



Cisco APIC NX-OS スタイル コマンドライン インターフェイス コンフィギュレーション ガイド

初版：2015 年 12 月 8 日

最終更新：2018 年 5 月 22 日

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>

【注意】シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

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.

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/go/trademarks>. 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)

© 2015–2018 Cisco Systems, Inc. All rights reserved.

【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。

FSF および GNU に関する一般的な質問は、gnu@gnu.org までお送りください。[別の方法](#)でFSFに連絡することもできます。リンク切れやその他の修正、または提案については、bug-bash@gnu.org までご連絡ください。本記事の翻訳の調整および提出については、[翻訳 README](#) をご覧ください。

Copyright © 2007, 2009, 2011 Free Software Foundation, Inc. 本許可告知および著作権表示を残す限り、媒体を問わず、使用料を支払うことなく、世界中で本記事全体を複製および配布することを許可する。最終更新日：2011/06/28 02:44:32

© 2015–2018 Cisco Systems, Inc. All rights reserved.



目次

Full Cisco Trademarks with Software License ?

はじめに :

はじめに **xxi**

対象読者 **xxi**

新機能および変更された機能に関する情報 **xxi**

表記法 **xxxvii**

関連資料 **xxxviii**

マニュアルに関するフィードバック **xxxix**

第 1 章

APIC CLI の使用 1

NX-OS スタイル CLI へのアクセス 1

APIC の NX-OS スタイル CLI の使用方法 2

NX-OS の使用方法の違い 5

NX-OS Style CLI および APIC GUI の混合 6

レイヤ 3 外部接続の設定のモードについて 6

第 2 章

ファブリックとインターフェイスの設定 9

ファブリックおよびインターフェイス設定 9

グレースフル挿抜 (GIR) モード 10

CLI を使用してメンテナンス モードにスイッチを移行する 11

CLI を使用して操作モードにスイッチを挿入する 12

NX-OS CLI を使用したリーフ ノードおよび FEX デバイス上の物理ポートの設定 12

NX-OS CLI を使用したリーフ ノードおよび FEX デバイスのポートチャネルの設定 15

NX-OS CLI を使用したリーフ ノードおよび FEX デバイスの仮想ポート チャネルの設定 22

NX-OS スタイル CLI とプロファイルを使用して FEX 接続の設定 28

リフレクティブ リレー (802.1Qbg) 29

NX-OS は、CLI を使用してリフレクティブ リレーの有効化 30

インターフェイスのポリシー グループの設定 31

インターフェイスのオーバーライドの設定 34

エラー訂正の転送について 36

NX-OS スタイル CLI を使用した FEC の設定 37

第 3 章

Cisco ACI スマート ライセンス 39

スマート ライセンスについて 39

スマート ライセンスの使用法のガイドラインと制限事項 40

事前登録の検証 40

CSSM 設定の検証チェックリスト 40

スマート ライセンスおよび APIC 設定の検証チェックリスト 40

CLI を使用して Smart Licensing に対して登録 41

第 4 章

APIC 高可用性の設定 45

に関する Cold Standby APIC クラスターの 45

CLI を使用してスタンバイ apic 内でアクティブな APIC 経由でスイッチング 46

第 5 章

テナントの設定 49

テナント、VRF、およびブリッジ ドメインの作成 49

ブリッジ ドメイン設定の追加 52

適用されるブリッジ ドメインの設定 54

適用されるブリッジ ドメインの設定 55

NX-OS スタイル CLI を使用した適用されるブリッジ ドメインの設定 55

アプリケーション エンドポイント グループの作成 56

ブリッジ ドメインのレガシー転送モードの設定 59

コントラクトの設定 61

コントラクトの継承 64

コントラクト継承について 64

NX-OS スタイルの CLI を使用したアプリケーションまたは uSeg EPG のコントラクト継承の設定	66
NX-OS スタイルの CLI を使用した L2Out EPG のコントラクト継承の設定	70
NX-OS スタイルの CLI を使用した外部 L3Out EPG のコントラクト継承の設定	72
コントラクト優先グループの設定	74
契約優先グループについて	74
NX-OS スタイル CLI を使用したコントラクト優先グループの設定	76
他のテナントへのコントラクトのエクスポート	77
コントラクトまたは件名除外の設定	79
コントラクトまたはコントラクトの件名の除外の設定	79
NX-OS スタイルの CLI を使用したコントラクトまたはコントラクトの件名除外の設定	81
クォータ管理の作成	82
APIC クォータ管理の設定について	82
NX-OS スタイル CLI を使用したクォータ設定の作成	82

第 6 章

レイヤ 2 外部接続の設定	85
レイヤ 2 外部接続の設定	85
VLAN ドメインの設定	89
VLAN ドメインについて	89
基本的な VLAN ドメイン設定	90
高度な VLAN ドメイン設定	91
ポートへの VLAN ドメインの関連付け	93
ポート チャネルへの VLAN ドメインの関連付け	95
テンプレート ポリシー グループへの VLAN ドメインの関連付け	95
テンプレート ポート チャネルへの VLAN ドメインの関連付け	96
バーチャル ポート チャネルへの VLAN ドメインの関連付け	97
EPG の Q-in-Q カプセル化マッピングの設定	97
EPG の Q-in-Q カプセル化マッピング	97
NX-OS スタイル CLI を使用した Q-in-Q カプセル化リーフ インターフェイスへの EPG のマッピング	98
ACI ファブリックでイーサネット トラフィックを介したファイバ チャネルのサポート	100

ACI ファブリック上のイーサネット トラフィックによるファイバチャネルをサポートする	100
FCoE NX-OS スタイル CLI 設定	103
NX-OS スタイル CLI を使用したポリシーまたはプロファイルのない FCoE 接続の設定	103
NX-OS スタイル CLI を使用したポリシーまたはプロファイルがある FCoE 接続の設定	107
NX-OS スタイル CLI を使用して FCoE オーバー FEX の設定	111
NX-OS スタイルの CLI を使用した FCoE 設定の検証	112
NX-OS スタイル CLI を使用した FCoE 要素の展開解除	113
ファイバチャネル NPV	115
ファイバチャネル接続の概要	115
FC NPV の注意事項と制約事項	117
NX-OS CLI を使用したポリシーまたはプロファイルのない FC 接続の設定	117
ポリシーまたは NX-OS は、CLI を使用したプロファイルと FC 接続の設定	119
802.1Q トンネルの設定	121
ACI 802.1 q トンネルについて	121
NX-OS スタイル CLI を使用した802.1Q トンネルの設定	123
例 : NX-OS スタイル CLI でポートを使用する 802.1Q トンネルを設定する	125
例 : NX-OS スタイル CLI でポート チャネルを使用する 802.1Q トンネルを設定する	126
例 : NX-OS スタイル CLI で仮想ポートチャネルを使用する 802.1Q トンネルを設定する	126
ダイナミック ブレークアウト ポートの設定	127
ダイナミック ブレークアウト ポートの設定	127
NX-OS スタイルの CLI を使用したダイナミック ブレイクアウト ポートの設定	128
ポート プロファイルの設定	132
ポート プロファイルの設定	132
ポート プロファイルの設定のまとめ	136
NX-OS スタイル CLI を使用したポート プロファイルの設定	138
NX-OS スタイル CLI を使用したポート プロファイルの設定と変換の確認	139
仮想スイッチ上のマイクロセグメンテーション	140
仮想スイッチでのマイクロセグメンテーションの設定	140

NX-OS スタイル CLI を使用した Cisco ACI でのマイクロセグメンテーションの設定	141
ベア メタル上のマイクロセグメンテーションの設定	143
ベア メタルでのネットワークベースの属性によるマイクロセグメンテーションの使用	143
NX-OS スタイルの CLI を使用したベアメタル環境でのネットワークベースのマイクロセグメント EPG の設定	143
レイヤ 2 IGMP スヌープ マルチキャストの設定	145
Cisco APIC と IGMP スヌーピングについて	145
静的ポート グループの IGMP スヌーピングを有効にする	146
NX-OS スタイル CLI を使用した IGMP スヌーピング ポリシーの設定とブリッジドメインへの割り当て	147
NX-OS スタイル CLI によりスタティック ポートで IGMP スヌーピングおよびマルチキャストの有効化	148
IGMP スヌープ アクセス グループの有効化	150
NX-OS スタイル CLI を使用した IGMP スヌーピングおよびマルチキャストグループへのアクセスの有効化	150
NX-OS スタイルの CLI を使用した APIC の特定のポートへの EPG の導入	152
ポートセキュリティの設定	154
ポートセキュリティと ACI について	154
ポートセキュリティに関するガイドラインと制約事項	154
ポート レベルでのポート セキュリティ	154
ポートセキュリティ ポリシー グループ テンプレートの設定	155
テンプレートを使用したインターフェイスでのポートセキュリティの設定	156
オーバーライドを使用したインターフェイスでのポート セキュリティの設定	157
802.1 x ポートおよびノードの認証	158
802.1 x ポートおよびノードの認証	158
ポート認証 ポリシーの設定	159
ノード認証ポリシーの設定	160
プロキシ ARP の設定	161
プロキシ ARP について	161
注意事項と制約事項	168

プロキシ ARP は、Cisco NX-OS スタイル CLI を使用しての設定	169
カプセル化のフラッドイングの設定	170
MACsec の設定	171
MACsec について	171
MACSec の注意事項と制約事項	172
NX-OS スタイルの CLI を使用したMACsecの設定	174
第 7 章	レイヤ 3 外部接続の設定 179
レイヤ 3 外部接続の設定のモードについて	179
レイヤ 3 外部接続の設定	181
外部ネットワークへのルーテッド接続	182
外部ネットワークヘルトされた接続について	182
外部ネットワークへのルーテッド接続のためのレイヤ 3 Out	182
外部ネットワークへの接続をルーティングするための注意事項	184
外部レイヤ 3 Outside 接続タイプ	186
NX-OS スタイルの CLI を使用したテナント ネットワークのレイヤ 3 Outside の設定	188
NX-OS スタイル CLI の例: L3Out の前提条件	192
NX-OS スタイル CLI の例 : L3Out	192
レイヤ 3 ルーティングとサブインターフェイス ポート チャネル	194
レイヤ 3 ポート チャネルについて	194
ポート チャネルの NX-OS は、CLI を使用してをルーテッドレイヤ 3 の設定	194
NX-OS CLI を使用したレイヤ 3 サブインターフェイス ポート チャネルの設定	197
NX-OS CLI を使用したレイヤ 3 ポート チャネルにポートを追加する	200
レイヤ 3 アウトからレイヤ 3 アウト内部 VRF への漏洩	201
NX-OS スタイル CLI を使用して共有 レイヤ 3 VRF 内リークを設定する - 名前が付けられた例	202
NX-OS Style CLI を使用した共有レイヤ 3 VRF 間リークの設定 : 名前を付けた例	204
SVI 外部カプセル化の範囲について	206
カプセル化スコープ構文	208
NX-OS スタイル CLI を使用して、SVI インターフェイスのカプセル化スコープの設定	208

