 ip address 30.0.0.2 255.255.255.0
 tunnel source GigabitEthernet1
 tunnel mode ipsec ipv4
 tunnel destination 173.39.145.79
 tunnel protection ipsec profile p2p_pf1
```

```
!
interface GigabitEthernet2
 ip address 10.10.10.140 255.255.255.0
 negotiation auto
 lisp mobility subnet1 nbr-proxy-reply requests 3
 no mop enabled
 no mop sysid
!
```

ステップ 2 企業サイトで 2 番目の Cisco Catalyst 8000V インスタンスを設定します。

```
interface Loopback1
 ip address 11.11.11.11 255.255.255.255

interface Tunnel2
 ip address 30.0.0.1 255.255.255.0
 tunnel source GigabitEthernet2
 tunnel mode ipsec ipv4
 tunnel destination 52.14.116.161
 tunnel protection ipsec profile p2p_pf1
!
!
interface GigabitEthernet3
 ip address 10.10.10.2 255.255.255.0
 negotiation auto
 lisp mobility subnet1 nbr-proxy-reply requests 3
 no mop enabled
 no mop sysid
!
```

AWS で実行されているインスタンスでの LISP xTR の設定

AWS で実行されている Cisco Catalyst 8000V インスタンスで LISP xTR を設定するには、「Configuring LISP (Location ID Separation Protocol)」のセクションの設定手順に従います。

例：

```
router lisp
 locator-set aws
  33.33.33.33 priority 1 weight 100
  exit-locator-set
 !
 service ipv4
  itr map-resolver 11.11.11.11
  itr
  etr map-server 11.11.11.11 key cisco
  etr
  use-petr 11.11.11.11
  exit-service-ipv4
 !
 instance-id 0
  dynamic-eid subnet1
   database-mapping 10.10.10.0/24 locator-set aws
   map-notify-group 239.0.0.1
   exit-dynamic-eid
  !
  service ipv4
   eid-table default
```

```
   exit-service-ipv4
  !
  exit-instance-id
 !
 exit-router-lisp
!
router ospf 11
 network 30.0.0.2 0.0.0.0 area 11
 network 33.33.33.33 0.0.0.0 area 11
!

router lisp
 locator-set dmz
  11.11.11.11 priority 1 weight 100
  exit-locator-set
 !
 service ipv4
  itr map-resolver 11.11.11.11
  etr map-server 11.11.11.11 key cisco
  etr
  proxy-etr
  proxy-itr 11.11.11.11
  map-server
  map-resolver
  exit-service-ipv4
 !
 instance-id 0
  dynamic-eid subnet1
   database-mapping 10.10.10.0/24 locator-set dmz
   map-notify-group 239.0.0.1
   exit-dynamic-eid
  !
  service ipv4
   eid-table default
   exit-service-ipv4
  !
  exit-instance-id
 !
 site DATA_CENTER
  authentication-key cisco
  eid-record 10.10.10.0/24 accept-more-specifics
  exit-site
 !
 exit-router-lisp
!
router ospf 11
 network 11.11.11.11 0.0.0.0 area 11
 network 30.0.0.1 0.0.0.0 area 11
!

!
!
```

AWS 上の Cisco Catalyst 8000V とエンタープライズシステム上の Cisco Catalyst 8000V 間における LISP レイヤ 2 トラフィックの確認

LISP レイヤ 2 トラフィックを確認するには、次の手順を実行します。

例：

```
Router#show ip lisp database
LISP ETR IPv4 Mapping Database for EID-table default (IID 0), LSBs: 0x1
Entries total 2, no-route 0, inactive 0

10.0.1.1/32, dynamic-eid subnet1, inherited from default locator-set aws
  Locator  Pri/Wgt  Source     State
33.33.33.33    1/100  cfg-addr   site-self, reachable
10.0.1.20/32, dynamic-eid subnet1, inherited from default locator-set aws
  Locator  Pri/Wgt  Source     State
33.33.33.33    1/100  cfg-addr   site-self, reachable
Router#show ip lisp map-cache
LISP IPv4 Mapping Cache for EID-table default (IID 0), 4 entries

0.0.0.0/0, uptime: 00:09:49, expires: never, via static-send-map-request
  Negative cache entry, action: send-map-request
10.0.1.0/24, uptime: 00:09:49, expires: never, via dynamic-EID, send-map-request
  Negative cache entry, action: send-map-request
10.0.1.4/30, uptime: 00:00:55, expires: 00:00:57, via map-reply, forward-native
  Encapsulating to proxy ETR
10.0.1.100/32, uptime: 00:01:34, expires: 23:58:26, via map-reply, complete
  Locator  Uptime    State      Pri/Wgt     Encap-IID
11.11.11.11  00:01:34  up           1/100       -
Router#show lisp dynamic-eid detail
% Command accepted but obsolete, unreleased or unsupported; see documentation.

