AWS を使用して Cisco 仮想化 APIC を展開します

改訂：2023 年 7 月 18 日,

新機能および変更された機能に関する情報

次の表は、この最新リリースまでの主な変更点の概要を示したものです。ただし、今リリースまでの変更点や新機能の一部は表に記載されていません。

Cisco APIC リリース 5.2（6）	機能
6.0(2)	AWS を使用した仮想 APIC の展開のサポート。

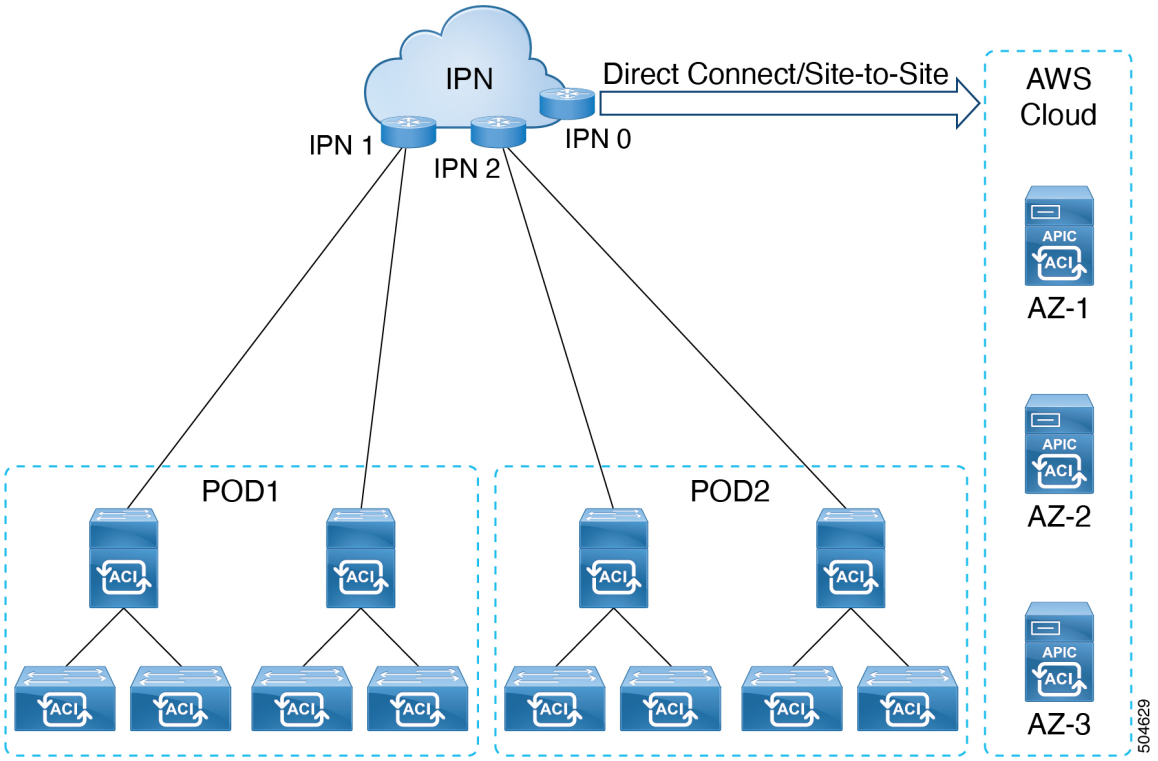
概要

Cisco APIC リリース 6.0（2）以降では、クラスタ内のすべての APIC が仮想 APIC であるクラスタを展開できます。VMware vCenter を使用して ESXi に仮想 APIC を展開するか、パブリック クラウドに仮想 APIC を展開できます。このリリースでサポートされているパブリック クラウドは、Amazon Web Services（AWS）です。

このドキュメントでは、AWS を使用した仮想 APIC の展開について詳しく説明します。 VMware vCenter を使用して ESXi ホストに仮想 APIC を展開する方法の詳細については、[VMware vCenter を使用した仮想 APIC の展開（Deploying Virtual APIC Using VMware vCenter）]に関するドキュメントを参照してください。

この展開では、仮想 APIC クラスタは AWS パブリック クラウドで実行され、ファブリック スイッチはお客様の構内（オンプレミス）に展開されます。APIC は、レイヤ 3 ネットワークを介してファブリックにリモート接続されます。3 ノード クラスタが推奨されます。高可用性と冗長性を確保するために、3 つの APIC は AWS の 3 つの異なる可用性ゾーン（AZ）の下に展開されます。各 AZ には、異なるサブネットが必要です。

図 *1* : *AWS* を使用した仮想 *APIC* の展開



IPN は、直接接続またはサイト間 VPN を使用して AWS クラウドに接続できます。IPN を介して接続されたオンプレミス デバイスと AWS クラウド上の APIC クラスタの間に、10 Gbps の最小帯域幅があることを確認します。

直接接続とサイト間 VPN の詳細については、関連する *[AWS* ドキュメント（*AWS documentation*）*]* を参照してください。

注意事項と制約事項

AWS を使用して仮想 APIC を展開するためのガイドラインと制限は次のとおりです。

- ファブリック スイッチは、Cisco APIC リリース 6.0（2）以降を実行している必要があります。リリース 6.0（2）より前のバージョンを実行しているファブリック スイッチは、自動ファームウェア アップデートを使用したファブリック検出中に、リリース 6.0（2）リリースに自動的にアップグレードできます。
- 混合モードはサポートされていません。つまり、クラスタのすべての APIC は同じタイプである必要があります。
  - AWS の仮想 APIC は、ESXi ホストの仮想 APIC でクラスタを形成できません。
  - AWS の仮想 APIC は、物理 APIC でクラスタを形成できません。
- ESXi を使用して展開された仮想 APIC は AWS に移行できず、その逆も同様です （AWS から ESXi に展開されます）。AWS を使用して展開された仮想 APIC から物理 APIC への移行もサポートされていません。サポートされている移行シナリオについては、物理 *APIC* から仮想 *APIC* への移行セクションを参照してください。

- AWS を使用して仮想 APIC（リリース 6.0（2））を展開した後は、Cisco APIC リリース 6.0（2）より前のリリースにダウングレードすることはできません。
- スタンバイ APIC サポートなし（冗長性なし）。