SVI 自動状態について	209
SVI 自動状態の動作のガイドラインと制限事項	210
NX-OS スタイル CLI を使用した SVI 自動状態の設定	210
インターフェイスとスタティック ルーティングの設定	211
OSPF の設定	214
『Configuring OSPF』	214
OSPF VRF とインターフェイス テンプレートの作成	218
BGP の設定	222
『Configuring BGP』	222
BGP アドレス ファミリとタイマー テンプレートの作成	223
BGP アドレス ファミリとタイマーの設定	224
BGP ネイバーの設定	226
NX-OS スタイル CLI を使用してノード BGP タイマー ポリシーあたりの VRF あたりを設定する	231
BGP 最長パスの設定	232
AS パス プリペンドの設定	233
NX-OS スタイルの CLI を使用した AS パスのプリペンド	234
BGP にディストリビューションをルーティング	235
テナントの範囲によるルート プロファイルの設定	235
再配布ルート プロファイルの設定	236
BGP ルート ダンプニングの設定	237
EIGRP の設定	240
EIGRP VRF とインターフェイス テンプレートの作成	240
EIGRP アドレス ファミリとカウンタの設定	243
EIGRP インターフェイスの設定	245
ルート マップの設定	248
テンプレートの設定	248
ルート プロファイルについて	248
テナント スコープ ルート プロファイルの設定	248
VRF スコープ ルート プロファイルの設定	250
ルート マップの作成	252

ルーティング プロトコルのルートマップの設定	257
エクスポート マップ（インター VRF ルート リーク）の設定	258
双方向ルート転送（BFD）の設定	259
BFD について	259
BFD グローバルの設定	260
NX-OS スタイル CLI を使用したリーフ スイッチでの BFD のグローバルな設定	263
NX-OS スタイル CLI を使用したスパイン スイッチ上の BFD のグローバル設定	264
グローバル BFD 設定のオーバーライド	265
BFD インターフェイス オーバーライド ポリシーの設定	265
インターフェイスへの BFD インターフェイス オーバーライド ポリシーの適用	268
コンシューマ プロトコル上の BFD の有効化	270
BGP コンシューマ プロトコルの BFD の有効化	270
EIGRP コンシューマ プロトコルの BFD の有効化	272
OSPF コンシューマ プロトコルの BFD の有効化	273
静的ルート コンシューマ プロトコルの BFD の有効化	273
NX-OS スタイルの CLI を使用した BFD コンシューマ プロトコルの設定	274
レイヤ 3 マルチキャストの設定	275
レイヤ 3 マルチキャスト	275
レイヤ 3 マルチキャストの設定に関するガイドライン	276
レイヤ 3 マルチキャストの設定手順	278
レイヤ 3 マルチキャストの PIM オプションの設定	278
レイヤ 3 マルチキャストの VRF での IGMP オプションの設定	281
レイヤ 3 マルチキャストの L3 Out の設定	287
例：レイヤ 3 マルチキャストの設定	291
外部 L3 EPG の設定	291
名前付きモードを使用したレイヤ 3 外部接続の設定	294
名前付き L3Out の作成	294
名前付き L3Out のレイヤ 3 インターフェイスの設定	296
名前付き L3Out のルート マップの設定	298
名前付き L3Out のルーティング プロトコルの設定	301
名前付き L3Out の BGP の設定	301

名前付き L3Out の OSPF の設定	303
名前付き L3Out の EIGRP の設定	305
名前付き L3Out の外部 L3 EPG の設定	308
IPv6 ネイバー探索	309
ネイバー探索	309
NX-OS スタイル CLI を使用したブリッジ ドメイン上の IPv6 ネイバー検索によるテナント、VRF、ブリッジ ドメインの設定	311
注意事項と制約事項	311
NX-OS スタイル CLI を使用したレイヤ 3 インターフェイス上の RA による IPv6 ネイバー探索インターフェイス ポリシーの設定	312
『Configuring HSRP』	314
NX-OS スタイル CLI での Cisco APIC を使用してインライン パラメータで HSRP の設定	314
NX-OS スタイル CLI のテンプレートとポリシーを使用した Cisco APIC の HSRP の設定	316
Cisco ACI GOLF	317
Cisco ACI GOLF	317
NX-OS スタイル CLI を使用して Cisco ACI GOLF サービスを設定する設定タスク	320
スパインを設定し、NX-OS スタイル CLI を使用して、BGP EVPN のテナント インフラ	321
複数のサイトで共有 APIC ゴルフ接続	324
NX-OS スタイル CLI を使用した推奨される共有 GOLF 設定	325
NX-OS スタイル CLI を使用して、スパインで BGP EVPN をサポートする BGP の設定	326
NX-OS スタイル CLI を使用した BG EVPN のテナントの設定	327
ルート マップの設定	329
NX-OS スタイル CLI を使用して DCIG への配布の BGP EVPN タイプ 2 のホスト ルートの有効化	332
NX-OS スタイル CLI を使用した Cisco ACI GOLF 設定の例:	332
DCIG への EVPN タイプ 2 ルート配信のトラブルシューティング	334
Multipod_Fabric	337
マルチポッド ファブリックについて	337
マルチポッド ファブリックでのスイッチの割り当て	337

マルチポッドファブリックのファブリックの外部接続の設定	338
マルチポッドファブリックのスパインインターフェイスと OSPF の設定	341
リモートリーフスイッチ	344
ACI ファブリックのリモートリーフスイッチについて	344
リモートのリーフハードウェアの要件	346
制約事項と制限	346
WAN ルータとリモートリーフ設定の注意事項	348
NX-OS スタイル CLI を使用したリモートリーフの設定	349
トランジットルーティング	352
中継 ACI ファブリックのルーティング	352
トピックに関連するトランジットルーティング	353
トランジットルーティングの概要	353
中継ルーティングのガイドライン	355
NX-OS スタイル CLI を使用したトランジットルーティングの設定	361
例：中継ルーティング	365

第 8 章

Cisco ACI QoS の設定 369

L3Outs の QoS 369

NX-OS スタイルの CLI を使用した L3Outs の QoS の設定 369

CoS の保持 370

802.1 P クラスのサービスの設定の保持 370

NX-OS スタイル CLI を使用した QoS CoS 設定の保持 371

Multipod QoS 372

NX-OS スタイル CLI を使用した DSCP ポリシーによるマルチポッド QoS の有効化 372

マルチポッドファブリックでの QoS 優先度の設定項目の保持 373

出力マーキングを QoS 入力マーキングの変換 374

QoS 入力マーキングを出力マーキングに変換する 374

NX-OS は、CLI を使用した QoS CoS 設定の変換 375

第 9 章

管理インターフェイスの設定 377

アウトオブバンド管理アクセスの設定 377

インバンド管理アクセスの設定	380
外部ネットワークからスイッチへのインバンド管理アクセスの設定	380
外部ネットワークからコントローラへのインバンド管理アクセスの設定	381
管理ステーションへのインバンド管理接続の設定	384
HTTPS/SSH ポートを開くインバンド管理コントラクトの設定	386

第 10 章

セキュリティの設定 387

セキュリティの設定について	387
AAA の設定	388
セキュリティ サーバの設定	392
RADIUS サーバの設定	392
TACACS+ サーバの設定	394
LDAP サーバの設定	396
パスワード ポリシーの設定	399
ユーザの設定	402
ローカルで認証されたユーザの設定	402
ローカル ユーザの証明書および SSH キーの設定	405
公開キー インフラストラクチャの設定	406
認証局との信頼チェーンの設定	406
キーとキーリングの設定	407
証明書署名要求の生成	408
ウェブトークンの設定	411
通信ポリシーの設定	412
HTTP ポリシーの設定	412
HTTPS ポリシーの設定	413
SSH ポリシーの設定	415
Telnet ポリシーの設定	416
AES 暗号化の設定	417
ファブリック セキュア モードの設定	418
COOP 認証の設定	419
COOP 認証について	419

COOP 認証の設定	420
FIPS の設定	421
連邦情報処理標準 (FIPS) について	421
注意事項と制約事項	421
NX-OS スタイル CLI を使用した Cisco APIC の FIPS 設定	422
コントロールプレーン ポリシングの設定	422
CoPP の概要	422
CoPP の注意事項と制約事項	425
Cisco NX-OS CLI を使用した CoPP の設定	426
NX-OS スタイル CLI を使用するプロトコル CoPP ポリシーごとのインターフェイスごとの設定	426
ファースト ホップ セキュリティの設定	427
ファースト ホップ セキュリティについて	427
ACI FHS の導入	428
注意事項と制約事項	428
NX-OS CLI を使用した FHS の設定	428
802.1X の設定	435
802.1X の概要	435
ホスト サポート	435
認証モード	436
注意事項と制約事項	436
コンフィギュレーションの概要	437
NX-OS スタイル CLI を使用した 802.1x ノード認証の設定	437
NX-OS スタイル CLI を使用した 802.1x ポート認証の設定	438