LISP Dynamic EID Information for VRF "default"

Dynamic-EID name: subnet1
  Database-mapping EID-prefix: 10.0.1.0/24, locator-set aws
  Registering more-specific dynamic-EIDs
  Map-Server(s): none configured, use global Map-Server
  Site-based multicast Map-Notify group: 239.0.0.1
  Number of roaming dynamic-EIDs discovered: 2
  Last dynamic-EID discovered: 10.0.1.20, 00:01:37 ago
    10.0.1.1, GigabitEthernet2, uptime: 00:09:23
      last activity: 00:00:42, discovered by: Packet Reception
    10.0.1.20, GigabitEthernet2, uptime: 00:01:37
      last activity: 00:00:40, discovered by: Packet Reception

Router-DC#show ip lisp
Router-DC#show ip lisp data
Router-DC#show ip lisp database
LISP ETR IPv4 Mapping Database for EID-table default (IID 0), LSBs: 0x1
Entries total 1, no-route 0, inactive 0

10.0.1.100/32, dynamic-eid subnet1, inherited from default locator-set dc
  Locator  Pri/Wgt  Source     State
11.11.11.11    1/100  cfg-addr   site-self, reachable
```

```
Router-DC#show ip lisp
Router-DC#show ip lisp map
Router-DC#show ip lisp map-cache
LISP IPv4 Mapping Cache for EID-table default (IID 0), 2 entries

10.0.1.0/24, uptime: 1d08h, expires: never, via dynamic-EID, send-map-request
  Negative cache entry, action: send-map-request
10.0.1.20/32, uptime: 00:00:35, expires: 23:59:24, via map-reply, complete
  Locator  Uptime    State      Pri/Wgt     Encap-IID
33.33.33.33  00:00:35  up           1/100

Router-DC#show lisp dynamic-eid detail
% Command accepted but obsolete, unreleased or unsupported; see documentation.

LISP Dynamic EID Information for VRF "default"

Dynamic-EID name: subnet1
  Database-mapping EID-prefix: 10.0.1.0/24, locator-set dc
  Registering more-specific dynamic-EIDs
  Map-Server(s): none configured, use global Map-Server
  Site-based multicast Map-Notify group: 239.0.0.1
  Number of roaming dynamic-EIDs discovered: 1
  Last dynamic-EID discovered: 10.0.1.100, 1d08h ago
    10.0.1.100, GigabitEthernet2, uptime: 1d08h
      last activity: 00:00:47, discovered by: Packet Reception

Router-DC#show lisp site
LISP Site Registration Information
* = Some locators are down or unreachable
# = Some registrations are sourced by reliable transport

Site Name      Last      Up     Who Last             Inst     EID Prefix
               Register         Registered           ID
dc             never     no     --                            10.0.1.0/24
               00:08:41  yes#   33.33.33.33                   10.0.1.1/32
               00:01:00  yes#   33.33.33.33                   10.0.1.20/32
               1d08h     yes#   11.11.11.11                   10.0.1.100/32
Router-DC#show ip cef 10.0.1.20
10.0.1.20/32
  nexthop 33.33.33.33 LISP0
Router-DC#
```

PMD マルチキューのサポート

Cisco IOS XE 17.7.1 以降では、AWS で実行される Cisco Catalyst 8000V インスタンスで PMD マルチキュー機能がサポートされます。現在、Cisco Catalyst 8000V で割り当てられる PMD RX キューと PMD TX キューはインターフェイスごとに 1 つだけです。この機能を使用すると、Cisco Catalyst 8000V で 4 つの PMD RX キューと 8 つの PMD TX キューが割り当てられます。これにより、パケット処理率が増加してパフォーマンスが向上します。

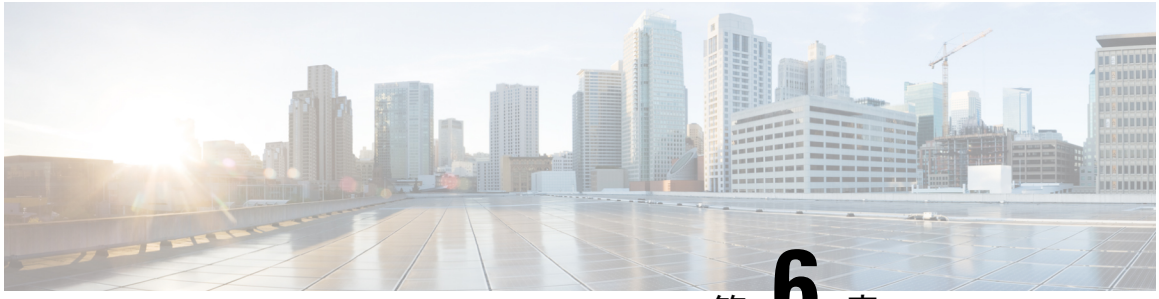
Cisco IOS XE 17.9.1 以降では、Cisco Catalyst 8000V による PMD TX キューの割り当てが 12 個に増加しています。

（注） IPsec トンネルの IP アドレスペアは PMD TXQ にハッシュされます。したがって、アドレスが競合してパフォーマンスが低下することがあります。この問題を回避するには、**show platform hardware qfp active datapath infrastructure sw-nic** コマンドを使用して、パフォーマンスが最適になるようにトラフィックが 8 つのキューすべてに均等に分散しているかどうかを確認します。

次に、**show platform hardware qfp active datapath infrastructure sw-nic** コマンドのサンプルのコマンド出力を示します。