  物理 APIC クラスタの場合、スタンバイ APIC は、現用系クラスタと一緒のファームウェア バージョンだと確認するためファームウェアアップデートと一緒に自動的にアップデートされます。これにより、障害発生時の APIC の置き換えが可能になります。ただし、仮想 APIC クラスタの場合、ユーザーは必要に応じて同じバージョンの APIC のインスタンスを作成できるため、スタンバイ APIC は必要ありません。
- クラスタとファブリックのセキュリティは、自己署名証明書を使用して提供されます。
- AWS を使用して展開された仮想 APIC では、IPv6 はサポートされていません。

  IPv6 は、契約によるインバンドおよびアウトオブバンド管理ではサポートされていません。
- AWS の仮想 APIC ではアプリまたはアプリ インフラがサポートされていません。つまり、DC AppCenter から仮想 APIC（AWS を使用して展開）に外部 ACI アプリをダウンロードしてインストールすることはできません。事前にパッケージ化されたアプリのみがサポートされています。

AWS を使用して仮想化 APIC を展開します

この手順を使用して、AWS を使用して Cisco 仮想 APIC を展開します。

始める前に

前提条件：

- Amazon アカウントにログインし、AWS の管理者アクセス権があることを確認します。
- AWS アカウントがインスタンスの展開の制限を許可したことを確認します。AWS 管理コンソールのアカウント インスタンスの制限は、**[サービス（Services）]** > **EC2**> **Limits**から確認できます。

  本番環境の APIC クラスタには少なくとも 3 つの EC インスタンスが必要です。つまり、AWS アカウントは 3 つの追加の *r6i.4xlarge* EC2 インスタンスを起動できる必要があります。

  以下の表は、AWS 上の仮想 APIC でサポートされているクラウド インスタンス タイプを示しています。

AWS EC2 インスタンス	**vCPU**	**メモリ（4 GB RAM）**
r6i.4xlarge（推奨）	16	128
r6i.8xlarge	32	256

- スタックの作成に使用するネットワーク情報技術を作成します。次の情報技術を作成します。
  - VPC 識別子 - 仮想 APIC が展開される仮想プライベートクラウド（VPC）識別子。VPC 識別子 プレフィックスがオンプレミス デバイスの IP プレフィックスと競合していないことを確認します。

    VPC 識別子がすでにある場合は、それを使用できます。新しい VPC の作成は必須ではありません。
  - サブネット識別子 — VPC のサブネット範囲。

高可用性のために、各可用性に3つのサブネット識別子を作成します。1つは帯域外管理、インフラおよび帯域内管理用です。3 ノード クラスタには 3 つの可用性ゾーンがあり、9 つのサブネット ID が必要です。展開手順で使用するので、手元に置いておいてください。各可用性ゾーンのサブネット識別子を明確に示します。

AZ1	APIC 1	OOB、インフラ、インバンド管理のサブネット識別子。
AZ2	APIC 2	OOB、インフラ、インバンド管理のサブネット識別子。
AZ3	APIC 3	OOB、インフラ、インバンド管理のサブネット識別子。

（注） 3 つのサブネットはすべて、1 つの APIC の 1 つの可用性ゾーンに属している必要があります。

VPC とサブネットの作成の詳細については、関連する *[AWS* ドキュメント（*AWS documentation*）*]*を参照できます。

例：

```
Vpc : vpc-062429c055a4a7416
Subnets:
VPC: vapic-example1-vpc, AZ: AZ1
vapic-oob-az1-subnet:   10.1.0.0/28    GW: 10.1.0.1
vapic-infra-az1-subnet: 10.1.0.16/28   GW: 10.1.0.17
vapic-inb-az1-subnet:   10.1.0.128/28  GW: 10.1.0.129
```

作成の前提条件の構成を自動化するクラウド形成テンプレートの例については、その他の参考資料 （19 ページ）セクションを参照してください。

- Amazon EC2 SSH キーペアを作成します。

  1. 画面の左上のエリアにある **[サービス（Services）]** リンクをクリックし、[EC2] リンクをクリックします。

     [EC2 ダッシュボード (EC2 Dashboard)] 画面が表示されます。

  2. EC2 ダッシュボード画面で、**[キー ペア（Key Pair）]** リンクをクリックします。

  3. **[キー ペアの作成（Create Key Pair）]** 画面で、次の詳細を入力します。

     - キー ペアの一意の名前を入力します。そして、**[作成（Create）]**をクリックします。
     - AWS に保存されている公開キーを示す画面が表示されます。さらに、プライバシー強化メール(PEM)ファイルが、秘密キーとともにシステムにローカルにダウンロードされます。
     - 秘密キー PEM ファイルをシステム上の安全な場所に移動し、場所をメモします。

       すでにキー ペアを作成している場合は、それを使用できます。新しいキー ペアの作成は必須ではありません。

- AWS クラウドとファブリック（オンプレミス デバイス）の間に 10 Gbps の帯域幅が必須です。

• 遅延耐性は最大 50 ミリ秒です。

手順

ステップ 1 左上の **[サービス（Service）]** 検索ボックスで、*CloudFormation* を検索します。仮想 APIC の展開に使用されるサービスは CloudFormation と呼ばれます。

ステップ 2 表示される **AWS CloudFormation** 画面で、Create Stack ボタンをクリックして CloudFormation スタックを作成します。

ステップ 3 表示されるスタックを作成画面で、次の詳細を入力します。

- 前提条件内**[テンプレートを準備（Prepare template）]** ペインで *[*テンプレート準備完了（*Template is ready*）*]*を選択します。
- **[テンプレートの指定（Specify template ）]** ウィンドウで、[テンプレート ファイルのアップロード（Upload a template file）] オプションを選択します。ローカル マシンからテンプレート ファイルをアップロードします。

（注） AWS での Cisco 仮想 APIC の AWS マーケットプレイス への更新が進行中です。

ステップ 4 [次へ（Next）] をクリックします。

ステップ 5 **[スタックの詳細を指定（Specify stack details）]** 画面で、次の詳細を入力します。

- AWS の仮想 APIC 設定の識別名である **[スタック名（Stack Name）]** を指定します。
- ドロップダウン リストから、仮想 APIC を展開する **VPC 識別子**を選択します。VPC 識別子作成の詳細については、上記の *[*前提条件（*Prerequisites*）*]* セクションを参照してください。
- ドロップダウン リストから、仮想 APIC によって使用される必要な **OOB 管理サブネット識別子**（アウトオブバンド管理サブネット）を選択します。可用性ゾーンに対応する正しいサブネット識別子を選択してください。
- ドロップダウン リストから、必要な**インフラ サブネット識別子**を選択します。仮想 APIC のインフラ インターフェイスに接続されるインフラ サブネット。可用性ゾーンに対応する正しいサブネット識別子を選択してください。
- ドロップダウン リストから、必要な**インバンド管理サブネット識別子**を選択します。インバンド管理のために仮想 APIC によって使用されるインバンド管理サブネット。可用性ゾーンに対応する正しいサブネット識別子を選択してください。
- ドロップダウン リストから、推奨される**インスタンス タイプ**、つまり *r6l.4xlarge* を選択します。
- ドロップダウン リストから、**[キー ペア（Key Pair）]** を選択します。

