



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

最終更新：2024年8月29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

## Full Cisco Trademarks with Software License ?

## 第 1 章

## はじめに 1

対象読者および適用範囲 1

機能の互換性 1

表記法 2

通信、サービス、およびその他の情報 3

マニュアルに関するフィードバック 4

トラブルシューティング 4

## 第 2 章

## Amazon Web Services での Cisco Catalyst 8000V エッジソフトウェアの概要 5

Amazon Web Services で Cisco Catalyst 8000V を実行するための展開オプション 5

ライセンス 6

ペイアズユーゴー ライセンシング 7

サポートされていない Cisco IOS XE テクノロジー 7

## 第 3 章

## AWS での Cisco Catalyst 8000V の展開 9

AWS でサポートされるインスタンスタイプ 9

注意事項とガイドライン 11

AWS で Cisco Catalyst 8000V を展開するための前提条件 12

AWS での Cisco Catalyst 8000V の展開に関する制約事項 12

Cisco Catalyst 8000V インスタンスの展開 12

Cisco Catalyst 8000V Marketplace オファァの選択 12

Web サイトを介したインスタンスの起動	13
EC2 コンソールを介したインスタンスの起動	13
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	15
SSH を使用したインスタンスへの接続	16
SSH キーペアの作成	16
暗号化された Elastic Block Storage を使用した AMI の作成	16

---

第 4 章	<b>ゲストシェルの有効化</b>	19
	ゲストシェルの有効化	19
	IAM インスタンスロールの作成	19
	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	21
	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	22
	ゲストシェルの例	23

---

第 5 章	<b>パブリッククラウド用 L2 拡張の設定</b>	27
	LISP レイヤ 2 拡張の設定	28
	LISP レイヤ 2 拡張の設定の前提条件	29
	LISP レイヤ 2 拡張の設定の制約事項	29
	LISP レイヤ 2 拡張の設定	29
	AWS での Cisco Catalyst 8000V インスタンスの作成	30
	サブネットの設定	31
	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	31
	AWS で実行されているインスタンスでの LISP xTR の設定	32
	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	34
	PMD マルチキューのサポート	35

---

第 6 章	<b>IPv6 機能の設定</b>	39
-------	-------------------	----

---

第 7 章	<b>AWS 移行ツールを使用した Cisco CSR1000V インスタンスの Cisco Catalyst 8000V への移行</b>	41
-------	--	----

移行の前提条件	42
制限事項と注意事項	42
Cisco Catalyst 8000V への CSR1000V インスタンスの移行	43
移行が成功したかどうかの確認	45
ロールバックの実行	45

---

**第 8 章**

<b>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</b>	<b>47</b>
AWS トランジット ゲートウェイ ソリューションの利点	49
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	49
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	49
AWS トランジット ゲートウェイ ソリューションの設定	49
設定例	51
AWS Transit Gateway ソリューションの削除	55

【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/](http://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。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2020–2024 Cisco Systems, Inc. All rights reserved.





# 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 \(1 ページ\)](#)
- [機能の互換性 \(1 ページ\)](#)
- [表記法 \(2 ページ\)](#)
- [通信、サービス、およびその他の情報 \(3 ページ\)](#)
- [マニュアルに関するフィードバック \(4 ページ\)](#)
- [トラブルシューティング \(4 ページ\)](#)

## 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 <b>Ctrl</b>	^ および <b>Ctrl</b> シンボルは、Ctrl キーを表します。たとえば、 <b>^D</b> または <b>Ctrl+D</b> というキーの組み合わせは、 <b>Ctrl</b> キーを押しながら <b>D</b> 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。
<i>string</i>	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 <b>public</b> 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
<b>ボールド</b>	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。
<b>bold screen</b>	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](#) 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。

### シスコバグ検索ツール

[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](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 \(5 ページ\)](#)
- [ライセンス \(6 ページ\)](#)
- [ペイアズユーゴー ライセンシング \(7 ページ\)](#)
- [サポートされていない Cisco IOS XE テクノロジー \(7 ページ\)](#)

## 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 SD-WAN & Router - PAYG - DNA Advantage
- Cisco Catalyst 8000V SD-WAN & Router - PAYG - DNA Essentials
- Cisco Catalyst 8000V for SD-WAN & Routing

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
データセンター相互接続	OTV、WCCPv2
MPLS	MPLS、EoMPLS、VRF、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。
- IP マルチキャストは、Amazon Transit Gateway ソリューションでのみ機能します。



## 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。

- [AWS でサポートされるインスタンスタイプ \(9 ページ\)](#)
- [AWS で Cisco Catalyst 8000V を展開するための前提条件 \(12 ページ\)](#)
- [AWS での Cisco Catalyst 8000V の展開に関する制約事項 \(12 ページ\)](#)
- [Cisco Catalyst 8000V インスタンスの展開 \(12 ページ\)](#)