第 11 章

エニーキャスト サービスの設定	441
エニーキャスト サービスについて	441
NX-OS スタイル CLI を使用したエニーキャスト サービスの設定	443

第 12 章

VMM の設定	447

第 13 章

Layer 4 - Layer 7 サービスの設定 449

第 14 章

グローバル ポリシーの設定 451

グローバル ポリシーについて 451

アウトオブバンド管理 NTP の設定 452

システム クロックの設定 454

エラー ディセーブル回復の設定 455

リンク レベル検出プロトコルの設定 456

Miscabling プロトコルの設定 457

エンドポイントのループ防止ポリシーの設定 459

不正なエンドポイントの制御ポリシーの設定 460

不正なエンドポイントの制御ポリシーについて 460

NX-OS スタイル CLI を使用した エンドポイント の設定 461

IP エージングの設定 462

概要 462

NX-OS スタイル CLI を使用した IP エージング ポリシーの設定 462

ダイナミック ロード バランサの設定 463

スパニングツリー プロトコルの設定 464

『Configuring IS-IS』 466

BGP ルート リフレクタの設定 469

ノードのデコミッション 470

電力管理の設定 471

スケジューラの設定 472

システム MTU の設定 475

PTP について 475

注意事項と制約事項 478

NX-OS CLI を使用した PTP の設定 479

第 15 章

Cisco Tetration Analytics の設定 483

概要 483

NX-OS スタイル CLI を使用した Cisco Tetration Analytics の設定 483

第 16 章

NetFlow の設定 487

NetFlow について 487

NX OS スタイル CLI を使用した仮想マシン ネットワーキングの NetFlow エクスポート ポリシーの設定 488

NX OS スタイル CLI を使用したノード制御ポリシーを介した NetFlow および Tetration Analytics 機能の優先順位設定 488

NX-OS スタイル CLI を使用した NetFlow ノード ポリシー の設定 489

NX-OS スタイル CLI を使用した NetFlow インフラ セクタの設定 490

NX-OS スタイル CLI を使用した NetFlow オーバーライドの設定 492

NX-OS スタイルの CLI を使用した NetFlow テナント階層の設定 493

VMware VDS の NX-OS スタイル CLI を使用して VMM ドメインで NetFlow エクスポート ポリシーを利用する 496

VMware 用 NX OS スタイル CLI を使用したエンドポイント グループ上の NetFlow の有効化または無効化 497

第 17 章

ファームウェアの管理 499

ファームウェアの管理 499

リポジトリ イメージの追加または削除 499

カタログ ファームウェアの変更 500

コントローラ ファームウェアのアップグレード 501

スイッチ ファームウェアのアップグレード 503

第 18 章

スナップショットによる設定の管理 507

構成管理とスナップショットについて 507

スナップショットのエクスポート 507

スナップショットのインポート 509

スナップショットを使用したロールバック設定 510

スナップショット ファイルのリモート パスへのアップロードまたはダウンロード 512

スナップショット ファイルとジョブの管理 513

第 19 章

モニタリングの設定 515

Syslog の設定 515

ロギング サーバグループの設定 515

Syslog の設定 517

Call Home の設定 518

Call Home ポリシーの設定 518

Call Home 宛先プロファイルの設定 520

Call Home 宛先プロファイル設定コマンド 523

Call Home クエリの設定 523

クエリ サブツリー カテゴリ 525

TACACS 外部ログの設定 526

NX OS スタイル CLI を使用して TACACS 外部ロギング宛先グループの作成 526

NX OS スタイル CLI を使用して TACACS 外部ログの送信元の作成 527

NX-OS スタイルの CLI を使用したオンデマンドの techsupport ファイルの送信 528

ファイルエクスポートのリモート パスの設定 530

モニタリングの Show コマンドの使用方法 531

show コマンドの使用について 531

show faults コマンドの使用方法 531

show events コマンドの使用方法 533

show health コマンドの使用方法 534

show audits コマンドの使用方法 535

show stats コマンドの使用方法 536

show コマンドのエンティティ フィルタ 536

SNMP の設定 537

CLI を使用した SNMP ポリシーの設定 539

Smart Callhome の設定 540

スマート Callhome について 540

NX OS スタイル CLI を使用して Smart Callhome 宛先グループを作成する 541

第 20 章

SPAN の設定 545

SPAN および ERSPAN の設定	545
SPAN の注意事項と制約事項	546
アクセス モードの SPAN の設定	546
アクセス モードの ERSPAN の設定	548
ファブリック モードの ERSPAN の設定	551
テナント モードの ERSPAN の設定	554

第 21 章

出力の設定を実行している別 Cisco APIC に show の適用	557
インポートおよびエクスポートの設定について	557
インポートおよびエクスポート設定の注意事項と制限事項	557
CLI 設定のエクスポート	558
CLI 設定のインポート	559

第 22 章

スケール プロファイル ポリシーの転送を設定する	561
スケール プロファイル ポリシーの転送についての概要	561
スケール プロファイル ポリシーの転送のサポートされるプラットフォーム	564
NX OS スタイル CLI を使用したスケール プロファイル ポリシー転送の設定	564

付録 A :

CLI を使用して確認されたスケーラビリティ	569
CLI スケーラビリティの制限値	569

付録 B :

使用例 : トランジット トポロジによる Three-Tier アプリケーション	571
トランジット トポロジによる Three-Tier アプリケーションの展開について	571
3 層アプリケーションの展開	573
OSPF と BGP によるトランジットルーティング	575

付録 C :

例 : show コマンド	577
例 : show コマンド	577



はじめに

- [対象読者](#) (xxi ページ)
- [新機能および変更された機能に関する情報](#) (xxi ページ)
- [表記法](#) (xxxvii ページ)
- [関連資料](#) (xxxviii ページ)
- [マニュアルに関するフィードバック](#) (xxxix ページ)

対象読者

このガイドは、アプリケーションセントリックインフラストラクチャファブリックの設定および維持に携わるネットワーク管理者およびシステム管理者を対象としています。

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、このガイドでの主な変更点の概要を示したものです。ただし、このリリースまでのこのガイドの変更点や新機能の中には一部、この表に記載されていないものもあります。

表 1: *Cisco ACI*, リリース 3.2(1) の新機能と変更された動作

機能	説明	参照先
スマート ライセンス	スマート ライセンスには、Cisco Smart Licensing 対応製品として Cisco APIC の内線番号と Cisco ACI ファブリックでは有効です。	Cisco ACI スマートライセンス (39 ページ)
レイヤ 3 ルーティングとサブインターフェイス ポートチャネル	レイヤ 3 のポート チャネルのサポートが追加されました。	レイヤ 3 外部接続の設定 (179 ページ)

機能	説明	参照先
ファイバチャネル NPV	ファブリック over FC トラフィックをサポートします。	レイヤ 2 外部接続の設定 (85 ページ)
802.1 x 拡張機能	IP 電話のサポート	セキュリティの設定 (387 ページ)
エニーキャスト サービス	エニーキャストサービスは、Cisco ACI ファブリックでサポートされます。Typical use(典型的な事例、一般的な使用) ケースは、multipod ファブリックのポッドの ASA ファイアウォールをサポートするため、DNS サーバや印刷サービスなど、他のサービスを有効にするエニーキャストを使用することもできます。	エニーキャスト サービスの設定 (441 ページ)
不正なエンドポイントの制御	不正な EPs 検出するように、グローバルの不正エンドポイント検出のサポートが追加されます。	グローバルポリシーの設定 (451 ページ)
N9K-C93180YC-FX スイッチで拡張ポート プロファイルのサポート	サポートが追加された N9K-C93180YC-FX スイッチポートをアップリンクからダウンリンクまたはアップリンクをダウンリンクに変更するポート プロファイルの。	レイヤ 2 外部接続の設定 (85 ページ)
N9K-C93180YC-FX スイッチでプロファイリングされ QSFP ポートで拡張ブレイクアウト サポート	100 ギガビット (Gb) (4X25Gb) のサポートが追加し、(ACI モード)ではスイッチのダイナミック ブレイクアウト N9K C93180YC FX でプロファイリングされ QSFP ポートで 40 Gb (4X10Gb)。	レイヤ 2 外部接続の設定 (85 ページ)

機能	説明	参照先
契約とサブジェクトの例外	Epg の間で契約が拡張され、件名や契約書を例外が含まれます。これには、契約のフィルタリングで除外する Epg のサブセットが有効にします。たとえば、プロバイダー EPG と通信できるものを除くすべてのコンシューマ Epg 一致基準が自分の通信を制御する契約でサブジェクト例外に設定されていること。	テナントの設定 (49 ページ)
NX-OS スタイル CLI および APIC GUI の混在	ファブリックを設定する 2 つのインターフェイスを混在についての注意事項が追加されます。	APIC CLI の使用 (1 ページ)
転送スケールプロファイルポリシー	高ほか スケール オプションは、転送に追加プロファイルポリシーをスケールします。高最長プレフィックス照合(ほか) は、ほかスケール 128,000 はポリシー スケールは 8,000 こと以外に、スケーラビリティと同様に、デュアルスタック ポリシーを提供します。 スケール改善他 forwardingscale オプションでは、このリリースでも追加されます。	スケールプロファイルポリシーの転送を設定する (561 ページ)
トランジットルーティング	NX-OS スタイル CLI を使用して、中継ルーティングを設定する手順は、このガイドに追加されます。	レイヤ 3 外部接続の設定 (179 ページ)
外部ネットワークへのルーテッド接続	外部ネットワークへの接続を L3Out を設定する新しい手順は、このガイドに追加されます。	レイヤ 3 外部接続の設定 (179 ページ)

表 2: Cisco ACI リリース 3.1(2m) の新機能および変更された動作

機能	説明	参照先
増加した最大 MTU	Cisco APIC リリース 3.1 (2) ま で範囲は 576 ～ 9000 バイトで す。リリース 3.1 (2) 以降、最 大 MTU 値は 9216 です。デ フォルトでは、9000 から変更 されていません。	グローバル ポリシー
L3Out の QoS	L3Out 入力トラフィックの QoS ポリシー適用が強化されま す。L3Out で QoS ポリシーを 設定するには、ポリシー制御 を有効にした状態で（ポリ シー制御適用の基本設定 = 「適用」）、VRF を出力モー ド（ポリシー制御適用の方向 「出力」）に設定する必要が あります。レイヤ 3 外部ネッ トワークを制御するコントラ クトで、QoS クラスの優先順 位または DSCP 設定を設定す る必要があります。	Cisco ACI QoS の設定
Out レイヤ 3 でネイバー探索 ルータ アドバタイズメント	RS/RA パケットは、自動設定 は使用されは、ルーテッドイ ンターフェイス、レイヤ 3 サ ブインターフェイス、SVI な どのレイヤ 3 インターフェイ スで設定できます。	レイヤ 3 外部接続の設定