（注） AWS の仮想 APIC ではパスワードベースの SSH 認証がデフォルトで無効になっているため、キー ペアは必須です。Cisco APIC GUI でパスワード認証を手動で有効にすることができます。**[コンソール アクセス（Console Access）]** タブに移動し、**[パスワード認証状態（Password Auth State）]** を *[有効（Enable）]* に設定します。パスは、**[ファブリック（Fabric）]** > **[ファブリック ポリシー（Fabric Policies）]** > **[ポリシー（Policies）]** > **[ポッド（Pod）]** > **[管理アクセス（Management Access）]** > **[デフォルト（Default）]** です。

- **[管理ユーザー パスワード（Admin User Password）]**を入力します。このパスワードは、展開後に仮想 APIC にアクセスするために使用されます。
- **[管理ユーザー パスワードを確認（Confirm Admin User Password）]** フィールド内で上記のパスワードをもう一度入力します。

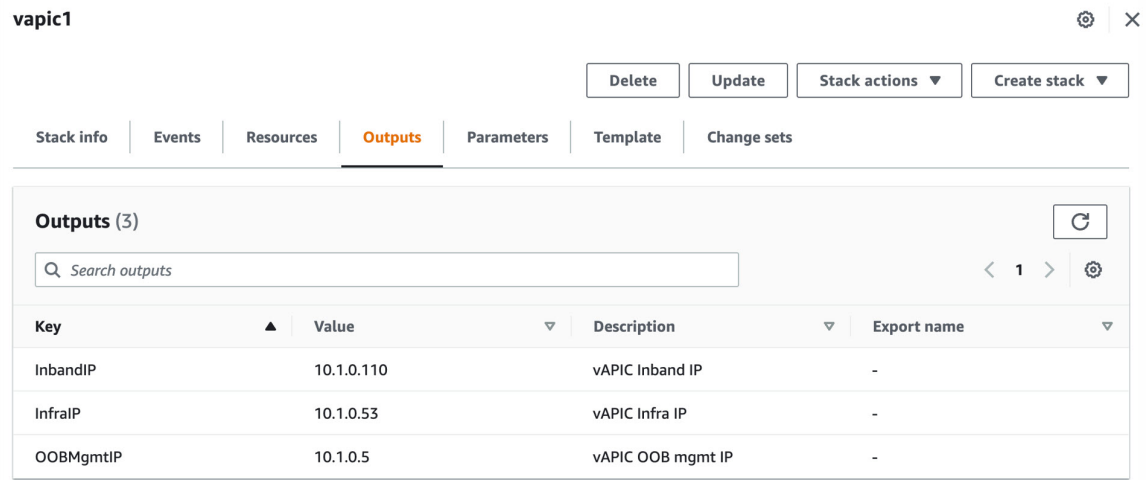
ステップ 6 [次へ（Next）] をクリックします。**[スタック オプションを構成（Configure Stack Options）]** ページが表示されます。

ステップ 7 [次へ（Next）] をクリックします。仮想 APIC の **[レビュー（Review）]** ページが表示されます。パラメータ ペインに表示される情報が正確であることを確認してください。

ステップ 8 [送信（Submit）] をクリックします。これにより、スタックの作成が開始されます。

スタック作成の進行状況は、**[イベント（Events）]** タブで確認できます。現在のステータスを表示するステータス列、CREATE_IN_PROGRESS を確認します。スタックの作成には約 8 ～ 10 分かかります。スタックが正常に作成されると、ステータス列に CREATE_COMPLETE のステータスが表示されます。

ステップ 9 **[出力（Outputs）]** タブを確認します。ここに表示されるパラメータを書き留めます。ここに表示される OOBMgmt IP アドレス（以下を参照）は、APIC クラスタのブリングアップ GUI を使用してノードを構成するときに必要です。



ステップ 10 AWS のホーム画面の **[サービス（Services）]** の横にある検索ボックスで、EC2 を検索します。

ステップ 11 **[技術情報（Resource）]** > **[インスタンス（Instances）]**.

ステップ 12 新しく作成されたスタックの **[ステータス チェック（Status Check）]** 列を確認します。合格したチェックが表示されていることを確認します。

ステップ 13 上記の手順を繰り返して、クラスタ内に各ノードを作成します。たとえば、3 ノード クラスタを構築している場合は、上記の手順を 3 回実行します。

すべての APIC ノードで管理者パスワードが同じであることを確認します。また、APIC クラスタ ブリングアップ GUI に進む前に、各ノードの **[ステータス チェック（Status Check）]** 列に、新しく作成されたすべてのインスタンス（スタック）に合格したチェックが表示されていることを確認してください。

次のタスク

最初の起動とクラスタの起動については、*Cisco APIC Getting Started Guide*のGUI を使用した Cisco APIC クラスタの起動手順を参照してください。OOB 管理 IP アドレスを使用して、クラスタ起動 GUI にアクセスします。

（注） 仮想 APIC AWS クラスタを拡張するには、最初のクラスターの起動後に、Cisco APIC GUI のノードの追加オプションを使用できます。**[システム（System）]** > **[コントローラ（コントローラ）]**に移動します。ナビゲーションウィンドウ内で、**[コントローラ（Controllers）]** > *apic_controller_name* > **[ノードから見たクラスタ（Cluster as Seen by Node）]** > **[アクション（Action）]** > **[ノードを追加（Add Node）]**を拡大します。

クラスタ サイズを増やしてノードをコミッショニングすることによるクラスタ拡張はサポートされていません。

レイヤ 3 接続された APIC クラスタと一緒に ACI ネットワークを作成

このセクションの手順では、作成された仮想APICクラスタ（クラウド上）とリモートACIファブリック間の接続を確立します。クラウド上の APIC クラスタは、DHCP リレーと、IPN によって提供される OSPF または BGP アンダーレイを使用して、ファブリック ノードを検出できます。