## AWS でサポートされ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。

リリース番号	サポートされているインスタンスタイプ
Cisco IOS XE 17.13.1a	<ul style="list-style-type: none"><li>• t3.medium</li><li>• c5.9xlarge、c5.2xlarge、c5.xlarge、c5.large</li><li>• c5n.18xlarge、c5n.4xlarge</li><li>• c6in.8xlarge、c6in.2xlarge、c6in.xlarge、c6in.large</li></ul>

リリース番号	サポートされているインスタンスタイプ
Cisco IOS XE 17.12.2、 Cisco IOS XE 17.12.1	<ul style="list-style-type: none"> <li>• t3.medium</li> <li>• c5.9xlarge、c5.2xlarge、c5.xlarge、c5.large</li> <li>• c5n.18xlarge、c5n.9xlarge、c5n.4xlarge、c5n.2xlarge、c5n.xlarge、c5n.large</li> </ul>
Cisco IOS XE 17.11.1a	<ul style="list-style-type: none"> <li>• t3.medium</li> <li>• c5.9xlarge、c5.2xlarge、c5.xlarge、c5.large</li> <li>• c5n.18xlarge、c5n.9xlarge、c5n.4xlarge、c5n.2xlarge、c5n.xlarge、c5n.large</li> </ul>
Cisco IOS XE 17.10.1a	<ul style="list-style-type: none"> <li>• t3.medium</li> <li>• c5.9xlarge、c5.2xlarge、c5.xlarge、c5.large</li> <li>• c5n.18xlarge、c5n.9xlarge、c5n.4xlarge、c5n.2xlarge、c5n.xlarge、c5n.large</li> </ul>
Cisco IOS XE 17.9.4a Cisco IOS XE 17.9.4 Cisco IOS XE 17.9.3a Cisco IOS XE 17.9.2a Cisco IOS XE 17.9.1a	<ul style="list-style-type: none"> <li>• t3.medium</li> <li>• c5.9xlarge、c5.4xlarge、c5.2xlarge、c5.xlarge、c5.large</li> <li>• c5n.18xlarge、c5n.9xlarge、c5n.4xlarge、c5n.2xlarge、c5n.xlarge、c5n.large</li> </ul>
Cisco IOS XE 17.8.1a	<ul style="list-style-type: none"> <li>• t3.medium</li> <li>• c5.9xlarge、c5.4xlarge、c5.2xlarge、c5.xlarge、c5.large</li> <li>• c5n.9xlarge、c5n.4xlarge、c5n.2xlarge、c5n.xlarge、c5n.large</li> </ul>
Cisco IOS XE 17.7.2 Cisco IOS XE 17.7.1a	<ul style="list-style-type: none"> <li>• t3.medium</li> <li>• c5.9xlarge、c5.4xlarge、c5.2xlarge、c5.xlarge、c5.large</li> <li>• c5n.9xlarge、c5n.4xlarge、c5n.2xlarge、c5n.xlarge、c5n.large</li> </ul>



リリース番号	サポートされているインスタンスタイプ
Cisco IOS XE 17.6.6a Cisco IOS XE 17.6.6 Cisco IOS XE 17.6.5a Cisco IOS XE 17.6.5 Cisco IOS XE 17.6.4 Cisco IOS XE 17.6.3a Cisco IOS XE 17.6.2 Cisco IOS XE 17.6.1a	<ul style="list-style-type: none"> <li>• t3.medium</li> <li>• c5.9xlarge、c5.4xlarge、c5.2xlarge、c5.xlarge、c5.large</li> <li>• c5n.9xlarge、c5n.4xlarge、c5n.2xlarge、c5n.xlarge、c5n.large</li> </ul>
Cisco IOS XE 17.5.1a	<ul style="list-style-type: none"> <li>• t3.medium、t2.medium</li> <li>• c4.8xlarge、c4.4xlarge、c4.2xlarge、c4.xlarge、c4.large</li> <li>• c5.9xlarge、c5.4xlarge、c5.2xlarge、c5.xlarge、c5.large</li> <li>• c5n.9xlarge、c5n.4xlarge、c5n.2xlarge、c5n.xlarge、c5n.large</li> </ul>
Cisco IOS XE 17.4.2 Cisco IOS XE 17.4.1b Cisco IOS XE 17.4.1a	<ul style="list-style-type: none"> <li>• t3.medium、t2.medium</li> <li>• c4.8xlarge、c4.4xlarge、c4.2xlarge、c4.xlarge、c4.large</li> <li>• c5.9xlarge、c5.4xlarge、c5.2xlarge、c5.xlarge、c5.large</li> <li>• c5n.9xlarge、c5n.4xlarge、c5n.2xlarge、c5n.xlarge、c5n.large</li> </ul>

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。

## 注意事項とガイドライン

- 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 \(35 ページ\)](#) 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。
- c5n.large、c5n.xlarge、c5n.2xlarge、c5n.9xlarge インスタンスタイプは、17.13.1a リリースからそれぞれ c6in.large、c6in.xlarge、c6in.2xlarge、c6in.8xlarge に置き換えられます。ただし、以前のリリースから Cisco IOS XE 17.13.1a にアップグレードすると、引き続き c5n イ