表 3: Cisco ACI リリース 3.1 の新機能および変更された動作

機能	説明	参照先
カプセル化のフラッド攻撃からの設定	Cisco ACI スイッチのアプリケーション スパイエンジン (ASE) と Cisco ACI リリース 3.1 (1) 以降では、すべてのプロトコルはカプセル化にフラッディングされます。複数の Epg が外部のスイッチでサポートされている 1 つのブリッジ ドメインします。2 個の EPG が同じ BD を共有し、カプセル化のフラッディング オプションがオンになっている場合、EPG フラッディング トラフィックは他の EPG に達しません。Virtual Connect (VC) トンネルネットワークを Cisco ACI スイッチを使用しての課題を克服します。	カプセル化のフラッディングの設定 (170 ページ)
プロトコルごとのインターフェイスごとの CoPP	CoPP の設定のサポート、インターフェイスあたりプロトコルごと。	コントロールプレーン ポリッシングの設定
リモート リーフ スイッチ	導入 ACI ファブリックと ACI サービスとローカル スパイ スイッチまたは接続された APIC を持たない Cisco ACI リーフ スイッチとリモート データ センターに APIC management を拡張することができます。	リモートのリーフスイッチでレイヤ 3 の外部接続の設定

機能	説明	参照先
Multipod しの新しいハードウェア サポート	マルチポッドおよび GOLF は、すべての Cisco Nexus 9300 プラットフォーム ACI モード スイッチと、Cisco Nexus 9500 プラットフォーム ACI モード スイッチラインカードとファブリック モジュールによりサポートされています。Cisco APIC3.1(x) のリリース、および以上の場合、これに N9K C9364C スイッチが含まれます。	Cisco ACI ゴルフ と Multipod ファブリック で 3 の外部接続をレイヤの設定
MACsec	MACsec は、暗号化キーにアウトオブバンド方式を使用して、有線ネットワーク上で MAC レイヤの暗号化を提供します。MACsec Key Agreement (MKA) プロトコルでは、必要なセッション キーを提供し、必要な暗号化キーを管理します。	MACsec の設定
使用して、複数のサイト間でゴルフ接続の共有	拡大 Vrf ゴルフ接続を共有する場合、複数のサイト トポロジでは、APIC サイトの VRF 間のトラフィックの問題を回避するガイドラインが追加されました。	レイヤ 3 外部接続の設定 の Cisco ACI GOLF
SVI 自動状態	SVI 自動状態スイッチ仮想インターフェイスの動作を有効にできます。これにより、VLAN 内のすべてのポートがダウンするとダウン状態に SVI 状態です。 この機能は、APIC リリース 2.2(3x) リリースおよび APIC リリース 3.1 (1) で使用できます。APIC リリース 3.0(x) ではサポートされていません。	レイヤ 3 外部接続の設定

機能	説明	参照先
スパイン スイッチの BFD サポート	Bidirectional Forwarding Detection (BFD) スパイン スイッチのサポートが追加されました。	双方向ルート転送 (BFD) の設定
SNMP トラップ集約	Apic のパフォーマンス、クラスタ内のいずれかに配信する SNMP トラップ集約ファブリック ノードからの SNMP トラップを有効にします。	SNMP の設定



(注) APIC リリース 2.2(3x) 機能では、この特定のリリースで使用可能なのみです。APIC リリース 3.0(x) またはリリース 3.1(x) ではサポートされません。

表 4: Cisco ACI リリース 2.3(3x) の新機能および変更された動作

機能	説明	参照先
SVI 自動状態	SVI 自動状態スイッチ仮想インターフェイスの動作を有効にできます。これにより、VLAN 内のすべてのポートがダウンするとダウン状態に SVI 状態です。	レイヤ 3 外部接続の設定

表 5: Cisco ACI リリース 3.0(1k) の新機能および変更された動作

機能	説明	参照先
スケール プロファイル ポリシーの転送	転送スケールプロファイルポリシーにより、デュアル スタック（デフォルト プロファイル）と IPv4 スケールの間で選択できるようになります。デュアルスタックポリシーに設定されている転送スケールプロファイルポリシーでは、Ipv6 設定に最大 6 K エンドポイントと Ipv4 設定に最大 12 K エンドポイントのスケラビリティを提供します。IPv4 スケール オプションでは、最大 24 K IPv4 エンドポイントにスケラビリティを増加する IPv6 設定を使用しないシステムを有効にします。	スケール プロファイル ポリシーの転送を設定する
グレースフル挿抜（GIR）モード	正常な挿入し、削除(登場させ)モードまたはメンテナンスモード最小サービスの中断でネットワークからのスイッチを特定するためにできます。	メンテナンス モードの CLI を使用してスイッチを削除
EPG の Q-in-Q カプセル化マッピング	Cisco APIC を使用して、通常のインターフェイスまたは VPC で二重タグ付き VLAN トラフィック入力を EPG にマッピングできます。この機能が有効になっている場合、二重タグ付きトラフィックが EPG のネットワークに入るとき、両方のタグがファブリックで個別に処理され、ACI スイッチを入力するとき二重タグに復元されます。単一タグおよびタグなしのトラフィックの入力はドロップします。	Epg の Q-で-Q カプセル化のマッピングの設定 で レイヤ 2 の外部接続の設定

機能	説明	参照先
802.1X ポート認証	このリリースでは、802.1 x ポートの認証ポリシーまたは 802.1 x 認証ポリシーのノードを設定できます。	802.1 x 認証ポリシーのポートの設定 および 802.1 x 認証ポリシーのノードを設定 で レイヤ 2 接続の設定
ファーストホップセキュリティ	有効には、レイヤ 2 リンク上で IPv4 と IPv6 のリンクセキュリティと管理を向上します。	セキュリティ設定のファーストホップセキュリティの設定
高精度時間プロトコル	ノードに対して IEF 1588 で定義された時間同期プロトコルは、APIC 全体で配信されました	PTP を設定 で グローバルポリシーの設定
適用されるブリッジドメイン	サブジェクトエンドポイントグループ (EPG) で、エンドポイントに ping できるの関連付けられているブリッジドメイン内のサブネット ゲートウェイのみ enforced ブリッジドメインをサポートします。 この設定を有効にすると、そのサブネット ゲートウェイに ping を実行できる IP アドレスのグローバル例外リストを作成できます。	ブリッジドメインを適用 でテナントの設定

表 6: Cisco ACI リリース 2.3(1e) の新機能および変更された動作

機能	説明	参照先
Cisco APIC クォータの管理	作成し、削除すると、特定のテナントでまたはテナント間でグローバルに追加できるような管理対象オブジェクトを制限する admin を有効にするクォータ管理設定を更新します。	クォータ管理の作成

機能	説明	参照先
コントラクトの継承	コントラクトを新しい EPG に合理的に関連付けられるように、同じテナント内で他の EPG に直接関連付けられたすべての（指定/消費）コントラクトを継承できるようになりました。コントラクトの継承は、アプリケーション EPG、マイクロセグメント EPG、L2Out EPG、および L3Out EPG に設定できます。EPG コントラクト マスターのコントラクトに対して変更を行うと、継承側の EPG が変更内容を受信します。	参照してください 契約継承で テナントの設定
802.1 q トンネル機能拡張	使用するためのコア スイッチのポートを設定するようになりました Dot1q トンネル 複数のお客様にします。アクセス VLAN を定義して、corePorts を消費する顧客を区別できます。MAC ラーニングを無効にすることもできます Dot1q トンネル 。	レイヤ 2 外部接続の設定
コントロールプレーンポリシー	コントロールプレーンを保護し、それをデータプレーンから分離することによって、ネットワークの安定性、到達可能性、およびパケット配信を保証します。	セキュリティの設定
SVI のレイヤ 3 外部ネットワーク経由でのカプセル化の範囲	このリリースでは、レイヤ 3 外部ネットワーク経由で SVI のカプセル化の範囲を設定できます。	「レイヤ 3 外部接続の設定」を参照してください
対称ハッシュ	ポート チャネルでは、対称ハッシュ化はサポートされています。	「NX-OS CLI を使用したリーフ ノードでのポートチャネルの設定」を参照してください

機能	説明	参照先
リフレクティブ リレー (802.1Qbg)	リフレクティブ リレーは、外部ネットワーク スイッチへのホストサーバから仮想マシンのスイッチングを転送します。同じ物理サーバ上の Vm と、ネットワークの残りの部分間の接続を提供します。これにより、同じサーバ上の Vm の間のトラフィックに適用する Cisco APIC で設定するポリシーができます。	「ファブリックとインターフェイスの設定」を参照してください
仮想スイッチでは"マイクロセグメンテーション"	VMware VDS、シスコの AV、および Microsoft vSwitch で microsegment Epg の設定のコンテンツを追加します。	参照してください 仮想スイッチでの"マイクロセグメンテーション"の設定

表 7: Cisco APIC 2.2(2e) リリースの新機能と変更された動作

機能または変更	説明	参照先
BGP タイマーのノードごとの VRF あたり	このリリースでは定義および BGP タイマーを関連付ける、VRF あたりノードごと。	レイヤ 3 外部接続の設定
レイヤ 3 をインター VRF 漏出 Out Out 3 をレイヤします。	このリリースでは、異なる Vrf に、共有のレイヤ 3 が記録されるコントラクトを使用して相互に通信できます。	レイヤ 3 外部接続の設定
ルートプレフィックスごとに割り当てられた複数の BGP コミュニティ	このリリースでは、複数の BGP コミュニティは、ルートプレフィックスの BGP プロトコルを使用してごここ割り当てられます。	レイヤ 3 外部接続の設定
Show config コマンドの出力を実行している別 Cisco APIC に適用します。	2 つの新しい CLI コマンド設定をエクスポートするには、設定をインポート、および別 Cisco APIC で show running-config コマンドの出力を実行中の有効化に追加されました。	<i>About Import and Export Configurations in</i> 出力の設定を実行している別 Cisco APIC に show の適用

機能または変更	説明	参照先
名称変更	「ファブリック WAN のレイヤ 3 EVPN サービス」の名称が「Cisco ACI GOLF」に変更されました	<i>Cisco ACI GOLF and Multipod in Configuring Layer 3 External Connectivity</i>