```
Router# show platform hardware qfp act datapath infrastructure sw-nic
pmd b19811c0 device Gi1
  RX: pkts 418  bytes 37655 return 0 badlen 0
    pkts/burst 1  cycl/pkt 0  ext_cycl/pkt 0
    Total ring read 91995516, empty 91995113
  TX: pkts 355  bytes 57833
      pri-0: pkts 60  bytes 5590
             pkts/send 1
      pri-1: pkts 32  bytes 2616
             pkts/send 1
      pri-2: pkts 6  bytes 303
             pkts/send 1
      pri-3: pkts 38  bytes 6932
             pkts/send 1
      pri-4: pkts 176  bytes 39279
             pkts/send 1
      pri-5: pkts 25  bytes 1962
             pkts/send 1
      pri-6: pkts 8  bytes 459
             pkts/send 1
      pri-7: pkts 10  bytes 692
             pkts/send 1
  Total: pkts/send 1  cycl/pkt 3160
    send 343  sendnow 0
    forced 343  poll 0  thd_poll 0
    blocked 0  retries 0  mbuf alloc err 0
    TX Queue 0: full 0  current index 0  hiwater 0
    TX Queue 1: full 0  current index 0  hiwater 0
    TX Queue 2: full 0  current index 0  hiwater 0
    TX Queue 3: full 0  current index 0  hiwater 0
    TX Queue 4: full 0  current index 0  hiwater 0
    TX Queue 5: full 0  current index 0  hiwater 0
    TX Queue 6: full 0  current index 0  hiwater 0
    TX Queue 7: full 0  current index 0  hiwater 0
pmd b1717380 device Gi2
  RX: pkts 289216546  bytes 102405925473 return 0 badlen 0
    pkts/burst 7  cycl/pkt 326  ext_cycl/pkt 381
    Total ring read 141222555, empty 103047391
  TX: pkts 757922  bytes 260498122
      pri-0: pkts 94302  bytes 32428428
             pkts/send 1
      pri-1: pkts 95525  bytes 32791822
             pkts/send 1
      pri-2: pkts 93002  bytes 31950500
             pkts/send 1
      pri-3: pkts 96799  bytes 33381108
             pkts/send 1
      pri-4: pkts 90823  bytes 31179044
             pkts/send 1
```