インスタンスタイプが表示されます。インスタンスタイプは、対応する c6in の代替に手動でアップグレードできます。

## 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）、または EC2 インスタンスコンソールへのアクセスを用意します。
- 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 オファターの選択

**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。次のオファターが表示されます。

- 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。

## Web サイトを介したインスタンスの起動

### 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、利用規約に同意して [Continue to Subscribe] をクリックします。

**ステップ 2** [Configure Software] ウィンドウで、Cisco Catalyst 8000V インスタンスの [Software Version] と [Region] を選択します。

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 3** [Fulfillment Option] フィールドで、[Amazon Machine Image] を選択します。

**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 from Website] を選択します。

**ステップ 6** ドロップダウンリストから [EC2 Instance Type] を選択します。

**ステップ 7** インスタンスの [VPC]、[Subnet]、[Security Group]、[Key Pair] を設定します。

AWS におけるこれらの設定の詳細については、「[Parameters for instance configuration](#)」 [英語] を参照してください。

**ステップ 8** [Launch] をクリックします。

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。

## EC2 コンソールを介したインスタンスの起動

AWS Marketplace にログインして Cisco Catalyst 8000V に登録したら、この手順を実行します。この手順では、EC2 コンソールを介してインスタンスを起動する方法について説明します。

- ステップ 1** AWS Marketplace から Cisco Catalyst 8000V オファーを選択したら、利用規約に同意して [Continue to Subscribe] をクリックします。
- ステップ 2** [Configure Software] ウィンドウで、Cisco Catalyst 8000V インスタンスの [Software Version] と [Region] を選択します。
- 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 3** [Fulfillment Option] フィールドで、[Amazon Machine Image] を選択します。
- ステップ 4** [Continue to Launch] をクリックし、[Launch with E2 Console] を選択します。
- ステップ 5** [Launch an Instance] ウィンドウで、[Name] フィールドにインスタンスの名前を入力します。
- ステップ 6** [Application and OS Images] エリアには、サブスクリプションと選択したソフトウェアバージョンに基づいて AMI が自動入力されます。
- ステップ 7** [Instance Type] フィールドで、ドロップダウンリストからサポートされているインスタンスタイプを選択します。各 IOS XE リリースでサポートされているインスタンスタイプの詳細については、「[AWS でサポートされるインスタンスタイプ \(9 ページ\)](#)」を参照してください。
- ステップ 8** インスタンスの [Key Pair] を設定します。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。新しいキーペアを作成するには、[Create Key Pair] をクリックし、キーペア名を入力してから [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ないことに注意してください。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いて `chmod 400 pem-file-name` コマンドを実行します。
- Cisco IOS XE 17.10.1a 以降を搭載した Cisco Catalyst 8000V 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。
- ステップ 9** インスタンスの [Network] を設定します。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) デフォルトでは、1つのインターフェイスが設定されます。[Network Interfaces] の [Instance Details] エリアから追加のインターフェイスを作成できます。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。
- ステップ 10** [Security Group] を設定します。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できます。Cisco Catalyst 8000V ではコンソールアクセスに SSH が必要です。また Cisco Catalyst 8000V では、セキュリティグループが少なくとも TCP/22 をブロックしないようにする必要があります。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。

- ステップ 11** [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 および 17.5.x リリースでは、バージョン 1 (V1) のみが適用可能です。Cisco IOS XE 17.6.1 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。
- ステップ 12** インスタンスの [Storage] を設定します。デフォルトのハードドライブ設定を保持します。仮想ハードドライブのサイズを変更できないことに注意してください。
- ステップ 13** [User Data] フィールドを使用して、カスタムデータ形式で Day 0 の設定データかブートストラッププロパティを指定します。サポートされているカスタムデータ形式については、「[Day 0 Configuration](#)」を参照してください。
- ステップ 14** [Review and Launch] をクリックします。
- ステップ 15** Cisco Catalyst 8000V インスタンスの情報を確認して、[Launch Instance] をクリックします。
- インスタンスが起動すると、[Instances] ページの上部に成功メッセージが表示されます。[Instances] リストで新たに起動したインスタンスを確認することもできます。新たに起動したインスタンスをクリックしてインスタンスにアクセスします。

## 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 アドレスと Cisco Catalyst 8000V インスタンスを関連付けます。

- 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** 表示された [Network interfaces] ウィンドウで、[eth0] をクリックします。
- 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 3** [Interface ID Value] をクリックします。
- ステップ 4** [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 5** [Elastic IP address] ドロップダウン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 6** 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 7** 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 8** [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] の順に選択してインスタンスのリストを表示します。
- 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。
- 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。
  - [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。
  - [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。
  - [Create] をクリックします。
  - [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。
  - [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] フィールドが表示されます。

**Copy AMI** ×

AMI ami-8feaf0e6 will be copied to a new AMI. Set the new AMI settings below.