表 8: Cisco APIC 2.2(1n) リリースの新機能と変更された動作

機能	説明	参照先
802.1 q トンネリング	802.1 q を設定するトンネルがファブリック、QoS (QoS) の優先順位の設定のイーサネットフレームのトンネリングをポイントするマルチ ポイントの有効化します。	802.1 q の設定のトンネル でレイヤ 2 の外部接続の設定
APIC クラスタハイアベイラビリティ	サポートは、apic のパフォーマンスがアクティブ/スタンバイ モードでクラスタ内の動作に追加されます。Apic 内で、指定したアクティブな apic のパフォーマンス共有負荷をクラスタ化し、指定のスタンバイ apic のパフォーマンスは、apic のパフォーマンスがアクティブなクラスタ内のいずれかの交換用として動作できます。	APIC ハイアベイラビリティ
優先グループを契約します。	優先契約グループ VRF で Epg 間の通信の高い制御を有効にするには、サポートが追加されます。VRF で Epg のほとんどは、オープンな通信である必要がありますが、いくつかの他の Epg との通信が制限のみする必要がありますと、は、通信を正確に制御するフィルタと契約優先グループと契約の組み合わせを設定できます。	グループを優先契約を構成する で テナントの設定

機能	説明	参照先
ダイナミック ブレークアウト ポート	(Cisco 40-4 X 10 ギガビット ブレークアウト ケーブルをギガビット) と 4-10 ギガビット対応 (ダウンリンク) デバイスに 40 のギガビットイーサネット (GE) リーフ スイッチ ポートを接続するためのサポートが追加されます。	ダイナミック ブレークアウト ポートの設定 で レイヤ 2 の外部接続の設定
FCoE over FEX	FCoE over FEX ポートを設定できるようになりました。	ACI ファブリックでイーサネットトラフィックを介したファイバチャネルのサポート
CDP FEX デバイスへのインターフェイスでのポリシーではサポート	このリリースでは、サポートはインターフェイスの CDP の FEX のデバイスに追加されました。	ファブリックとインターフェイスの設定
HSRP	デフォルトのルータ IP アドレスで設定されたイーサネットネットワーク上の IP ホストに、ファーストホップルーティング冗長性を提供するプロトコルである HSRP にサポートが追加されます。	HSRP の設定 で レイヤ 3 の外部接続の設定
NetFlow	ネットワークトラフィックアカウンティング、従量制のネットワーク課金、ネットワークプランニング、そしてサービス拒絶に対する監視機能、ネットワーク監視、社外マーケティング、およびサービスプロバイダと企業顧客向け両方のデータマイニングなど、主要な一連のアプリケーションの計測基盤を効果的にする NetFlow テクノロジーにサポートが追加されます。	NetFlow の設定
VLAN ドメイン	レイヤ 2 外部接続の設定へ移動しました	VLAN のドメインの設定 で レイヤ 2 の外部接続の設定

表 9: Cisco APIC 2.1(1h) リリースの新機能と変更された動作

機能	説明	参照先
IP エーijing	このリリースでエーijingを追跡し、エンドポイントの未使用の ip アドレスをエーijing ポリシー IP はサポートされています。	IP エーijing の設定
新しい一致タイプを使用して、明示的なプレフィックスリストを使用してルートマップ/プロファイルを作成します。	このリリースで明示的なプレフィックス リストは一致のルートの宛先と呼ばれる新しい一致タイプでサポートされています。	ルート マップの作成
FIPS の設定	このリリースでは、FIPS のサポートします。FIPS は特定の暗号アルゴリズムがセキュアであることを条件とするほか、FIPS 準拠である場合は、どのアルゴリズムを使用すべきかも指定しています。	Cisco APIC の FIPS の設定
EVPN タイプ 2 ホスト ルートの配布	このリリースでは、EVPN トポロジの最適なトラフィック転送を実現するため、ファブリックのスパインが BGP EVPN タイプ 5 (IP プレフィックス) ルートの形式のパブリック BD サブネットとともに、EVPN タイプ 2 (MAC-IP) ルートを使用するホスト ルートを DCIG にアドバタイズするように設定できます。	ファブリック WAN 上のレイヤ 3 EVPN サービスの設定の NX OS を使用した EVPN タイプ 2 ホスト ルートの配信を有効化

機能	説明	参照先
設定 IGMP スヌープ レイヤ 2 マルチキャストのサポート	このリリースでは、IGMP トラフィックを監視し、レイヤ 2 トラフィックをフラッディングからマルチキャストをフィルタリングするネットワークスイッチをちなみに IGMP スヌープ サポートを実装します。実装される機能の間でスタティック ポート グループの設定とアクセスグループ設定です。	IGMP スヌープ スタティック ポート グループの有効化 および IGMP スヌープ アクセス グループの有効化 で レイヤ 2 IGMP スヌープ マルチキャストの設定
ベアメタル環境でのネットワークベースのマイクロセグメント EPG の設定	このリリースでは、物理エンドポイントデバイスの IP アドレス属性または MAC アドレス属性を使用してマイクロセグメント EPG を設定できます。	ベアメタルで"マイクロセグメンテーション"の設定 1
CoS の QoS 設定の変換	このリリースでは、CoS 値のみに基づいてトラフィックを分類するデバイスのトラフィックを分類して、ACI ファブリック有効にすることができます。	NX-OSは、CLIを使用してQoS CoS 設定の変換

表 10: Cisco APIC 2.0(2f) リリースの新機能と変更された動作

機能	説明	参照先
『Proxy ARP』	プロキシ ARP Cisco ACI では、ネットワークまたはサブネット、エンドポイントの実際の MAC アドレスがわからなくても他のエンドポイントとの通信を内でエンドポイントを有効にするのに追加されます。	プロキシ ARP について (161 ページ)
Tetration Analytics	Cisco Tetration 分析エージェントの設定が追加されます。	概要 (483 ページ)
Multipod QoS	Multipod トポロジの維持 CoS と DSCP 設定のサポートが追加されます。	Multipod ファブリックでの QoS 優先度の設定項目の保持

機能	説明	参照先
ファブリック WAN 経由のレイヤ 3 EVPN サービス	レイヤ 3 EVPN サービスを設定する方法の詳細が追加されました。	NX-OS スタイル CLI を使用して Cisco ACI GOLF サービスを設定する設定タスク (320 ページ)

リリース	機能	ここで、
2.0(1)	ポート セキュリティ	ポートセキュリティと ACI について (154 ページ)
2.0(1)	COOP 認証	COOP 認証について (419 ページ)
2.0(1)	レイヤ 3 マルチキャスト	レイヤ 3 マルチキャスト (275 ページ)
2.0(1)	ファブリック WAN 経由のレイヤ 3 EVPN サービス	Cisco ACI GOLF (317 ページ)
2.0(1)	マルチポッドファブリック	マルチポッドファブリックについて (337 ページ)
2.0(1)	CLI を使用して確認されたスケーラビリティ	CLI を使用して確認されたスケーラビリティ (569 ページ)
1.2(2)	BFD	BFD について (259 ページ)
	ルート集約	EIGRP インターフェイスの設定 (245 ページ) 『Configuring OSPF』 (214 ページ)
	ルート ダンプニング	レイヤ 3 外部接続の設定 (179 ページ)
	レイヤ 3 外部接続を設定するための名前付きモード	レイヤ 3 外部接続の設定 (179 ページ)
	IPv6 サポート	レイヤ 3 外部接続の設定 (179 ページ)
1.2(1)	初回リリース	--

表記法

コマンドの説明には、次のような表記法が使用されます。

表記法	説明
bold	太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。
<i>italic</i>	イタリック体の文字は、ユーザが値を入力する引数です。
[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。
[x y]	いずれか1つを選択できる省略可能なキーワードや引数は、角カッコで囲み、縦棒で区切って示しています。
{x y}	必ずいずれか1つを選択しなければならない必須キーワードや引数は、波かっこで囲み、縦棒で区切って示しています。
[x {y z}]	角かっこまたは波かっこが入れ子になっている箇所は、任意または必須の要素内の任意または必須の選択肢であることを表します。角かっこ内の波かっこと縦棒は、省略可能な要素内で選択すべき必須の要素を示しています。
variable	ユーザが値を入力する変数であることを表します。イタリック体不能使用できない場合に使用されます。
string	引用符を付けない一組の文字。string 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 string とみなされます。

例では、次の表記法を使用しています。

表記法	説明
screen フォント	スイッチが表示する端末セッションおよび情報は、スクリーンフォントで示しています。
太字の screen フォント	ユーザが入力しなければならない情報は、太字のスクリーンフォントで示しています。
イタリック体の screen フォント	ユーザが値を指定する引数は、イタリック体のスクリーンフォントで示しています。
<>	パスワードのように出力されない文字は、山カッコ (<>) で囲んで示しています。
[]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。

表記法	説明
!、#	コードの先頭に感嘆符 (!) またはポンド記号 (#) がある場合には、コメント行であることを示します。

このマニュアルでは、次の表記法を使用しています。



(注) 「注釈」です。役立つ情報やこのマニュアルに記載されていない参照資料を紹介しています。



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。



警告 安全上の重要な注意事項

「危険」の意味です。人身事故を予防するための注意事項が記述されています。機器の取り扱い作業を行うときは、電気回路の危険性に注意し、一般的な事故防止対策に留意してください。Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

これらの注意事項を保存しておいてください

関連資料

Cisco Application Centric Infrastructure (ACI) Documentation

ACI のマニュアルは、次の URL で入手できます。<http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html>。

シスコアプリケーションセントリックインフラストラクチャ (ACI) シミュレータのマニュアル

Cisco ACI Simulator のマニュアルは、次の URL から入手できます：<http://www.cisco.com/c/en/us/support/cloud-systems-management/application-centric-infrastructure-simulator/tsd-products-support-series-home.html>

Cisco Nexus 9000 シリーズ スイッチのマニュアル

Cisco Nexus 9000 シリーズ スイッチのマニュアルは、次の URL で入手できます。<http://www.cisco.com/c/en/us/support/switches/nexus-9000-series-switches/tsd-products-support-series-home.html>

Cisco Application Virtual Switch のマニュアル

Cisco Application Virtual Switch (AVS) のマニュアルは、次の URL で入手できます。

<http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/tsd-products-support-series-home.html>

シスコアプリケーションセントリックインフラストラクチャ (ACI) と OpenStack の統合に関するマニュアル

Cisco ACI と OpenStack の統合に関するマニュアルは、次の URL から入手できます。

<http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html>

マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、apic-docfeedback@cisco.com までご連絡ください。ご協力をよろしくお願いいたします。



第 1 章

APIC CLI の使用

- [NX-OS スタイル CLI へのアクセス \(1 ページ\)](#)
- [APIC の NX-OS スタイル CLI の使用方法 \(2 ページ\)](#)
- [NX-OS の使用方法の違い \(5 ページ\)](#)
- [NX-OS Style CLI および APIC GUI の混合 \(6 ページ\)](#)

NX-OS スタイル CLI へのアクセス



(注) Cisco APIC リリース 1.0 からリリース 1.2 まで、デフォルト CLI は管理対象オブジェクト (MO) および管理情報モデルのプロパティから上で直接動作するコマンドの Bash シェルでした。Cisco APIC リリース 1.2 以降のデフォルト CLI は NX-OS スタイル CLI です。The object model CLI is available by typing the **bash** command at the initial CLI prompt.