次のリストは、クラウド上の APIC クラスタを使用して ACI ネットワークを展開する手順の概要を示しています：

手順

ステップ 1 ファブリックに接続された IPN デバイスをプロビジョニング （10 ページ）とAPIC クラスタに接続された IPN デバイスをプロビジョニング （9 ページ） — これらの手順の説明に従って IPN を構成します。

ステップ 2 上記のように APIC クラスタを起動します。AWS を使用して仮想化 APIC を展開します （4 ページ）を参照してください。

ステップ 3 APICクラスタのレイヤー3接続を構成して、IPNを介してファブリック ポッドと通信します。ファブリック ポッドへの接続を準備 （12 ページ）を参照してください。

ステップ 4 ファブリック ポッドを持ち上げます。ファブリックは、ファブリック検出と登録の概要 （15 ページ）で説明されているように、レイヤ 3 接続を介して APIC クラスタによって検出されます。

次のタスク

同様の方法で、追加のファブリック ポッドとリモート リーフ サイトを、レイヤ 3 で接続された APIC クラスタに接続できます。

APIC クラスタ接続をレイヤ 3 ネットワークへ展開するときのガイドラインと制限。

仮想レイヤ 3 接続の APIC クラスタを展開するときは、次のガイドラインと制限に従ってください。

- すべての APIC クラスタ サイズは、レイヤ 3 で接続された APIC ポッドでサポートされます。
- レイヤ 3 で接続された APIC ポッド内の APIC は、ファブリック ポッド内の APIC でクラスタを形成できません。このトポロジでは、ファブリック ポッドに APIC がないようにする必要があります。
- レイヤ 3 で接続された APIC は、同じサブネットまたは異なるサブネットに配置できます。
- APIC 間の遅延とファブリック ポッドとの遅延が 50 ミリ秒の往復時間（RTT）を超えないことを条件として、レイヤー 3 で接続された APIC を地理的に分散させることができます。これは、およそ最大 2,500 マイルの地理的距離に相当します。
- IPN ネットワーク要件を満たすことができるデバイスはすべて IPN デバイスとして使用できますが、可能であれば、Cisco Nexus 9300 クラウド スケール ファミリのスイッチを展開することをお勧めします。これらは、実稼働環境で最も一般的に見られるデバイスであり、シスコの内部テストでより生産に検証されるデバイスでもあります。IPN デバイス要件の詳細については、[ACI マルチポッド ホワイト ペーパー（ACI Multi-Pod White Paper）]の「ポッド間接続の展開に関する考慮事項」を参照してください。
- APIC サブネットは、OSPF または BGP ルートとしてスパインにアドバタイズする必要があります。OSPF/BGP アンダーレイがサポートされています。
- APIC クラスタとファブリック ポッド間のすべてのコントロール プレーン トラフィックは IPN を通過するため、このトラフィックに QoS を構成することをお勧めします。このガイドの *[QoS の構成（Configuring QoS）]* セクションを参照してください。
- レイヤ 3 ネットワークを介したファブリックへの APIC クラスタ接続は、次をサポートしません：
  - Kubernetes の ACI CNI（Redhat Openshift、SUSE/Rancher RKE、Ubuntu 上のアップストリーム Kubernetes）
  - Openstack 用の ACI ML2（Redhat Openstack、Canonical Openstack）
- レイヤー 3 ネットワークを介したファブリックへの仮想 APIC AWS クラスタ接続は、デフォルトで許可モードをサポートします。

APIC クラスタに接続された IPN デバイスをプロビジョニング

このセクションでは、APIC クラスタ ポッドであるポッド 0 に接続された IPN デバイスの構成について説明します。この概要 （2 ページ）セクションのトポロジを参照すると、クラスタに面した IPN デバイスは IPN0 として示されています。推奨されるプラクティスとして、IPN0 は冗長性のために 2 つのデバイスで構成されます。各 APIC のファブリック インターフェイスは、2 つのデバイスにデュアルホーム接続されています。

次の構成例では、2 つの Cisco Nexus 9000 シリーズ スイッチ（IPN0a および IPN0b）が次の選択肢で構成されています：

- VLAN 1500 は、APIC のインターフェイス VLAN として使用されます。
- スイッチ インターフェイスは、レイヤ 2 トランク ポートとして構成されます。代わりに、APIC ファブリック インターフェイスが APIC セットアップ中に VLAN 0 を使用するように構成されている場合、インターフェイスはアクセス ポートである可能性があります。
- どちらのスイッチも、APIC サブネットのデフォルト ゲートウェイ アドレスとして機能する単一の IP アドレスを共有するように HSRP を使用して設定されています。
- APIC サブネットは、アンダーレイ プロトコルとして OSPF を使用してスパインにアドバタイズされます。代わりに、BGP アンダーレイを展開できます。

```
# Example configuration of IPN0a:

interface Vlan1500
  no shutdown
  vrf member IPN
  ip address 172.16.0.252/24
  ip ospf passive-interface
  ip router ospf 1 area 0.0.0.0
  hsrp version 2
  hsrp 1500
    ip 172.16.0.1

interface Ethernet1/1
  switchport mode trunk
  switchport trunk vlan 1500
  spanning-tree port type edge trunk


# Example configuration of IPN0b:

interface Vlan1500
  no shutdown
  vrf member IPN
  ip address 172.16.0.253/24
  ip ospf passive-interface
  ip router ospf 1 area 0.0.0.0
  hsrp version 2
  hsrp 1500
    ip 172.16.0.1

interface Ethernet1/1
  switchport mode trunk
  switchport trunk vlan 1500
  spanning-tree port type edge trunk
```

ファブリックに接続された IPN デバイスをプロビジョニング

このセクションでは、ファブリック ポッドに接続された IPN デバイスである MPod IPN の構成について説明します。IPN は APIC では管理されません。これは、次の情報が事前する必要があります。

- ファブリック ポッドの背表紙に接続されているインターフェイスを設定します。
- OSPF プロセスとエリア識別子を指定して、サブインターフェイスで OSPF（または BGP）を有効にします。
- スパインに接続されている IPN インターフェイスで DHCP リレーを有効にします。

- PIM をイネーブルにします。
- PIM 双方向としてブリッジ ドメイン GIPo 範囲の追加（**bidir**）の範囲をグループ化（デフォルトでは 225.0.0.0/15）。

  グループを **bidir** モードが機能の転送を共有ツリーのみ。
- PIM として 239.255.255.240/28 を追加 **bidir** 範囲をグループ化します。
- すべての背表紙に接続されたインターフェイスで PIM を有効にします。