Destination region\* US East (N. Virginia) ▾

Name RoadTripBlogServer\_2014\_04\_23

Description Copy of RoadTripBlogServer\_2014\_04\_23

Encryption  Encrypt target EBS snapshots ⓘ

Master Key (default) aws/ebs ▾ ⓘ

**Key Details**

Description	Default master key that protects my EBS volumes when no other key is defined
Account	This account ( )
KMS Key ID	6cfb2f97-4972-4f85-b3e2-c040ea97fb38
KMS Key ARN	arn:aws:kms:us-east-1: :key/6cfb2f97-4972-4f85-b3e2-c040ea97fb38

Cancel Copy AMI

396321

ステップ 6 [Destination region] ドロップダウンリストから宛先 ([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 Amazon 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 \(19 ページ\)](#)
- [IAM インスタンスロールの作成 \(19 ページ\)](#)
- [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 \(21 ページ\)](#)
- [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 \(22 ページ\)](#)
- [ゲストシェルの例 \(23 ページ\)](#)

## 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。

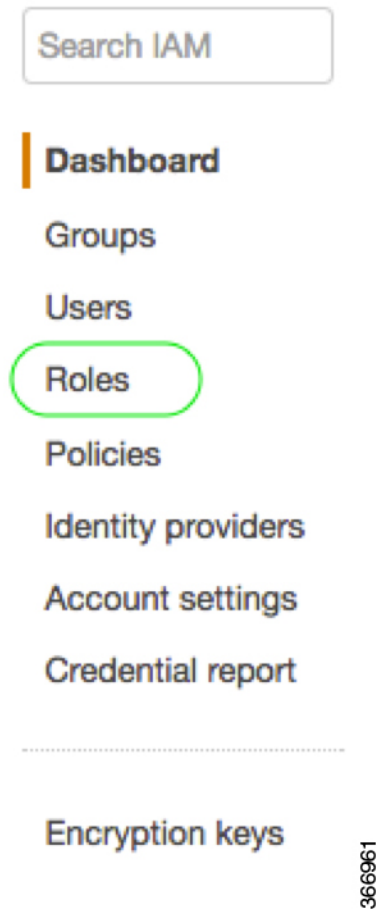
これらのタスクを実行する方法の詳細については、「Cisco Catalyst 8000V への IAM インスタンスロールの割り当て」と「新しい Cisco Catalyst 8000V への IAM インスタンスロールの割り当て」を参照してください。

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

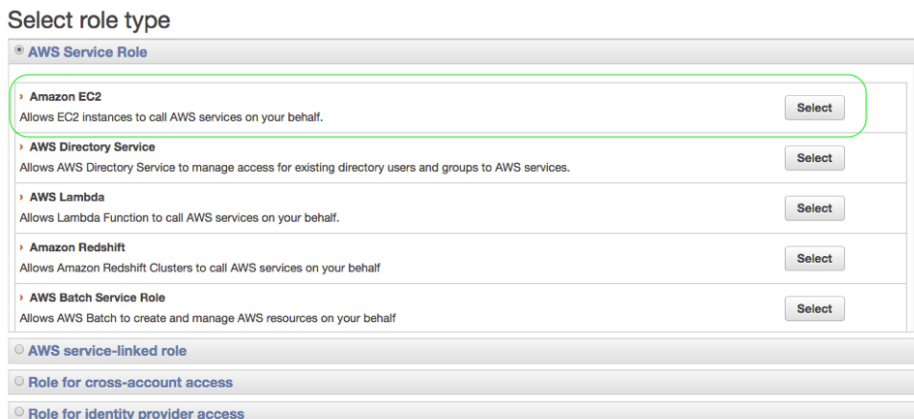
1. IAM ロールを作成する権限を持つ管理者として、AWS にサインインします。
2. [EC2] をクリックして、EC2 コンソールを開始します。
3. [IAM] をクリックして、IAM コンソールを開始します。
4. [Roles] をクリックします。

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



366960

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、指定しておくことで、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。