手順

ステップ 1 From a secure shell (SSH) client, open an SSH connection to APIC at *username@ip-address*.

初期設定時に設定した管理者のログイン名とアウトオブバンド管理 IP アドレスを使用します。たとえば、*admin@192.168.10.1* などがこれに該当します。

ステップ 2 プロンプトが表示されたら、管理者パスワードを入力します。

次のタスク

NX-OS スタイル CLI を入力する場合、最初のコマンドレベルは EXEC レベルになります。このレベルから、次のコンフィギュレーション モードに移行できます。

- To continue in the NX-OS style CLI, you can stay in EXEC mode or you can type **configure** to enter global configuration mode.

NX-OS スタイル CLI コマンドの詳細については、『*Cisco APIC NX-OS Style CLI Command Reference*』を参照してください。

- To reach the object model CLI, type **bash**.

オブジェクトモード CLI コマンドの詳細については、『*Cisco APIC Command-Line Interface User Guide, APIC Releases 1.0 and 1.1*』を参照してください。

APIC の NX-OS スタイル CLI の使用方法

CLI コマンド モードの使用方法

NX-OS スタイルの CLI は、ルートに EXEC モードを持つコマンドモードの階層にまとめられています。この中には、グローバルコンフィギュレーションモードで始まるコンフィギュレーションサブモードのツリーも含まれます。利用できるコマンドは、現在のモードによって異なります。任意のモードで使用可能なコマンドのリストを取得するには、システムプロンプトで疑問符 (?) を入力します。

この表では、サブモードの例 (DNS) とともに最もよく使用される 2 つのモード (EXEC およびグローバルコンフィギュレーション) を挙げて説明します。表には、モードの開始方法と終了方法、および結果のシステムプロンプトを示しています。システムプロンプトから、現在実行しているモードを識別して、そのモードで使用できるコマンドを判断できます。

モード	アクセス方法	プロンプト	終了方法
EXEC	APIC のプロンプトから、 execsh を入力します。	apic#	終了してログインプロンプトに戻るには、 exit コマンドを使用します。
グローバルコンフィギュレーション	EXEC モードから、 configure コマンドを入力します。	apic(config)#	コンフィギュレーションサブモードを終了して親モードに戻るには、 exit コマンドを使用します。
DNS の設定	グローバルコンフィギュレーションモードから dns コマンドを入力します。	apic(config-dns)#	コンフィギュレーションモードまたはサブモードを終了して EXEC モードに戻るには、 end コマンドを使用します。

CLI のコマンド階層

コンフィギュレーションモードには、同じような機能を実行するコマンドが同じレベルに集められた、いくつかのサブモードがあります。For example, all commands that display information about the system, configuration, or hardware are grouped under the **show** command, and all commands that allow you to configure the switch are grouped under the **configure** command.

EXEC モードでは使用できないコマンドを実行するには、階層の最上位となるサブモードから開始します。For example, to configure DNS settings, use the **configure** command to enter the global configuration mode, then enter the **dns** command. DNS 設定サブモードに入ると、次の例のように、使用可能なコマンドを照会できます。

```
apicl# configure
apicl(config)# dns
apicl(config-dns)# ?
address    Configure the ip address for dns servers
domain     Configure the domains for dns servers
exit       Exit from current mode
fabric     Show fabric related information
no         Negate a command or set its defaults
show       Show running system information
use-vrf    Configure the management vrf for dns servers
where      Show the current mode

apicl(config-dns)# end
apicl#
```

各サブモードを使用すると、プロンプトのより下の階層で作業できます。To view the hierarchy for the current mode, use the **configure** command, as shown in this example:

```
apicl# configure
apicl(config)# bgp-fabric
apicl(config-bgp-fabric)# where
configure t; bgp-fabric
apicl(config-bgp-fabric)#
```

To leave the current level and return to the previous level, type **exit**. To return directly to the EXEC level, type **end**.

EXEC モード コマンド

CLI セッションを開始する場合、最初は EXEC モードから始めます。この EXEC モードから、コンフィギュレーションモードを開始できます。EXEC コマンドの大半は、現在の設定状態を表示する **show** コマンドのような 1 回限りのコマンドです。

コンフィギュレーション モード コマンド

コンフィギュレーションモードでは、既存の設定を変更できます。変更した設定を保存すると、スイッチの再起動後も変更内容が保存されます。コンフィギュレーションモードを開始すると、さまざまなプロトコル固有モードに入ることができます。コンフィギュレーションモードは、すべてのコンフィギュレーション コマンドの開始点です。

コマンドおよび構文の一覧表示

すべてのコマンドモードで、疑問符 (?) を入力することにより、使用できるコマンドのリストを表示できます。

```
apicl(config-dns)# ?
address    Configure the ip address for dns servers
domain     Configure the domains for dns servers
```

```

exit      Exit from current mode
fabric    Show fabric related information
no        Negate a command or set its defaults
show      Show running system information
use-vrf   Configure the management vrf for dns servers
where     Show the current mode

```

```

apic1(config-dns)# end
apic1#

```

特定の文字シーケンスで始まるコマンドの一覧を表示するには、それらの文字を入力した後に疑問符 (?) を入力します。疑問符の前にスペースを入れないでください。

```

apic1(config)# sh ?
aaa          Show AAA information
access-list  Show Access-list Information
accounting   Show accounting information
acllog       Show acllog information
. . .

```

コマンドの入力を完了するには、Tab キーを押します。

```

apic1# qu<TAB>
apic1# quota

```

キーワードまたは引数のリストを表示するには、キーワードまたは引数の代わりに疑問符を入力します。疑問符の前にスペースを1つ入れてください。この形式のヘルプをコマンド構文ヘルプと呼びます。入力したコマンド、キーワード、および引数に基づいて、使用できるキーワードまたは引数を表示するためです。

```

apic1(config-dns)# use-vrf ?
inband-mgmt  Configure dns on inband
oob-mgmt     Configure dns on out-of-band

apic1(config-dns)#

```

略語が明確であれば、コマンドを省略できます。この例では、**configure** コマンドは省略されています。

```

apic1# conf
apic1(config)#

```

「no」プレフィックスを使用して、取り消すかデフォルト値または条件に戻る

For many configuration commands, you can precede the command with the **no** keyword to remove a setting or to restore a setting to the default value. この例では、以前に設定された DNS アドレスを設定から削除する方法を示しています。

```

apic1(config-dns)# address 192.0.20.123 preferred
apic1(config-dns)# show dns-address
Address          Preferred
-----
192.0.20.123     yes

```



```
apicl(config-dns)# no address 192.0.20.123
apicl(config-dns)# show dns-address
Address                Preferred
-----
```

NX-OS スタイル CLI から Bash コマンドを実行する

To execute a single command in the bash shell, type **bash -c 'path/command'** as shown in this example.

```
apicl# bash -c '/controller/sbin/acidiag avread'
```

NX-OS スタイル CLI のすべてのモードまたはサブモードから Bash コマンドを実行できます。

スペースや特殊文字を含むコンフィギュレーション テキストを入力する

設定フィールドがユーザ定義のテキストで構成されている場合、Bashでの誤った解釈を避けるため、「\$」などの特殊文字はエスケープ（「\」）し、単語または文字列全体は単一引用符で囲む必要があります。

NX-OS の使用方法の違い

APIC の NX-OS スタイル CLI の使用方法には、次のような従来の NX-OS CLI とは異なる点があります。

- グローバルコンフィギュレーションモードは、**configure**の**configure terminal**代わりにコマンドを使用して入力します。
- To perform node-level configuration on a particular leaf switch, you must first navigate to that switch using the **leaf** command.
- 物理ポートを指定するためのコマンド構文は若干異なります。たとえば、イーサネットポートはの**eth x/yethx/y**代わりに指定されます。
- 設定フィールドが、パスワードなどのユーザ定義のテキストで構成されている場合、Bashでの誤った解釈を避けるため、「\$」または「!」などの特殊文字はバックスラッシュ（「\」）でエスケープするか、単語または文字列全体を単一引用符で囲む必要があります。
- いくつかのコマンドショートカットは、Bash の動作により以下のように異なります。
 - **Ctrl-D** exits a session.
 - **Ctrl-Z** suspends a job.
- OSPF の設定**area route-map**は**area connectivity**、およびコマンドを追加します。

NX-OS Style CLI および APIC GUI の混合

基本的なモードは、Cisco APIC リリース 3.0 (1) 以降推奨されません。そのリリースにおいて GUI は 1 つだけです。



注意 NX-OS スタイル CLI を使用して実行された設定は、APIC GUI に表示されます。これらを表示できますが、時折 GUI で編集できない可能性があります。APIC GUI で行われた変更は、NX-OS スタイル CLI で表示できる可能性があります。部分的にのみ動作する可能性があります。次の例を参照してください。

- APIC でインターフェイスごとの設定を行う際に、GUI と CLI を混在させないでください。GUI で行われた設定が、NX-OS CLI では部分的にしか機能しない可能性があります。

たとえば、GUI の **[Tenants] > [tenant-name] > [Application Profiles] > [application-profile-name] > [Application EPGs] > [EPG-name] > [Static Ports] > [Deploy Static EPG on PC, VPC, or Interface]** でスイッチ ポートを設定したと仮定します。

次に NX-OS スタイルの CLI で `show running-config` コマンドを使用すると、以下のような出力を受信します。

```
leaf 102
interface ethernet 1/15
switchport trunk allowed vlan 201 tenant t1 application ap1 epg ep1
exit
exit
```