（注） マルチキャストは、レイヤ 3 で接続された APIC クラスタを持つ単一のポッド ファブリックには必要ありませんが、マルチポッド ファブリックのポッド間で必要です。

（注） PIM **bidir** を展開する際には、どの時点であっても、特定のマルチキャスト グループ範囲に対して、1 つのアクティブな RP（ランデブー ポイント）を設定することだけが可能です。RP の冗長性が活用することで実現そのため、 **ファントム RP** 設定します。希薄モードの冗長性を提供するために使用するエニーキャストまたは MSDP メカニズムはのオプションではありませんマルチキャスト ソースの情報は、Bidir で利用可能なは不要であるため **bidir** 。

次のスイッチ構成の例は、MPod IPN として展開されたスイッチ用です。DHCP リレー設定により、APIC クラスタによるファブリックの検出が可能になります。ポッド間接続用の IPN での専用 VRF の展開はオプションですが、ベスト プラクティスとして推奨されます。代わりにグローバル ルーティング ドメインを使用することもできます。

```
Example: OSPF as the underlay protocol
feature dhcp
feature pim
service dhcp
ip dhcp relay

# Create a new VRF.
vrf context overlay-1
  ip pim rp-address 12.1.1.1 group-list 225.0.0.0/15 bidir
  ip pim rp-address 12.1.1.1 group-list 239.255.255.240/28 bidir

interface Ethernet1/54.4    #spine connected interface
  mtu 9150
  encapsulation dot1q 4
  vrf member overlay-1
  ip address 192.168.0.1/30
  ip ospf network point-to-point
  ip router ospf infra area 0.0.0.0
  ip dhcp relay address 172.16.0.2  #infra address of APIC 1
  ip dhcp relay address 172.16.0.3  #infra address of APIC 2
  ip dhcp relay address 172.16.0.4  #infra address of APIC 3
  no shutdown

interface loopback29
  vrf member overlay-1
  ip address 12.1.1.2/30

router ospf infra
  vrf overlay-1
```

```
    router-id 29.29.29.29

Example: BGP as the underlay protocol
router bgp 65010
  vrf IPN
    neighbor 192.168.0.2 remote-as 65001
      address-family ipv4 unicast
        disable-peer-as-check
```

BGP 構成では、各ポッドが同じ ASN を使用するため、マルチポッドに `disable-peer-as-check` コマンドが必要です。

ファブリック ポッドへの接続を準備

ファブリック ポッド（ポッド 1）を起動する前に、IPN を介してファブリック ポッド内のスパインに接続できるように、レイヤ 3 で接続された APIC クラスタ（ポッド 0）を事前に構成する必要があります。これは、自動ファブリック検出に必要です。

始める前に

- レイヤ 3 接続された仮想 APIC クラスタがファブリックとは別のセキュリティ ゾーンに展開されている場合は、必要なプロトコルとポートを許可するようにファイアウォールを構成します。
- ファブリック ポッド スパインに接続されているポッド間ネットワーク（IPN）デバイスを構成します。
- ファブリックの外部ルーティング プロファイルを構成します。
- アンダーレイ プロトコルとして OSPF を使用している場合は、OSPF インターフェイス ポリシーを構成します。

手順

ステップ 1 レイヤ 3 接続クラスタ内の APIC の 1 つにログインします。

ステップ 2 **[ファブリック（Fabric）] > [インベントリ（Inventory）] > [ポッド ファブリック セットアップ ポリシー（Pod Fabric Setup Policy）]**を選択します。

ステップ 3 作業ウィンドウで、+ 記号をクリックします。

[ポッド TEP プールの設定（Set Up Pod TEP Pool）] ダイアログ ボックスが開きます。

ステップ 4 **[ポッド TEP プールの作成（Set Up Pod TEP Pool）]**ダイアログボックスで、次の手順を行います。

a) **ポッド 識別子** セレクターを使用して、ポッド 1 を選択します。

b) **[TEP プール]** フィールドに、ファブリック ポッドの TEP プールを入力します。

c) [送信（Submit）] をクリックします。

ステップ 5 ナビゲーション ペインで、**[クイック スタート（Quick Start）** を展開し、**[ポッドの追加（Add Pod）]** をクリックします。

ステップ 6 作業ペインで、[Add Pod] をクリックします。

ステップ 7 [Configure Interpod Connectivity STEP 1 > Overview] パネルで、ポッド間ネットワーク（IPN）接続の設定に必要なタスクを確認し、[Get Started] をクリックします。

ステップ 8 [Configure Interpod Connectivity STEP 2 > IP Connectivity] ダイアログボックスで、次の手順を実行します。

a) [L3 Outside 設定（L3 Outside Configuration）] 領域の [名前（Name）] フィールドがある場合、[名前（Name）] ドロップダウン リストから既存のファブリック外部ルーティング プロファイルを選択します。

b) **スパイン識別子** セレクタを使用して、ポッド 0 の APIC 1 と通信するための最初のスパインとなるポッド 1 の 1 つのスパインを選択します。

c) [Interfaces] 領域の [Interface] フィールドで、IPN への接続に使用されるスパイン スイッチ インターフェイス（スロットおよびポート）を入力します。

さらにインターフェイスを追加するには [+]（プラス記号）をクリックします。

d) [IPV4 Address] フィールドに、インターフェイスの IPv4 ゲートウェイ アドレスとネットワーク マスクを入力します。

e) [MTU（bytes）] ドロップダウンリストで、外部ネットワークの最大伝送ユニットの値を選択します。

MTU は 9150（デフォルト）にする必要があります。この値は、IPN インターフェイスでも構成する必要があります。

f) [次へ（Next）] をクリックします。

ステップ 9 **[ポッド間接続構成 ステップ 3（Configure Interpod Connectivity STEP 3）] > [ルーティング プロトコル（Routing Protocols）]** ダイアログボックスの **[OSPF]** エリアで、スパインを IPN インターフェイスへ OSPF を構成するために次の手順を実行します：

a) [Use Defaults] をオンのままにするか、オフにします。

[Use Defaults] チェックボックスをオンにすると、Open Shortest Path（OSPF）を設定するための GUI のオプション フィールドが非表示になります。オフにした場合は、すべてのフィールドが表示されます。デフォルトでは、このチェックボックスはオフになっています。

b) [Area ID] フィールドに OSPF エリア ID を入力します。

c) [Area Type] 領域で、OSPF エリア タイプを選択します。

[NSSA エリア（NSSA area）] または **[通常のエリア（Regular area）]**（デフォルト）から選択できます。 **スタブ エリア**はサポートされていません。

d) （オプション）[Area Cost] セレクタで、適切な OSPF エリア コスト値を選択します。

e) [Interface Policy] ドロップダウン リストで、OSPF インターフェイス ポリシーを選択するか設定します。

既存のポリシーを選択するか、[Create OSPF Interface Policy] ダイアログボックスでポリシーを作成できます。次の表に例が表示されます：

表 *1 : OSPF* インターフェイス ポリシー例

プロパティ（Property）	**設定**
Name	ospfIfPol
Network Type	ポイント ツー ポイント
優先度（Priority）	1

プロパティ（**Property**）	設定
インターフェイスのコスト	未指定
インターフェイス コントロール（**Interface Control**）]	チェックされていません
Hello 間隔（秒）（**Hello Interval (sec)**）	10
デッド間隔（**Dead Interval**）（秒）	40
再送信間隔（**Retransmit Interval**）（秒）	5