- [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。
- [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/38/6f2b06f0f00e6b381ac4741d0b97eb2e3fa232382dfe31577154a207c/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/45cdefb4d119e963163117e6c570e08029902f2ee2c143c7a0a5cc5/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/e6af338b04e96cefd690f741d7db24547ff1f94240c997a26fa808b/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/a6ae9591a619918cd550234cd151949c643d5f5b3f4fd7ee0c6e8/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/eb/8127f1b0294f044121d20aac7785fcb810e159098447967a6103bdfb96/rsa-4.5-py2.py3-none-any.whl
Collecting boto3==1.18.16 (from awscli)
  Downloading
https://files.pythonhosted.org/packages/2d/9e/afa41d0c91186808b7839d021e67e23c87b317aa4662d3f3cf/boto3-1.18.16-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/a94e8635fa3e4cfc75060035481a2447ae76ff5ca5392970fa3053f/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
boto3==1.18.16->awscli)
  Downloading
https://files.pythonhosted.org/packages/9f/f0/a391d1463d1b233795ca0f0ef383bb442339e68894702619e6987/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/41/70/3045c31146e875862207e80709e0b306af2b5f134d7615b/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/d5/5f01272b5fab231c9e0acc0448aaf5c8621770920e3469c6e0139/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/48de5c0f13094d729fe40316a2a247463ff1c528248a4b04078a/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/c72726166f80988223ef48f5d7fa2cf8809525a1199161281cd080a/csr_aws_guestshell-0.0.17.dev.tar.gz
Collecting awscli (from csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/23/1b/265d14e18a8b2341375991cc22021233f3c3f1d1099d0170ce/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/c0578c27298029e445e472a19190e20b182d1b1662ec7f2ca1d0c523/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/c965c3981e689c933c65a27a695afcf758850a7994ad3ac6599a8a/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/f8/8127fcb0294f044121d20aac7785feb610e159098447967a6103eddf96/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/96/35fa364675cf17e3a190ae08716c4078ca86a2ef071d32c886c52c/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.pythhosted.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.pythhosted.org/packages/69/79/e6af3380de96ef0b60f741d7db2457ff1f9424099a26fa8083/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.pythhosted.org/packages/22/cd/a6a959da619918cd550234db15194c644c5553f4ff7ee0c6e8/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.pythhosted.org/packages/c9/db/45cdef1b4d119e963163117e6c5708a0802992b2fee2c1437a05cc5/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.pythhosted.org/packages/07/db/5f00127b6fa23c1c9e0acc0438eaf5c862170920e3469660139/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.pythhosted.org/packages/62/1e/a4a8635fa3e4cfc7f5060035480e2447ae76f65ca53932970fe3053f/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.pythhosted.org/packages/56/aa/4ef5a67a9a62505b1245d526233cd1d153462d8f899fa4ef582/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.pythhosted.org/packages/d4/70/604503b08e87862407ae897090d03062ae5d13d7861db/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.pythhosted.org/packages/ee/ff/48ae50f01309d729fe40316a2a2477403ff152b92e84d04078a/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 プレフィックスを登録する 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) にアダプタイズすることで、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 \(28 ページ\)](#)
- [LISP レイヤ 2 拡張の設定の前提条件 \(29 ページ\)](#)
- [LISP レイヤ 2 拡張の設定の制約事項 \(29 ページ\)](#)
- [LISP レイヤ 2 拡張の設定 \(29 ページ\)](#)
- [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 \(34 ページ\)](#)
- [PMD マルチキューのサポート \(35 ページ\)](#)

## 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。
- Default Subnet : 10.0.0.0/24 (パブリック IP にマッピングされる)。
  - 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。
- Name : SSH アクセス
  - TCP Port 22 traffic : インバウンド許可
  - 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 LISPO
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的な 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

# AWS 移行ツールを使用した Cisco CSR1000V インスタンスの Cisco Catalyst 8000V への移行

AWS に Cisco CSR1000V インスタンスを展開しているユーザーが Cisco Catalyst 8000V にアップグレードする場合は、インスタンスを Cisco Catalyst 8000V に移行する必要があります。現在のところ、新規に Cisco Catalyst 8000V のインストールを実行し、Cisco CSR1000V から構成ファイルをコピーすることで、Cisco CSR1000V から Cisco Catalyst 8000V にインスタンスを移行できます。

Cisco IOS XE 17.14.1a リリース以降では、自動化機能を使用して作業を簡素化し、移行プロセスをシームレスにするために、AWS で展開可能な CloudFormation テンプレートとして AWS 移行ツールが用意されています。このワンクリックツールを使用すると、既存の Cisco CSR1000V 環境から新しい Cisco Catalyst 8000V 環境にユーザーを自動的に移行できます。この移行により、セキュアなオブジェクトストレージなど、Cisco Catalyst 8000V に搭載されている高度な機能を利用できるようになります。

このツールは、自律型（非 SDWAN）PAYG および BYOL Cisco CSR1000V 環境で使用できます。

このツールを使用して Cisco CSR1000V インスタンスを移行するには、AWS Marketplace にログインし、CloudFormation テンプレートを選択します。次のステップとして、移行する必要がある CSR1000V インスタンスのリスト、移行先の Cisco IOS XE バージョンなどの追加情報を入力します。入力内容を送信し、移行プロセスを開始します。

ツールはすべての入力を受け取り、次の2つの AWS Lambda 関数を作成します。これらの関数は、Lambda でコードを実行するために呼び出すことができる AWS リソースです。

- **Trigger Lambda** : この Lambda 関数は、移行ワークフローをトリガーします。この Lambda 関数は、移行のすべての前提条件が満たされているかどうかを検証します。さらに、このコンポーネントは、Worker Lambda を呼び出す前に、移行対象として指定した各 CSR1000V インスタンスも検証します。
- **Worker Lambda** : この Lambda 関数は、Trigger Lambda からの要求を処理します。同じクラウド設定で、移行するすべての CSR1000V インスタンスに対して同一の Cisco Catalyst