```
      pri-5: pkts 97436  bytes 33455916
             pkts/send 1
      pri-6: pkts 93243  bytes 32113540
             pkts/send 1
      pri-7: pkts 96792  bytes 33197764
             pkts/send 1
  Total: pkts/send 1  cycl/pkt 760
    send 685135  sendnow 3
    forced 685117  poll 0  thd_poll 0
    blocked 0  retries 0  mbuf alloc err 0
    TX Queue 0: full 0  current index 0  hiwater 31
    TX Queue 1: full 0  current index 0  hiwater 31
    TX Queue 2: full 0  current index 0  hiwater 0
    TX Queue 3: full 0  current index 0  hiwater 0
    TX Queue 4: full 0  current index 1  hiwater 31
    TX Queue 5: full 0  current index 0  hiwater 0
    TX Queue 6: full 0  current index 0  hiwater 0
    TX Queue 7: full 0  current index 0  hiwater 0
pmd b14ad540 device Gi3
  RX: pkts 758108  bytes 302121148 return 0 badlen 0
    pkts/burst 1  cycl/pkt 572  ext_cycl/pkt 811
    Total ring read 78867251, empty 78155478
  TX: pkts 756904  bytes 301747138
      pri-0: pkts 9  bytes 540
             pkts/send 1
      pri-1: pkts 200064  bytes 80223776
             pkts/send 1
      pri-3: pkts 244086  bytes 97204792
             pkts/send 1
      pri-4: pkts 3  bytes 822
             pkts/send 1
      pri-5: pkts 250502  bytes 99404344
             pkts/send 1
      pri-7: pkts 62240  bytes 24912864
             pkts/send 1
  Total: pkts/send 1  cycl/pkt 737
    send 705364  sendnow 3
    forced 705355  poll 0  thd_poll 0
    blocked 0  retries 0  mbuf alloc err 0
    TX Queue 0: full 0  current index 0  hiwater 0
    TX Queue 1: full 0  current index 0  hiwater 31
    TX Queue 2: full 0  current index 0  hiwater 0
    TX Queue 3: full 0  current index 0  hiwater 31
    TX Queue 4: full 0  current index 0  hiwater 0
    TX Queue 5: full 0  current index 0  hiwater 0
    TX Queue 6: full 0  current index 0  hiwater 0
    TX Queue 7: full 0  current index 0  hiwater 0
```



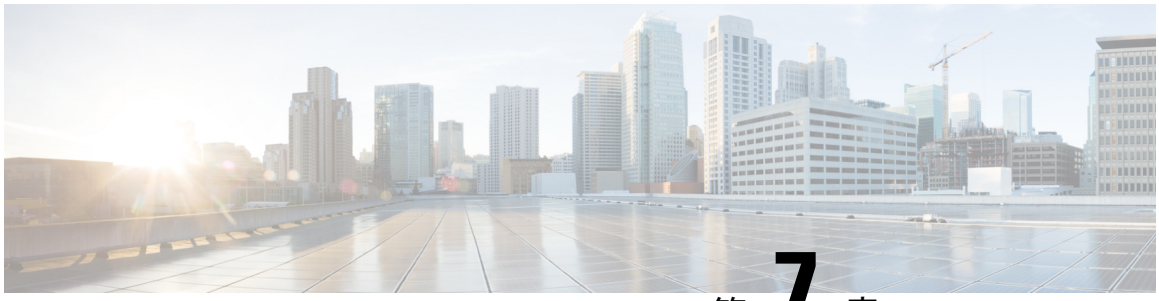
第 6 章

IPv6 機能の設定

インターネットプロトコルバージョン6（IPv6）は、ネットワークアドレスビット数を（IPv4の）32ビットから128ビットに拡張しているため、地球上のすべてのネットワークデバイスにグローバルに一意なIPアドレスを十分に提供できます。IPv6により実現する無制限のアドレス空間により、シスコは信頼性があり、ユーザエクスペリエンスとセキュリティが強化された新しいアプリケーションとサービスをより多く提供できます。

シスコソフトウェアでの基本的なIPv6接続の実装は、個々のデバイスインターフェイスへのIPv6アドレスの割り当てで構成されます。IPv6トラフィックの転送はグローバルに有効化でき、IPv6のCisco Express Forwardingスイッチングを有効にすることもできます。ユーザーは、ドメインネームシステム（DNS）の名前からアドレスおよびアドレスから名前のルックアッププロセスでAAAAレコードタイプのサポートを設定し、IPv6ネイバー探索を管理することで、基本接続の機能を拡張できます。