ステップ 10 **[ポッド間接続構成 ステップ 3（Configure Interpod Connectivity STEP 3）] > [ルーティング プロトコル（Routing Protocols）]** ダイアログボックスの **[BGP]** エリアで、**[デフォルトを使用（Use Defaults）]** をチェックしたままにするか、チェックを外します。

デフォルトでは、**[デフォルトを使用（Use Defaults）]** チェックボックスはオフになっています。チェックボックスをオンにすると、Border Gateway Protocol（BGP）を構成するための GUI のフィールドが非表示になります。オフにした場合は、すべてのフィールドが表示されます。チェックボックスをオフにした場合は、次の手順を構成します。

a) [Use Defaults] をオンのままにするか、オフにします。

b) **Community** フィールドには、コミュニティ名を入力します。

デフォルトのコミュニティ名を使用することをお勧めします。別の名前を使用する場合は、デフォルトと同じ形式に従ってください。

c) [Peering Type] フィールドで、ルート ピアリング タイプとして [Full Mesh] または [Route Reflector] のいずれかを選択します。

[Peering Type] フィールドで [Route Reflector] を選択し、後でコントローラからスパイン スイッチを削除する必要がある場合は、事前に必ず [BGP Route Reflector] ページで [Route Reflector] を無効にしてください。そうしないとエラーになります。

ルート リフレクタを無効にするには、[BGP Route Reflector] ページの [Route Reflector Nodes] 領域で、該当するルート リフレクタを右クリックし、[Delete] を選択します。『*CiscoAPIC* レイヤ *3* ネットワーク コンフィギュレーション ガイド』で、「MP-BGP ルート リフレクタ」の「GUI を使用した MP-BGP ルート リフレクタの設定」の項を参照してください。

d) [Peer Password] フィールドに、BGP ピア パスワードを入力します。[Confirm Password] フィールドに、パスワードを再入力します。

e) **[ルート リフレクタ ノード（Route Reflector Nodes）]** エリアで、+（プラス記号）アイコンをクリックしてノードを追加します。

冗長性を図るため、複数のスパインがルート リフレクタ ノードとして設定されます（1 つのプライマリ リフレクタと 1 つのセカンダリ リフレクタ）。冗長性を確保するために、ポッドごとに少なくとも 1 つの外部ルート リフレクタを導入することをお勧めします。

[External Route Reflector Nodes] フィールドは、ピアリング タイプとして [Route Reflector] を選択した場合にのみ表示されます。

ステップ 11 [次へ] をクリックします。

ステップ 12 [Configure Interpod Connectivity STEP 4 > External TEP] ダイアログボックスで、次の手順を実行します。

a) [Use Defaults] をオンのままにするか、オフにします。

デフォルトでは、**[デフォルトを使用（Use Defaults）]** チェックボックスはオフになっています。チェックボックスをオンにすると、外部 TEP プールを構成するための GUI のオプション フィールドが非表示になります。オフにした場合は、すべてのフィールドが表示されます。

b) [Pod] および [Internal TEP Pool] フィールドの設定できない値に注意してください。

c) [External TEP Pool] フィールドに、物理ポッドの外部 TEP プールを入力します。

外部 TEP プールは、内部 TEP プール、または他のポッドに属する外部 TEP プールと重複しないようにする必要があります。

d) **[データ プレーン TEP IP（Data Plane TEP IP）]** フィールドでデフォルトを受け入れます。このデフォルトは **[外部 TEP プール（External TEP Pool）]** を構成したときに生成されます。別のアドレスを入力する場合は、そのアドレスが外部 TEP プールの範囲外である必要があります。

ステップ 13 [次へ（Next）] をクリックします。
[概要（Summary）] パネルが表示され、このウィザードで作成されたポリシーのリストが表示されます。ここでこれらのポリシーの名前を変更できます。

ステップ 14 [Finish] をクリックします。

次のタスク

次のセクションで要約されているように、APIC によるファブリック ノードの検出と登録をモニタします。

ファブリック検出と登録の概要

以下は、スイッチの登録および検出プロセスの概要です。

- DHCP リレー エージェントとして機能する IPN は、DHCP 要求をスパインから APIC に転送します。
- スパイン スイッチが APIC に表示されるようになり、**[ファブリック メンバーシップ（Nodes Pending Registration）]** 画面の **[ノード保留中の登録（Nodes Pending Registration）]** の下に表示されます。**[ファブリック（Fabric）]** > **[インベントリ（Invertory）]** > **[ファブリック メンバーシップ（Fabric Membership）]**に移動します。
- APIC にスパイン スイッチを登録します。
- APIC は、スパイン インターフェイスの IP アドレスを IPN に割り当て、スパインを含むファブリック ポッド用に構成された TEP プールから TEP IP を割り当てます。この時点で、スパインは ACI ファブリックに参加します。
- スパインは、IPN がこのサブネットを学習できるように、OSPF または BGP を介して IPN に TEP サブネットをアドバタイズします。
- スパインは、接続されたリーフ スイッチの DHCP リレーエージェントとして機能し、要求を APIC に転送します。

- リーフ スイッチは APIC に表示され、**[ファブリック（Fabric）] > [インベントリ（Inventory）] > [ファブリック メンバーシップ（Fabric Membership）]** の **[登録保留中のノード]** のセクションに表示されます。
- APIC でこれらのリーフ スイッチを手動で登録する必要があります。
- リーフ スイッチが登録されると、APIC は TEP IP アドレスと DHCP 設定情報をリーフに転送し、リーフは ACI ファブリックに参加します。この検出プロセスを通じて、レイヤ 3 に接続された APIC クラスタは、ファブリック ポッド内のすべてのスイッチを検出します。