8000V インスタンスを起動します。移行するすべての CSR1000V インスタンスに対して、ツールは 1 つの Worker Lambda を生成します。

ワークフローが処理されると、移行が完了します。Cisco Catalyst 8000V インスタンスは AWS アカウントで使用できるようになります。

- [移行の前提条件 \(42 ページ\)](#)
- [制限事項と注意事項 \(42 ページ\)](#)
- [Cisco Catalyst 8000V への CSR1000V インスタンスの移行 \(43 ページ\)](#)
- [移行が成功したかどうかの確認 \(45 ページ\)](#)
- [ロールバックの実行 \(45 ページ\)](#)

## 移行の前提条件

- 移行を開始する前に、適切なライセンスで Cisco Catalyst 8000V に登録する必要があります。移行ツールは、マーケットプレイスの C8000V 製品リストの AMI を使用します。このツールが AWS アカウントから AMI を取得するには、登録が必要です。
- Cisco CSR1000V インスタンスのユーザー名とパスワードを使えるようにしておく必要があります。自動化機能はユーザーの PEM ファイルにアクセスできないため、これらのログイン情報は、Cisco CSR1000V インスタンスにアクセスするための入力値としてツールで使用されます。移行を開始する前に、`username [user name] privilege 15 password [password]` 設定を実行します。
- Cisco CSR 1000V インスタンスには、パブリック IP アドレスが必要です。AWS 移行ツールがデバイスにログインできるように、パブリック IP アドレスはパブリックインターネット経由で接続する必要があります。
- 使用中の Cisco CSR1000V インスタンスが自律モードで実行状態になっている必要があります。

## 制限事項と注意事項

- 移行できるクラウドプロパティは、サブネット、AZ、セキュリティグループ、セカンダリ IP、Elastic IP、タグ (複数のタグ) のみです。
- この移行は、t3、c5n、および c5 インスタンスタイプの Cisco CSR1000V インスタンスに対してのみ実行できます。選択したインスタンスタイプとサイズが Cisco CSR 1000V と Cisco Catalyst 8000V の両方でサポートされている必要があります。
- あるライセンスタイプから別のライセンスタイプへの移行はサポートされていません。たとえば、PayG ライセンスタイプのインスタンスを BYOL ライセンスタイプに移行することはできません。



- この移行は、自律モードで実行される Cisco CSR1000V インスタンスに対してのみ実行できます。AWS 移行ツールは、SD-WAN モードで展開された Cisco CSR1000V インスタンスでは機能しません。
- リリース時に、他のログ、構成ファイル、パッケージは移行されません。コントローラモードで Cisco CSR1000V インスタンスを指定すると、移行は失敗します。
- この移行では、次のデイズロ設定コマンドはサポートされません。
  - 'Building.\*?bytes\n',
  - 'hostname.\*?\n',
  - '\ncrypto pki.\*?quit',
  - 'no aaa new-model\n',
  - 'login local\n',
  - 'spanning-tree extend system-id\n',
  - 'no mop enabled\n',
  - 'no mop sysid'

Cisco Catalyst 8000V でサポートされていない Cisco CSR1000V の IOS 設定は、Cisco Catalyst 8000V インスタンスには適用されません。このシナリオで移行が失敗することはありませんが、Cisco Catalyst 8000V の設定に不整合が生じる可能性があります。

- 障害のリスクを最小限に抑えるために、1回の CloudFormation 展開で同時に移行する Cisco CSR1000V インスタンスを 10 個までにすることを推奨します。
- 実行中の現在の IOS 設定のみが移行されます。

## Cisco Catalyst 8000V への CSR1000V インスタンスの移行

**ステップ 1** AWS Marketplace にログインします。

**ステップ 2** 次のいずれかをクリックして、該当する Cisco CSR1000V から Cisco Catalyst 8000V への移行ツールテンプレートを選択します。

- [BYOL](#)
- [PAYG Network-Advantage](#)
- [PAYG Network-Essentials](#)