NX-OS スタイルの CLI でこれらのコマンドを使用してスタティック ポートを設定すると、次のエラーが発生します。

```
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 1/15
apic1(config-leaf-if)# switchport trunk allowed vlan 201 tenant t1 application ap1
epg ep1
No vlan-domain associated to node 102 interface ethernet1/15 encap vlan-201
```

これは、CLI に APIC GUI では実行されない検証があることが原因です。show running-config コマンドによって出力されたコマンドが NX-OS CLI で機能するためには、VLAN ドメインが事前に設定されている必要があります。設定の順序は GUI に適用されません。

このようなオブジェクトを削除する手順については、『*APIC Troubleshooting Guide*』の「*Troubleshooting Unwanted _ui_ Objects*」を参照してください。

レイヤ 3 外部接続の設定のモードについて

APIC は設定のための複数のユーザ インターフェイス (UI) をサポートしているので、1 つの UI を使用して設定を作成し、その後、別の UI を使用して設定を変更する場合は、予期しないインタラクションが潜んでいます。ここでは、さらに他の APIC のユーザ インターフェイスを使用した可能性がある場合、APIC NX-OS スタイルの CLI を使用してレイヤ 3 外部接続を設定するための考慮事項を説明します。

APIC NX-OS スタイルの CLI を使用してレイヤ 3 外部接続を設定する場合、次の 2 つのモードを選択することができます。

- よりシンプルな暗黙 モードは、APIC GUI または REST API と互換性がありません。
- 名前付き（または明示）モードは、APIC GUI および REST API と互換性があります。

いずれの場合も、設定は互換性がない UI では読み取り専用であると考えてください。

モードの違いについて

どちらのモードでも、構成設定は API の **l3extOut** クラスのインスタンスである内部コンテナオブジェクト「L3 Outside」（または「L3Out」）内で定義されます。2 つのモード間の主な違いは、このコンテナ オブジェクト インスタンスの命名にあります。

- 暗黙モード：コンテナのネーミングは潜在的であり、CLI コマンドには表示されません。CLI は、これらのオブジェクトを内部的に作成し保持します。
- 名前付きモード：名前はユーザーが決定します。名前付きモードの CLI コマンドには、追加の **l3Out** フィールドがあります。名前付き L3Out がを正常に設定され障害を回避するためには、ユーザーが外部レイヤ 3 用の API オブジェクト モデルを理解する必要があります。



(注) 「名前付きモードセクションを使用したレイヤ 3 外部接続の設定」セクションの手順を除き、このガイドでは、暗黙モードの手順を説明します。

注意事項および制約事項

- 同じ APIC インターフェイスでは、両方のモードを、次の制限でレイヤ 3 外部接続を設定するために一緒に使用することができます。テナント VRF、およびリーフの特定の組み合わせのレイヤ 3 外部接続設定は、1 つのモードを介してのみ実行できます。
- 特定のテナント VRF の場合、外部 L3 EPG を配置できるポリシー ドメインは、名前付きモードまたは暗黙モードのいずれかになります。推奨する設定方式は、特定のテナント VRF が、レイヤ 3 外部接続用に展開されたすべてのノード全体で、特定のテナント VRF の組み合わせに対して 1 つのモードだけを使用することです。モードは、異なるテナントまたは異なる VRF 全体で変えることができ、制限は適用されません。
- 外部レイヤ 3 機能は、次の例外を除いて、両方の設定モードでサポートされます
 - L4 ～ L7 サービス アプライアンスを使用したルーティング ピアリングとルート ヘルスインジェクション (RHI) は、名前付きモードでのみをサポートされます。名前付きモードは、ルーティング ピアリングが含まれるテナント VRF のすべての境界リーフスイッチ全体で使用する必要があります。
- 暗黙モード CLI 手順を使用して作成されたレイヤ 3 外部ネットワーク オブジェクト (**l3extOut**) は、「__ui_」で始まる名前で識別され、GUI で読み取り専用としてマークさ

れます。CLIは、インターフェイス、プロトコル、ルートマップ、EPGなどの機能で、これらの外部L3ネットワークを分割します。REST APIを介して実行される設定変更は、この構造を破棄することができ、CLIを介してさらなる変更を防ぐことができます。

このようなオブジェクトを削除する手順については、『*APIC Troubleshooting Guide*』の「*Troubleshooting Unwanted _ui_ Objects*」を参照してください。



第 2 章

ファブリックとインターフェイスの設定

- [ファブリックおよびインターフェイス設定 \(9 ページ\)](#)
- [グレースフル挿抜 \(GIR\) モード \(10 ページ\)](#)
- [NX-OS CLI を使用したリーフ ノードおよび FEX デバイス上の物理ポートの設定 \(12 ページ\)](#)
- [NX-OS CLI を使用したリーフ ノードおよび FEX デバイスのポートチャネルの設定 \(15 ページ\)](#)
- [NX-OS CLI を使用したリーフ ノードおよび FEX デバイスの仮想ポート チャネルの設定 \(22 ページ\)](#)
- [NX-OS スタイル CLI とプロファイルを使用して FEX 接続の設定 \(28 ページ\)](#)
- [リフレクティブ リレー \(802.1Qbg\) \(29 ページ\)](#)
- [インターフェイスのポリシー グループの設定 \(31 ページ\)](#)
- [インターフェイスのオーバーライドの設定 \(34 ページ\)](#)
- [エラー訂正の転送について \(36 ページ\)](#)

ファブリックおよびインターフェイス設定

ACI ファブリックを形成するために、Cisco Nexus 9000 シリーズ ACI モードスイッチは、APIC コントローラにより管理されるリーフおよびスパインのトポロジで展開されます。各リーフ ノードは、リーフ ノード間で接続のないすべてのスパイン ノードに接続されます。リーフ/スパイン型ノード間の相互接続リンクは、ファブリックリンクと呼ばれ、各ポートはファブリックポートと呼ばれます。ファブリックポートは、自動検出され、工場出荷時のデフォルト設定が、ファブリックの立ち上げ中に適用されるため、正常に動作するためのユーザ設定を必要としません。すべてのエンドポイント デバイスは、アクセスポートを通じてリーフ ノードに接続されます。アクセスポートは、NX-OS スイッチと同様に設定する必要があります。ファブリックおよびアクセスポートともに、NX-OS と同様に、インターフェイスで表されます。

リーフ/スパイン型ノードは、ACI モデルのさまざまなオブジェクトと見なされ、ポリシーのさまざまなセットをサポートします。CLI では、これらのノードはそれぞれ、リーフ/スパインとして表されますが、一般には両方ともノードと呼ばれます。リーフ/スパイン型ノード値は、ファブリックのすべてのポッド全体で一意です。FEX モジュールは、リーフ ノードに接続さ

れている場合、各リーフ内での一意のFEX ID 値を持っています。たとえば、2つのリーフ ノードに、それぞれ FEX 101 を付けることができます。



- (注) FEX ID 165 ～ 199 の FEX 接続の設定は、APIC GUI ではサポートされていません。これらの FEX ID のいずれかを使用するには、NX-OS スタイル CLI を使用してプロファイルを設定します。詳細については、「NX-OS スタイル CLI のインターフェイス プロファイルを使用して FEX 接続を設定する」を参照してください。

Cisco APIC としてリリース 3.0(1k) において FEX モジュールへの接続プロファイルとして設定できます。

リーフおよび FEX インターフェイスのインターフェイスの命名

ACI ファブリックでは、ほとんどのインターフェイス設定は、物理ポート、ポートチャネル、vPC（リーフ ノードに直接接続されているか、FEX モジュールを通じて接続されている）で実行されます。各インターフェイス タイプの一般的なコマンド構文を次の表に示します。

Interface Type	コマンド構文	例
Port	interface ethernet <i>slot/port</i>	interface eth 1/1
FEX Port	interface ethernet <i>fex-id/slot/port</i>	interface eth 101/1/1
Port-channel	interface port-channel <i>name</i>	interface port-channel foo
FEX Port-channel	interface port-channel <i>name</i> fex <i>fex-id</i>	interface port-channel foo flex 101
Virtual Port-channel (VPC)	interface vpc <i>name</i>	interface vpc foo
vPC over FEX	interface vpc <i>name</i> fex <i>fex-a</i> <i>fex-b</i>	interface vpc foo flex 101 102

グレースフル挿抜（GIR）モード

正常な挿入および削除（登場させ）モードでは、またはメンテナンス モードでは、最小サービスの中断でネットワークからのスイッチを特定するためにできます。登場させモードでトラフィックの影響を与えずにリアルタイムのデバッグを実行することができます。

グレースフル挿抜を使用してスイッチを正常に取り出し、そのスイッチをネットワークから分離して、デバッグ操作を実行することができます。スイッチは、最小限のトラフィックの中断だけで、通常の転送パスから取り外されます。デバッグ操作の実行が終了したら、グレースフル挿入を使用して、そのスイッチを完全な運用（通常）モードに戻すことができます。グレースフル削除では、ファブリックプロトコル（ISIS）以外のすべての外部プロトコルの電源が適切に切断され、スイッチがネットワークから分離されます。メンテナンス モード時に、ACI ファブリック内で最大メトリックが ISIS にアドバタイズされるため、メンテナンス モード TOR はスパインからのトラフィックに影響を与えません。さらに、前面パネルのすべてのイン

ターフェイスは、ファブリック インターフェイスを除いてスイッチでシャット ダウンされます。グレースフル挿入で、スイッチは自動的に廃棄、再起動、および再稼働されます。再稼働が完了したら、外部のすべてのプロトコルが復元され、ISIS の最大のメトリックが 10 分後にリセットされます。

次のプロトコルがサポートされています。

- Border Gateway Protocol (BGP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Intermediate System-to-Intermediate System (ISIS)
- Open Shortest Path First (OSPF)
- リンク集約制御プロトコル (LACP)

特記事項

- メンテナンスモードでスイッチをアップグレードまたはダウングレードすることはサポートされていません。
- スイッチがメンテナンス モードの場合、イーサネット ポート モジュールは通知に関連するインターフェイスの伝達を停止します。その結果、この時点でリモートスイッチを再起動するかファブリック リンクがフラップされる場合、ファブリック リンクはスイッチを手動で再起動 (acidiag touch clean コマンドまで)、廃棄、再稼働するまで起動しません。
- マルチ ポッドの再配布されたルートの ISIS メトリック を 63 以下に設定する必要があります。再配布されたルートの ISIS メトリック を選択するには、[ファブリック]>[ファブリック ポリシー]>[ポッド ポリシー]>[ISIS ポリシー]を選択します。
- 既存の GIR はすべてのレイヤ 3 トラフィック迂回をサポートしています。LACP でレイヤ 2 のすべてのトラフィックは、冗長ノードを迂回します。ノードがメンテナンス モードに入ると、ノードで実行している LACP がすぐにポート チャネルの一部として集約される必要がなくなったネイバーに通知します。すべてのトラフィックは vPC ピア ノードを迂回されます。