IPv6アドレス指定は、Amazon Web Servicesで実行されているCisco Catalyst 8000Vインスタンスでサポートされます。インスタンスのIPv6機能を設定する方法については、『IPv6 Addressing and Basic Connectivity Configuration Guide』を参照してください。



第 7 章

トランジットゲートウェイを使用したトランジット VPC の展開

トランジット ゲートウェイ ソリューションに関する情報

Amazon Virtual Private Cloud（Amazon VPC）を使用して、必要な数の仮想ネットワークを作成できます。AWS では、これらのネットワークを相互に接続したり、非 AWS インフラストラクチャ（オンプレミスのデータセンター、離れた場所にある本社、その他のオフィス）に接続したりするためのさまざまなオプションも提供しています。

トランジット VPC ソリューションを使用して Cisco Catalyst 8000V インスタンスを展開すると、Amazon VPC でハブアンドスポークトポロジを構築してエッジ接続を一元化できます。トランジット VPC では、VPC での共有サービスまたはパケットインスペクション/レプリケーションを導入できます。複数のアカウントにわたって機能し、AWS CloudFormation スタックを介して簡単に設定できます。ただし、このソリューションではトランジットゲートウェイではなく VPN ゲートウェイを使用するため、新しいスポークの追加にはある程度複雑な操作が伴います。

この制限を克服するため、トランジット ゲートウェイ ソリューションを使用して Cisco Catalyst 8000V トランジット VPC を展開できるようになりました。トランジットゲートウェイは、AWS クラウドとオンプレミスネットワークで VPC を相互接続するために AWS が提供する地域ネットワーク トランジット ハブ サービスです。トランジットゲートウェイを使用した Cisco Catalyst 8000V トランジット VPC ソリューションでは、スポーク側のトランジットゲートウェイを使用して、同じ地域内の全スポーク VPC 間の接続を可能にします。トランジットゲートウェイは、VPN 接続を使用してトランジット VPC の 2 つの Cisco Catalyst 8000V インスタンスに接続されます。Cisco Catalyst 8000V インスタンスは、さまざまなオンプレミス ブランチ ロケーションへの VPN 接続を提供します。

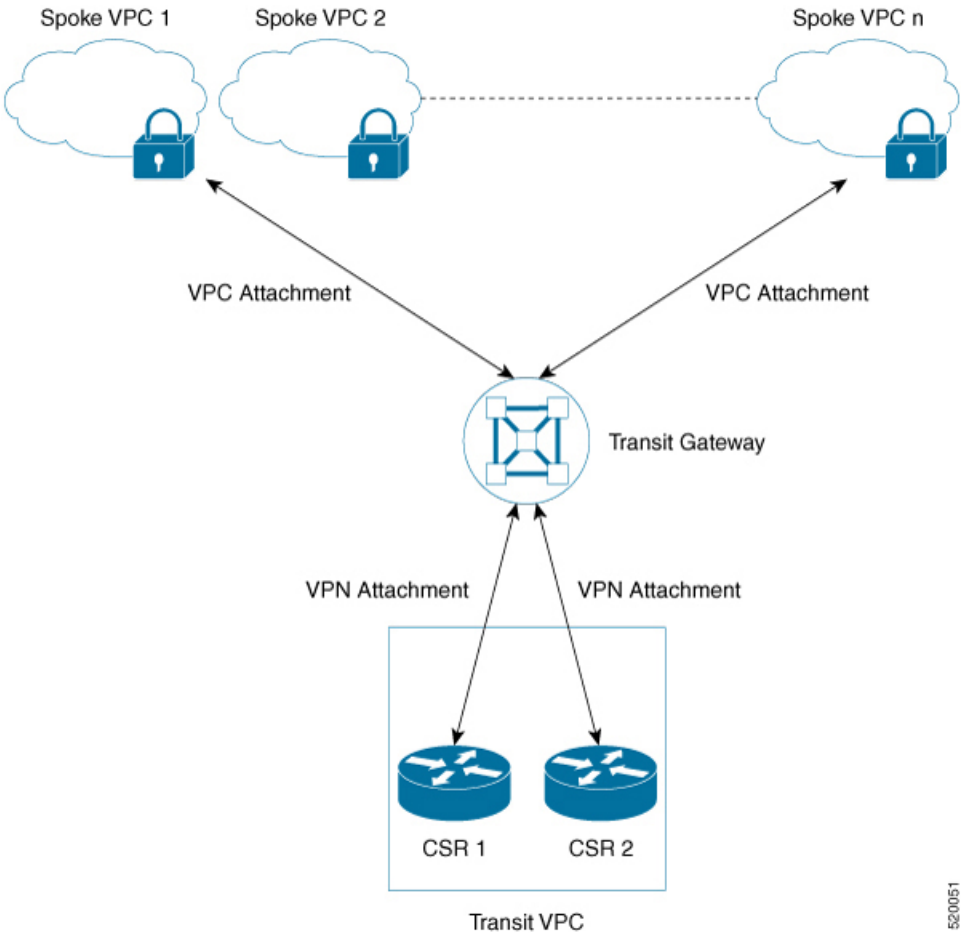
トランジット ゲートウェイ ソリューションを使用して AWS トランジット VPC を展開する方法を確認するには、この章で説明する設定手順を実行します。

トランジット VPC - トランジット ゲートウェイ コンポーネント

トランジット ゲートウェイ ソリューションには、スポーク間 VPC 接続を提供するためのハブとして機能するトランジットゲートウェイがあります。トランジット VPC は、スポーク VPC からリモートネットワークに流れるトラフィックの中央ハブとして機能するもう 1 つのコアコ

ンポーネントです。トランジット VPC は、VPN の終端とルーティングを可能にする 2 つの Cisco Catalyst 8000V インスタンスをホストします。

図 *3:* トランジット ゲートウェイ ソリューションのサンプルトポロジ



このソリューションでは、Solution Helper と Cisco Configurator という 2 つの AWS Lambda 関数を使用して、インスタンスとスポーク VPC 間の VPN 接続を自動的に設定します。

- **Solution Helper Lambda**：このコンポーネントは、cloudformation テンプレートを展開するとトリガーされます。このコンポーネントでは、トランジットゲートウェイ、Cisco Catalyst 8000V インスタンスとの VPN 接続、およびインスタンスとトランジットゲートウェイ間の VPN 接続が作成されます。その後、Lambda 関数は S3 SSE-KMS を使用して VPN 接続情報を Amazon S3 バケットに保存します。

- **Cisco Configurator Lambda**：S3 Put イベントによって Cisco Configurator Lambda 関数が呼び出されます。この関数により、VPN 接続情報が解析され、新しい VPN 接続を作成するために必要な設定ファイルが生成されます。Cisco Configurator Lambda は、SSH を使用して IOS 設定を Cisco Catalyst 8000V インスタンスにプッシュします。シスコの設定が Cisco Catalyst 8000V インスタンスに適用されると、即座に VPN トンネルが起動し、トランジットゲートウェイとの間にボーダー ゲートウェイ プロトコル（BGP）ネイバー関係が確立されます。

AWS トランジット ゲートウェイ ソリューションの利点

- トランジット ゲートウェイ ソリューションには、拡張性と復元力があります。
- トランジット ゲートウェイ ソリューションはマネージドサービスです。つまり、高可用性およびモニタリング機能が組み込まれており、CloudWatch などのメトリックを使用してソリューションを追跡できます。
- トランジット ゲートウェイ ソリューションを使用すると、ネットワークアーキテクチャの簡素化が可能になるため、運用コストの削減を実現できます。
- セキュリティを含めて、ソリューションを一元管理できます。