パスワードベース SSH ログイン

デフォルトでは、SSH のパスワード認証は AWS の仮想 APIC で無効になっています。公開キー認証を使用してSSH経由でログインできます。

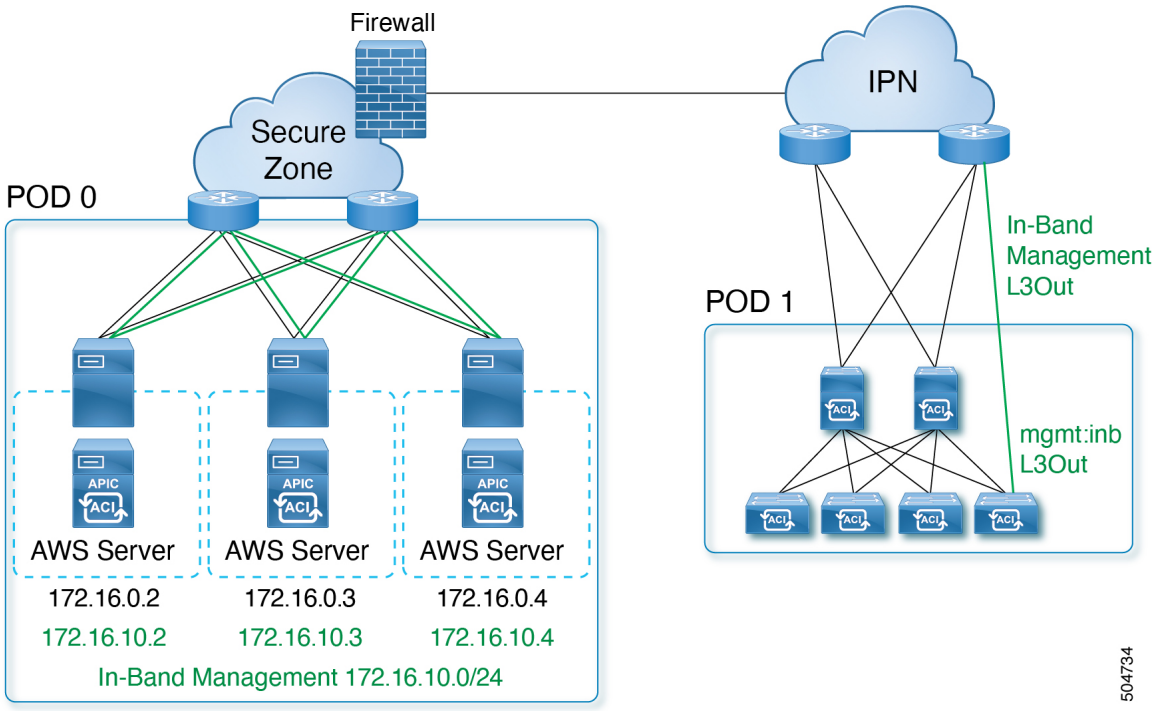
仮想 APIC クラスタが起動して完全に適合したら、標準の Cisco APIC GUI を使用してパスワード認証を有効にできます。**[ファブリック（Fabric）] > [ファブリック ポリシー（Fabric Policies）] > [ポリシー（Policies）] > [ポッド（Pod）] > [管理アクセス（Management Access）]** に移動します。管理アクセス デフォルト ページの SSH ペインの **[パスワード認証状態（Password Auth State）]** フィールドで、ドロップダウン リストから *[有効化（Enabled）]* を選択します (以前は *[無効化（Disabled）]* に設定されていました)。

CLI を使用したパスワード認証の有効化については、以下を参照してください。

```
aws-vapic# config
aws-vapic(config)# comm-policy default
aws-vapic(config-comm-policy)# ssh-service
aws-vapic(config-ssh-service)# passwd-auth-enable
aws-vapic(config-ssh-service)# exit
```

インバンド管理の構成

この手順を使用して、レイヤ 3 で接続された APIC クラスタのインバンド管理を構成します。



始める前に

レイヤ 3 接続された APIC クラスタを使用してインバンド管理を展開すると、インバンド管理 VRF（mgmt:inb）はスパインを介して IPN にルーティングされません。APIC のインバンド管理インターフェイスへの接続は、リーフ スイッチから構成された L3Out からルーティングする必要があります。この L3Out は、インバンド管理 VRF の **mgmt** テナントで構成する必要があります。

AWS での仮想 APIC のインバンド管理に関するガイドラインと制限事項

- 展開中に、各 APIC は、AWS によって割り当てられたインバンド管理インターフェイスと IP アドレスを使用します。インバンド管理インターフェイスは、無効化状態です。インバンド ポリシーが使用可能になると、インバンド管理インターフェイスが有効になります。
- インバンド IP アドレスの構成は、ユーザーが変更することはできません。

手順

ステップ 1 APIC 上流に位置するスイッチを構成します。

例：

```
interface Vlan100
  no shutdown
  vrf member IPN
  ip address 172.16.0.252/24
  ip ospf passive-interface
```

```
  ip router ospf 1 area 0.0.0.0
  hsrp version 2
  hsrp 100
    ip 172.16.0.1

interface Vlan101
  no shutdown
  vrf member IPN
  ip address 172.16.10.252/24
  ip ospf passive-interface
  ip router ospf 1 area 0.0.0.0
  hsrp version 2
  hsrp 101
    ip 172.16.10.1
```

ステップ 2 通常のインバンド管理構成手順を使用して、インバンド管理用の APIC を構成します。

a) APIC ごとに静的ノード管理アドレスを構成します。

b) インバンド EPG を構成します。カプセル化 VLAN には、最初のクラスタ起動時に指定されたインフラストラクチャ VLAN 識別子を除く任意の VLAN を使用できます。

インバンド管理の設定の詳細については、[Cisco APIC と静的管理アクセス（Cisco APIC and Static Management Access ）] テクニカル ノートの「Static In-band Management」の章を参照してください。

ステップ 3 上流に位置するスイッチ内で APIC インターフェイスを構成する。

APIC 接続インターフェイスはトランク インターフェイスである必要があります。APIC が 0 以外のインフラ VLAN で初期化された場合、次の例のようにインターフェイスを設定できます。

```
interface Ethernet1/1
  switchport mode trunk
  switchport trunk allowed vlan 100-101
```