**ステップ 3** テンプレートがすでに用意されている場合は、[Create Stack] ウィンドウが表示されます。このページでは詳細情報を入力する必要はありません。[Next] をクリックして次の画面に移動します。

- ステップ 4** [Specify Stack Details] ウィンドウの [Stack Name] フィールドで、スタックの名前を指定します。[Name] フィールドでは、アルファベット A～Z、a～z、数字 0～9、およびダッシュを使用できます。[Stack Name] フィールドでは先頭にアルファベットを使用し、128 文字以内で指定する必要があります。
- ステップ 5** [Parameters] 領域で、移行先のバージョンを選択します。
- ステップ 6** [CSR1000V Instance IDs] ドロップダウンリストから、移行するインスタンスを選択します。フィールドには、現在のすべての EC2 インスタンスが一覧表示されます。同時に移行する Cisco CSR1000V インスタンスを最大 10 個選択します。移行に成功すると、同じ数の Cisco Catalyst 8000V インスタンスが作成されます。
- ステップ 7** (任意) [C8000V BYOL Token] フィールドにトークンを入力すると、Cisco Catalyst 8000V インスタンスにトークンが登録されます。このトークンは、Cisco Catalyst 8000V が適切なライセンスレベルで起動するために必要です。このトークンは、Cisco Catalyst 8000V が CSSM サーバーとの信頼を確立し、Cisco Catalyst 8000V に登録されているターゲットライセンスを取得するために必要です。このフィールドは、BYOL Cisco CSR1000V インスタンスを移行する場合にのみ入力する必要があります。このトークンを指定しなくても移行は完了しますが、Cisco Catalyst 8000V にライセンスは登録されません。
- ステップ 8** [SSH Key to Associate With C8000V] フィールドで、[SSH Key] を入力します。Cisco Catalyst 8000V インスタンスの作成後、このキーを使用してインスタンスにログします。
- ステップ 9** (任意) 電子メールや SMS で障害通知を受信する場合は、[SNS Topic ARN] フィールドで SNS ARN を指定します。
- (注) SRS ARN を指定する前に、Amazon Simple Notification Service を有効にして、ARN を使用できるようにしておく必要があります。
- ステップ 10** [Username to SSH Into Device] フィールドに、移行用に CSR1000V インスタンスで作成したユーザー名を入力します。
- ステップ 11** [Password to SSH Into Device] フィールドに、移行用に CSR1000V インスタンスで作成したパスワードを入力します。
- この SSH ユーザー名とパスワードにより、CSR1000V インスタンスへの SSH アクセスが有効になります。移行が完了すると、同じ SSH ログイン情報を使用して Cisco Catalyst 8000V にログインします。
- ステップ 12** 移行が失敗した場合にスタックでリソースをロールバックし、Cisco CSR1000V インスタンスを再展開するには、[Rollback Option] ドロップダウンリストで [Yes] を選択します。
- ステップ 13** 移行が失敗した場合に Cisco Catalyst 8000V インスタンスを終了するには、[Terminate C8KV Option] ドロップダウンリストで [Yes] を選択します。
- ステップ 14** 移行が成功した後に CSR1000V インスタンスを自動的に終了する場合は、[Terminate CSR Option] ドロップダウンリストで [Yes] を選択します。自動的に終了しない場合は、[No] を選択します。
- ステップ 15** [Next] をクリックします。
- ステップ 16** (任意) [Stack Failure Options] 領域の [Select Rollback All Stack Resources] フィールドで、[Rollback to Last Known Stable State] オプションボタンをクリックします。
- ステップ 17** [Next] をクリックします。
- ステップ 18** [Review and Create] ウィンドウで、すべての設定を確認します。このページの下部で、チェックボックスをオンにすることで、この移行で作成されるリソースを承認します。

ステップ 19 [Submit] をクリックして移行を開始します。

## 移行が成功したかどうかの確認

このタスクでは、Cisco CSR1000V から Cisco Catalyst 8000V へのインスタンスの移行が成功したかどうかを確認する手順を示します。

### 始める前に

移行を開始した後、次の確認手順を実行します。

- ステップ 1** スタックを送信すると、[CloudFormation Stack] ウィンドウが表示されます。このページで、スタックイベントの作成と完了を確認します。移行ツールのイベントにより、Cisco CSR1000V インスタンスを検証して、同一の Cisco Catalyst 8000V インスタンスを作成する Lambda 関数がトリガーされます。スタックの CREATE\_COMPLETE は、スタックリソースまたは Lambda 関数が正常に作成されたことを示します。
- ステップ 2** 移行対象の各リソースやログを表示するには、[CloudWatch]>[Log Groups] に移動します。スタック名がこのウィンドウに表示されます。スタック名をクリックすると、このスタックに関連する詳細とログがすべて表示されます。
- ステップ 3** Cisco Catalyst 8000V インスタンスが作成されているかどうかを確認します。Cisco Catalyst 8000V インスタンスには、CSR1000V インスタンスと同じ名前が指定されます。ただし、インスタンス ID は、CSR1000V インスタンスと対応する Cisco Catalyst 8000V インスタンスで異なります。

(注) Cisco Catalyst 8000V の展開中に移行が失敗した場合、Cisco Catalyst 8000V コンソールログが移行 S3 バケットに保存されます。Cisco Catalyst 8000V インスタンスを終了せずに、新しい Cisco Catalyst 8000V デバイスで失敗した移行をデバッグすることもできます。

## ロールバックの実行

移行ワークフローが成功しなかった場合は、ロールバックプロセスが自動的に実行されます。ただし、[Submit] をクリックして移行を開始する前に、[Rollback Option] ドロップダウンリストで [Yes] を選択しておく必要があります。

[Rollback Option] ドロップダウンリストで [No] を選択した場合、移行は停止し、Cisco CSR1000V インスタンスは実行状態に復元されません。