AWS トランジット ゲートウェイ ソリューションの前提条件

- Elastic IP、VPC、TGW、および VPN 接続に十分な制限が課されている必要があります。
- *cloudformation* サービスを管理する IAM 権限があることを確認します。

AWS トランジット ゲートウェイ ソリューションの制限事項

- 自動スケーリングは、このバージョンのソリューションではサポートされていません。
- このソリューションを展開した後、VPC 接続を使用して、スポーク VPC をトランジットゲートウェイに手動で追加する必要があります。

AWS トランジット ゲートウェイ ソリューションの設定

ステップ 1 Amazon Web Services Marketplace にログインします。

ステップ 2 **Cisco Catalyst 8000V – Transit Network VPC** テンプレートを検索して、このテンプレートを選択します。

ステップ 3 自分の所在地に該当する地域でテンプレートを起動します。[AWS Cloudformation Service] ページが表示されます。[Next] をクリックします。

ステップ 4 次の [Stack Details] を指定します。

パラメータ	説明
C8000V Throughput Requirements	Cisco Catalyst 8000V インスタンスに必要なスループット。この値により、起動するインスタンスタイプが決まります。デフォルト値は 2 x 500 Mbps です。
SSH Key to access C8000V	Cisco Catalyst 8000V インスタンスの起動後に、インスタンスへのセキュアな接続を可能にする公開/秘密キーペア。 公開/秘密キーペアを入力する必要があります。このキーペアは、AWS アカウントの作成時に、設定した地域で作成されます。
License Model	BYOL は、現在サポートされている唯一のライセンスモデルです。
Enable Termination Protection	Cisco Catalyst 8000V インスタンスの終了保護を有効にするには、このフィールドを有効にします。この機能により、偶発的な Cisco Catalyst 8000V の終了が防止されます。実稼働環境でこのフィールドを有効にすることを推奨します。デフォルトでは、このフィールドの値は [Yes] に設定されます。
Prefix for S3 Objects	Amazon S3 オブジェクトの作成時にプレフィックスとして使用する必要があるテキスト文字列。デフォルトの値は **vpnconfigs/** です。
Additional AWS Account ID	S3 バケットと AWS KMS カスタマーマスターキーへのアクセスを許可するトランジットネットワークに関連付けられた AWS アカウントのアカウント ID。 （注） このフィールドには、追加の AWS アカウント ID を 1 つだけ入力できます。複数の追加の AWS アカウントをトランジットネットワークに接続する場合は、追加のアカウントのアクセス許可を手動で設定する必要があります。
Transit VPC CIDR Block	トランジット VPC の CIDR ブロック。VPC とサブネット CIDR のアドレス範囲を変更して、ネットワークとのコリジョンを回避します。デフォルトの値は **100.64.127.224/27** です。

パラメータ	説明
1st Subnet Network	AZ1 で作成されたトランジット VPC サブネットの CIDR ブロック。デフォルトの値は **100.64.127.224/28** です。
2nd Subnet Network	AZ2 で作成されたトランジット VPC サブネットの CIDR ブロック。デフォルトの値は **100.64.127.240/28** です。
Transit VPC BGP ASN	トランジット VPC の BGP 自律システム番号（ASN）。デフォルトの値は **64512** です。
Spoke VPC Tag Name	トランジット VPC に接続するスポーク VPC の識別に使用するタグ。
Preferred VPN Endpoint Tag Name	トランジット VPC Cisco Catalyst 8000V インスタンスを通過するトラフィックフローを制御する優先 Cisco Catalyst 8000V VPN エンドポイントを設定するために使用するタグ。たとえば、ステートフルオンプレミス ファイアウォールと統合する場合に使用します。
Optional AZ configuration 1st Subnet	Public Subnet1 の可用性ゾーン番号。
Optional AZ configuration 2nd Subnet	Public Subnet2 の可用性ゾーン番号。

ステップ 5 設定を確認して確定します。AWS Identity and Access Management（IAM）によってリソースが作成され、CAPABILITY_AUTO_EXPAND 機能が必要になる可能性があることを承認するには、このチェックボックスをオンにします。

ステップ 6 [Create] をクリックして、スタックを展開します。
展開が成功すると、AWS Cloud Formation コンソールの [Status] 列に [CREATE_COMPLETE] と表示されます。

設定例

次に、トランジット ゲートウェイ ソリューションを使用して AWS トランジット VPC を展開する設定例を示します。