（注） クラスタの形成後に APIC インフラ VLAN 構成を変更することはできません。

物理 APIC クラスタから AWS上 の仮想化 APIC クラスタへの移行

この手順を使用して、物理 APIC クラスタを AWS の仮想 APIC クラスタに移行します。移行は、両方の APIC（物理および仮想）がレイヤ 3 ネットワークを介して ACI ファブリックにリモート接続されている場合にのみサポートされます。

手順

ステップ 1 物理レイヤ 3 クラスタから構成をエクスポートします。

Cisco APIC GUI 上で **[管理（Admin）]** > **[インポート/ エクスポート（Import/ Export）]**に移動します。詳細な手順については、GUI を使用したエクスポート ポリシーの構成を参照してください。

ステップ 2 前に説明した展開手順 (AWS を使用して仮想化 APIC を展開します （4 ページ）) を使用し、AWS を使用して仮想 APIC クラスタを作成します。

物理 APIC と仮想 APIC で使用されるインフラ VLAN が同じであることを確認します。

ステップ 3 物理クラスタが正常かどうかを確認します。

ステップ 4 標準の Cisco APIC GUI 手順を使用して、構成を（物理 APIC から仮想 APIC に）インポートします。[GUI を使用してインポート ポリシーを構成（Configuring an Import Policy Using the GUI）]を参照します。

a) インポートが成功したら、以前に構成されたノードの既存のインバンド構成 (mgmtRsInBStNode Mo) を削除し、手動で再作成します。

b) **[テナント（Tenants）]** > **[管理（mgmt）]** > **[ノード管理アドレス（Node Management Addresses）]** > **[静的ノード管理アドレス（Static Node Management Addresses）]** に移動します。**[静的ノード管理アドレスの作成（Create Static Node Management Addresses）]** 画面で、構成モードを **[自動（Auto）]** に設定します。

ステップ 5 AWS の新しいレイヤ 3 仮想 APIC クラスタに参加するには、すべてのファブリック ノードをクリーン リロードする必要があります。

ステップ 6 ACI ファブリック ノードから仮想クラスタに到達できるように、IPN は仮想 APIC のインフラ IP アドレスで手動で更新する必要があります。

その他の参考資料

このクラウド形成テンプレートは、Cisco 仮想 APIC（vAPIC）を起動するために必要な前提条件の構成の作成を自動化します。

```
Metadata:
  "AWS::CloudFormation::Interface":
    ParameterGroups:
      - Label:
          default: VAPIC Network configuration
        Parameters:
          - VPCCidrBlock
          - AvailabilityZones
          - NumberOfAZs

    ParameterLabels:
      VPCCidrBlock:
        default: VPC CIDR
      AvailabilityZones:
        default: Availability Zones
      NumberOfAZs:
        default: Number of Availability Zones

Parameters:
  VPCCidrBlock:
    Description: VPC Cidr block used to launch VAPIC cluster
    Type: String
    AllowedPattern: "^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.){3}([0])(\\/(24))$"
    Default: 10.1.0.0/24
    ConstraintDescription: "must be a valid IP unused VPC CIDR - x.x.x.x/24"
```

```
  AvailabilityZones:
    Description: >-
      List of Availability Zones used to launch vAPIC nodes. Choose 3 AZs for high
      availability. For regions that only supports 2 AZs, choose 2 AZs (2nd &
      3rd vAPIC will be launched in the second AZ). Make sure that the value of the
      NumberOfAZs parameter matches the number of selections
    Type: "List<AWS::EC2::AvailabilityZone::Name>"
  NumberOfAZs:
    AllowedValues:
      - "2"
      - "3"
    Default: "3"
    Description: >-
      Number of Availability Zones used to launch vAPIC cluster. This count must
      match the number of AZ selections you make from the AvailabilityZones
      parameter; otherwise, deployment will fail.
    Type: String

Conditions:
  IsAZ3Available: !Equals
    - !Ref NumberOfAZs
    - "3"
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VPCCidrBlock
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-vapic-vpc

  InternetGateway:
    Type: AWS::EC2::InternetGateway
    DependsOn: VPC
  AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId: !Ref VPC
      InternetGatewayId: !Ref InternetGateway

  #  subnets
  MgmtSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select
        - "0"
        - !Ref AvailabilityZones
      CidrBlock: !Select [0, !Cidr [!Ref VPCCidrBlock, 9, 4]]
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-oob-mgmt-subnet-1

  InfraSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select
        - "0"
        - !Ref AvailabilityZones
      CidrBlock: !Select [3, !Cidr [!Ref VPCCidrBlock, 9, 4]]
      Tags:
```

```
      - Key: Name
        Value: !Sub ${AWS::StackName}-infra-subnet-1

InbandSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select
      - "0"
      - !Ref AvailabilityZones
    CidrBlock: !Select [6, !Cidr [!Ref VPCCidrBlock, 9, 4]]
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-inband-subnet-1

MgmtSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select
      - "1"
      - !Ref AvailabilityZones
    CidrBlock: !Select [1, !Cidr [!Ref VPCCidrBlock, 9, 4]]
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-oob-mgmt-subnet-2

InfraSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select
      - "1"
      - !Ref AvailabilityZones
    CidrBlock: !Select [4, !Cidr [!Ref VPCCidrBlock, 9, 4]]
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-infra-subnet-2

InbandSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select
      - "1"
      - !Ref AvailabilityZones
    CidrBlock: !Select [7, !Cidr [!Ref VPCCidrBlock, 9, 4]]
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-inband-subnet-2

MgmtSubnet3:
  Condition: IsAZ3Available
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select
      - "2"
      - !Ref AvailabilityZones
    CidrBlock: !Select [2, !Cidr [!Ref VPCCidrBlock, 9, 4]]
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-oob-mgmt-subnet-3
```

```
  InfraSubnet3:
    Condition: IsAZ3Available
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select
        - "2"
        - !Ref AvailabilityZones
      CidrBlock: !Select [5, !Cidr [!Ref VPCCidrBlock, 9, 4]]
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-infra-subnet-3

  InbandSubnet3:
    Condition: IsAZ3Available
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select
        - "2"
        - !Ref AvailabilityZones
      CidrBlock: !Select [8, !Cidr [!Ref VPCCidrBlock, 9, 4]]
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-inband-subnet-3

Outputs:
  VAPICVPC:
    Description: VPC ID.
    Value: !Ref VPC
```

【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。

この資料の記載内容は2008年10月現在のものです。
この資料に記載された仕様は予告なく変更する場合があります。

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/

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。