## 第 8 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。

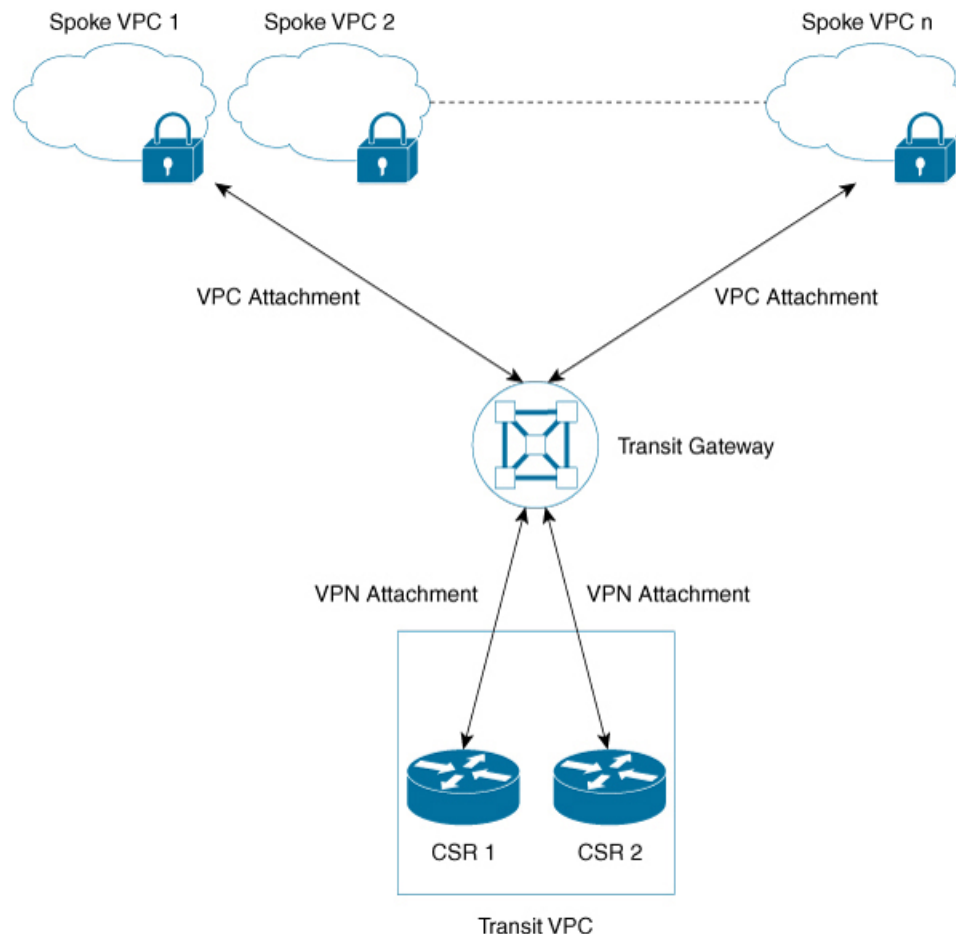
トランジット ゲートウェイ ソリューションを使用して AWS トランジット VPC を展開する方法を確認するには、この章で説明する設定手順を実行します。

### 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

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 \(49 ページ\)](#)
- [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 \(49 ページ\)](#)
- [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 \(49 ページ\)](#)
- [AWS トランジットゲートウェイ ソリューションの設定 \(49 ページ\)](#)
- [設定例 \(51 ページ\)](#)
- [AWS Transit Gateway ソリューションの削除 \(55 ページ\)](#)

## 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 <b>vpnconfigs/</b> 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



パラメータ	説明
	クとのコリジョンを回避します。デフォルトの値は <b>100.64.127.224/27</b> です。
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 <b>100.64.127.224/28</b> 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 <b>100.64.127.240/28</b> 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 <b>64512</b> 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 $!$Gf9p$OfANl/ujuCIvpunuRDwKil
username automate privilege 15 secret 8
$8$g62y2elpz004/n$M8DmVAM/G9yySvjbBlI2tBJAW4IWZRIc44Icent4bps
!
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 Qq4fLolOMfliW3d7gJhtzF8h8Tu3I1NT
!
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 Tunnell
  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
```

## AWS Transit Gateway ソリューションの削除

AWS Transit Gateway ソリューションを削除するには、次の手順を実行します。

**ステップ 1** [CloudFormation] ページに移動します。

**ステップ 2** 削除するスタックをクリックし、[Delete] をクリックします。

削除操作が開始されます。ただし、一部のリソースは手動で削除する必要があるため、[Deletion Failed] ステータスが表示される場合があります。CloudFormation の外部で作成されたリソースは、[Delete] をクリックしても削除されません。このシナリオでは、次のようなリソースを手動で削除する必要があります。

- サイト間 VPN 接続の削除 (c8000v-tgw から tgw-xxxxxx を削除)
- カスタマーゲートウェイの削除 (Tranist VPC エンドポイント 1 および 2)
- トランジットゲートウェイ接続の削除
- トランジットゲートウェイの削除 (c8000v-tgw)

(注) それでもソリューションが正常に削除されない場合は、Cisco Catalyst 8000V インスタンスで終了保護が有効になっているかどうかを確認してください。終了保護が有効になっている場合は、無効にしてから、Transit Gateway ソリューションを削除してみてください。



## 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。