```
ip-100-64-127-234#sh run
Building configuration...

Current configuration : 7284 bytes
!
! Last configuration change at 14:10:57 UTC Thu Oct 10 2020
!
version 17.4
```

```
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console virtual
!
hostname ip-100-64-127-234
!
boot-start-marker
boot-end-marker
!
!
vrf definition GS
 rd 100:100
 !
 address-family ipv4
 exit-address-family
!
logging persistent size 1000000 filesize 8192 immediate
!
no aaa new-model
!
ip vrf vpn-0f56b2afc60b1d492
 rd 64525:1
 route-target export 64525:0
 route-target import 64525:0
!
ip vrf vpn0
 rd 64525:0
!
ip admission watch-list expiry-time 0
!
subscriber templating
!
multilink bundle-name authenticated
!
crypto pki trustpoint TP-self-signed-572041569
 enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-572041569
 revocation-check none
 rsakeypair TP-self-signed-572041569
!
!
crypto pki certificate chain TP-self-signed-572041569
 certificate self-signed 01
  3082032E 30820216 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  30312E30 2C060355 04031325 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 35373230 34313536 39301E17 0D313931 30313031 34303631
  355A170D 33303031 30313030 30303030 5A303031 2E302C06 03550403 1325494F
  532D5365 6C662D53 69676E65 642D4365 72746966 69636174 652D3537 32303431
  35363930 82012230 0D06092A 864886F7 0D010101 05000382 010F0030 82010A02
  82010100 A974EDB7 292BBB6A 09026F6A 381F7852 714775E3 E25F1F89 CED40FCB
  F45204F9 2F2F5FEE C46A9D16 A8D7307A C5433234 10D3F709 B4B18B3D 009B4A7A
  85980EEB 1282D1F7 C3CD4429 16042D4D 544315F4 E3ABA673 21E66C52 187AD1E6
  6B21F98A F0537D0A 8171618E 6CDF3B70 E2C8B553 8096C2D6 B4CD1AE4 B6DFD615
  844924B8 83DBE166 3CBC90F1 889CB00F 1644ECCE F2E70D81 CA35B555 D9757BE4
  34440FD9 D15580FA C50181CD D646AB6C 22F707A7 1D9F98CA 19897AF4 7488762B
  35ECA78F D2B249C7 8079255F 72BE5CF8 214B5135 E97B1104 A9CB449E A4A1D996
  9B99EC0E 18EF94FE FE73706A BF417262 12771D33 FF61A325 4479CAFB 10D0EEAA
  810E3437 02030100 01A35330 51300F06 03551D13 0101FF04 05300301 01FF301F
  0603551D 23041830 16801476 E85FEE9B EAE114A4 74C542FD E923856D 6F17F830
  1D060355 1D0E0416 041476E8 5FEE9BEA E114A474 C542FDE9 23856D6F 17F8300D
  06092A86 4886F70D 01010505 00038201 010043A6 03287F7E 1F13A7D4 26D661FE
```

```
  D11FED41 FE195D3E 6ADEA111 267C534B 266F587A 6A2F395D C50F5894 4C01F62B
  A179B852 F5F8ED62 DFF35587 3CFF352C 523F8D3D 8A786E61 A73EA8BB C8FC0A8D
  C2F0C260 0BB25D28 01B26B2B 27D71A31 2CE81DA5 6296D4AA 756A6658 0ADB89FB
  52BE1E9F A8BF17AA B2A0379A 1921AF64 834455CF B6307205 CE12C83A 2D29AEF2
  D79B79F7 9701F86E EB51B8E2 95BA7D5A C67A05F8 2AA7A8E0 3626D155 FC2D79EC
  9506D897 D79B8E65 A1D89F8A 6EC21FD1 15BFBD79 8A6FEB77 15C10DEE 0A50A7A5
  C8109573 9C58A869 D2740BC4 61D953F2 7AA92870 69BF035C 08DA0EFB B4AB9AC1
  BD4DB053 66ADD9E3 B5957D2B 8E467A91 258A
   quit
!
license udi pid CSR1000V sn 9YGGWBVUY3N
no license smart enable
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
username ec2-user privilege 15 secret 5 $1$Gf9p$OfANl/ujuCIvpunuRDwKi1
username automate privilege 15 secret 8
$8$g62y2e1pz0O4/n$M8DmVAM/G9yySvjbB1I2tBJAW4IWZRIc44Icent4bps
!
redundancy
!
crypto keyring keyring-vpn-0f56b2afc60b1d492-2
  local-address GigabitEthernet1
  pre-shared-key address 52.54.79.47 key lhvPlpTYxUTno.lNTbR25F9743HEguaH
crypto keyring keyring-vpn-0f56b2afc60b1d492-1
  local-address GigabitEthernet1
  pre-shared-key address 52.44.80.94 key Qq4fLolOMf1iW3d7gJhtzF8h8Tu3IlNT
!
crypto isakmp policy 200
 encr aes
 authentication pre-share
 group 2
 lifetime 28800
crypto isakmp keepalive 10 10 periodic
crypto isakmp profile isakmp-vpn-0f56b2afc60b1d492-1
   keyring keyring-vpn-0f56b2afc60b1d492-1
   match identity address 52.44.80.94 255.255.255.255
   local-address GigabitEthernet1
   rekey
crypto isakmp profile isakmp-vpn-0f56b2afc60b1d492-2
   keyring keyring-vpn-0f56b2afc60b1d492-2
   match identity address 52.54.79.47 255.255.255.255
   local-address GigabitEthernet1
   rekey
!
crypto ipsec security-association replay window-size 1024
!
crypto ipsec transform-set ipsec-prop-vpn-aws esp-aes esp-sha-hmac
 mode tunnel
crypto ipsec df-bit clear
no crypto ipsec nat-transparency udp-encapsulation
!
crypto ipsec profile ipsec-vpn-aws
 set transform-set ipsec-prop-vpn-aws
 set pfs group2
!
interface Tunnel1
 description vpn-0f56b2afc60b1d492 from TGW to cgw-00d8fbb76cc59295e for account
902347396780
 ip vrf forwarding vpn-0f56b2afc60b1d492
 ip address 169.254.185.70 255.255.255.252
 ip tcp adjust-mss 1387
 tunnel source GigabitEthernet1
```

```
 tunnel mode ipsec ipv4
 tunnel destination 52.44.80.94
 tunnel protection ipsec profile ipsec-vpn-aws
 ip virtual-reassembly
!
interface Tunnel2
 description vpn-0f56b2afc60b1d492 from TGW to cgw-00d8fbb76cc59295e for account
902347396780
 ip vrf forwarding vpn-0f56b2afc60b1d492
 ip address 169.254.232.90 255.255.255.252
 ip tcp adjust-mss 1387
 tunnel source GigabitEthernet1
 tunnel mode ipsec ipv4
 tunnel destination 52.54.79.47
 tunnel protection ipsec profile ipsec-vpn-aws
 ip virtual-reassembly
!
interface VirtualPortGroup0
 vrf forwarding GS
 ip address 192.168.35.101 255.255.255.0
 ip nat inside
 no mop enabled
 no mop sysid
!
interface GigabitEthernet1
 ip address 100.64.127.234 255.255.255.240
 ip nat outside
 negotiation auto
 no mop enabled
 no mop sysid
!
router bgp 64525
 bgp log-neighbor-changes
 !
 address-family ipv4 vrf vpn-0f56b2afc60b1d492
  neighbor 169.254.185.69 remote-as 64526
  neighbor 169.254.185.69 timers 10 30 30
  neighbor 169.254.185.69 activate
  neighbor 169.254.185.69 next-hop-self
  neighbor 169.254.185.69 default-originate
  neighbor 169.254.185.69 as-override
  neighbor 169.254.185.69 soft-reconfiguration inbound
  neighbor 169.254.232.89 remote-as 64526
  neighbor 169.254.232.89 timers 10 30 30
  neighbor 169.254.232.89 activate
  neighbor 169.254.232.89 next-hop-self
  neighbor 169.254.232.89 default-originate
  neighbor 169.254.232.89 as-override
  neighbor 169.254.232.89 soft-reconfiguration inbound
 exit-address-family
!
iox
ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 vrf GS overload
ip forward-protocol nd
ip tcp window-size 8192
ip http server
ip http authentication local
ip http secure-server
ip route 0.0.0.0 0.0.0.0 GigabitEthernet1 100.64.127.225
ip route vrf GS 0.0.0.0 0.0.0.0 GigabitEthernet1 100.64.127.225 global
!
ip ssh rsa keypair-name ssh-key
ip ssh version 2
ip ssh pubkey-chain
```

```
  username ec2-user
   key-hash ssh-rsa F1B0DF92FC2E25F7D98A01B99FCE5F13 ec2-user
  username automate
   key-hash ssh-rsa ED4B0757CE2AC22C89B28BE55EDE7691
ip ssh server algorithm authentication publickey
ip scp server enable
!
ip access-list standard GS_NAT_ACL
 permit 192.168.35.0 0.0.0.255
!
control-plane
!
line con 0
 stopbits 1
line vty 0 4
 login local
 transport input ssh
!
app-hosting appid guestshell
 app-vnic gateway1 virtualportgroup 0 guest-interface 0
  guest-ipaddress 192.168.35.102 netmask 255.255.255.0
 app-default-gateway 192.168.35.101 guest-interface 0
 name-server0 8.8.8.8
end
```

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。