CLI を使用してメンテナンス モードにスイッチを移行する

CLI を使用してメンテナンス モードにスイッチを移行するには、次の手順を使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	[no]debug-switch <i>node_id</i> or <i>node_name</i>	メンテナンス モードにスイッチを移行します。

CLI を使用して操作モードにスイッチを挿入する

この手順を使って、スイッチを CLI を使用している動作モードに挿入します。

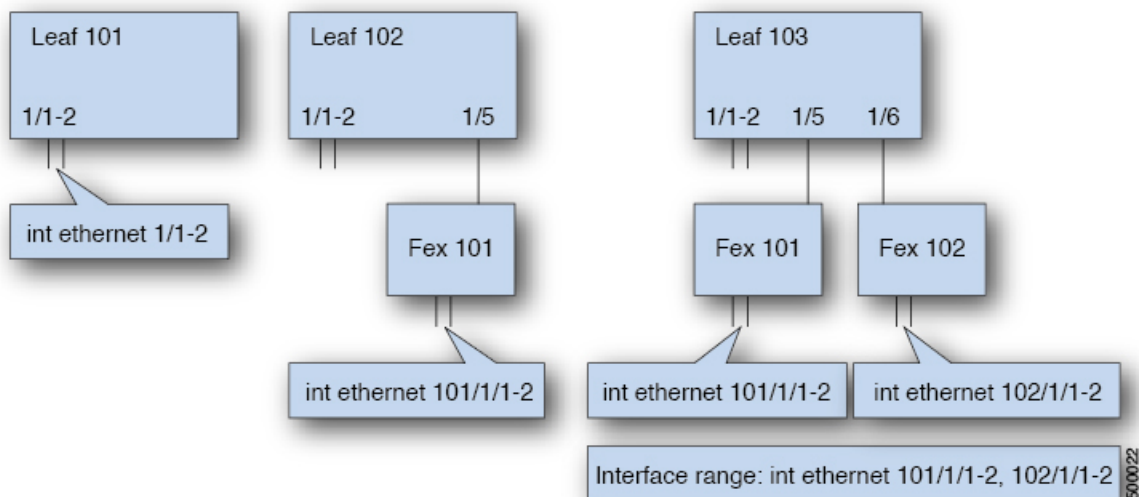
手順

	コマンドまたはアクション	目的
ステップ 1	<code>[no]no debug-switch node_id or node_name</code>	動作モードにスイッチを挿入します。

NX-OS CLI を使用したリーフ ノードおよび FEX デバイス上の物理ポートの設定

次の例のコマンドでは、REST API/SDK および GUI と完全な互換性がある多数の管理対象オブジェクト（MO）を ACI ポリシー モデルで作成します。ただし、CLI ユーザは ACI モデル内部ではなく意図したネットワーク設定に注力できます。

次の図に、リーフ ノードに直接のイーサネット ポート、またはリーフ ノードに接続された FEX モジュールの例と、CLI でそれぞれがどのように表示されるのかを示します。FEX ポートでは、*fex-id* はポート自体の名前に **ethernet 101/1/1** として含まれます。インターフェイス範囲を記述する際は、**ethernet** キーワードを NX-OS で繰り返す必要はありません。例：**interface ethernet 101/1/1-2, 102/1/1-2**。



- リーフ ノードの ID 番号はグローバルです。
- *fex-id* 番号は各リーフにローカルです。
- キーワード **ethernet** の後のスペースに注意してください。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 102	設定するリーフを指定します（複数可）。 <i>node-id</i> には、設定の適用対象となる単一のノード ID、または ID の範囲を <i>node-id1-node-id2</i> という形式で指定できます。
ステップ 3	interface type 例 : apic1(config-leaf)# interface ethernet 1/2	設定するインターフェイスを指定します。インターフェイス タイプと ID を指定できます。イーサネット ポートの場合は、「ethernet slot / port」を使用します。
ステップ 4	（任意） fex associate node-id 例 : apic1(config-leaf-if)# fex associate 101	設定するインターフェイスが FEX インターフェイスの場合、このコマンドを使用して、設定前に FEX モジュールをリーフ ノードに接続する必要があります。 （注） この手順は FEX ポートを使用してポートチャネルを作成する前に行う必要があります。
ステップ 5	speed speed 例 : apic1(config-leaf-if)# speed 10G	ここでの速度設定は一例です。ここでは、以下の表に示す任意のインターフェイス設定を設定できます。

次の表に、この時点で設定できるインターフェイス設定を示します。

コマンド	目的
[no] shut	物理インターフェイスをシャットダウンします
[no] speed <i>speedValue</i>	物理インターフェイスの速度を設定します
[no] link debounce time <i>time</i>	リンク でバウンスを設定します
[no] negotiate auto	ネゴシエートを設定します
[no] cdp enable	Cisco Discovery Protocol (CDP) を無効または有効にします

コマンド	目的
[no] mcp enable	Mis-Cabling Protocol (MCP) を無効または有効にします
[no] lldp transmit	物理インターフェイスの送信を設定します
[no] lldp receive	物理インターフェイスの LLDP 受信を設定します
spanning-tree {bpduguard bpdupfilter} {enable disable}	スパンニング ツリー BPDU を設定します
[no] storm-control level <i>percentage</i> [burst-rate <i>percentage</i>]	ストーム制御 (パーセント) を設定します
[no] storm-control pps <i>packets-per-second</i> burst-rate <i>packets-per-second</i>	ストーム制御 (秒当たりのパケット) を設定します

例

リーフ ノードに 1 つのポートを設定します。次に、プロパティ **speed**、**cdp**、および **admin state** についてリーフ 101 のインターフェイス **eth1/2** を設定する例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/2
apic1(config-leaf-if)# speed 10G
apic1(config-leaf-if)# cdp enable
apic1(config-leaf-if)# no shut
```

複数のリーフ ノードの複数のポートを設定します。次に、リーフ ノード 101 ~ 103 のそれぞれのインターフェイス **eth1/1-10** での速度設定の例を示します。

```
apic1(config)# leaf 101-103
apic1(config-leaf)# interface eth 1/1-10
apic1(config-leaf-if)# speed 10G
```

リーフ ノードに FEX を接続します。次に、リーフ ノードに FEX モジュールを接続する例を示します。NX-OS とは異なり、リーフ ポート **Eth1/5** は暗黙的にファブリックポートとして設定され、FEX ファブリック ポートチャネルは FEX アップリンク ポートで内部で作成されます。ACI では、FEX ファブリック ポートチャネルはデフォルト設定を使用し、ユーザ設定は使用できません。



(注) 次の例に示すように、この手順は FEX ポートを使用してポートチャネルを作成する前に行う必要があります。

```
apicl(config)# leaf 102
apicl(config-leaf)# interface eth 1/5
apicl(config-leaf-if)# fex associate 101
```

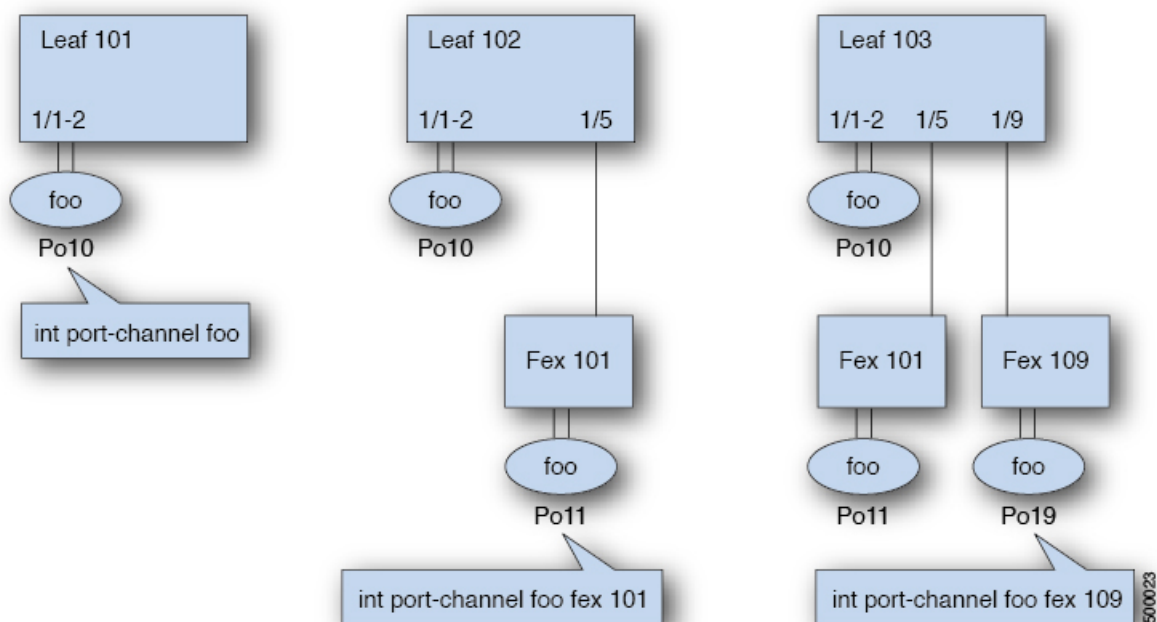
リーフ ノードに接続した FEX ポートを設定します。次に、リーフ ノード 102 ~ 103 のそれぞれに接続した FEX モジュール 101 のインターフェイス eth1/1-10 での速度設定の例を示します。FEX ID 101 はポート ID に含まれています。FEX ID は 101 から始まり、リーフに対してローカルです。

```
apicl(config)# leaf 102-103
apicl(config-leaf)# interface eth 101/1/1-10
apicl(config-leaf-if)# speed 1G
```

NX-OS CLI を使用したリーフ ノードおよび FEX デバイスのポートチャネルの設定

ポートチャネルは NX-OS の論理インターフェイスです。これは、複数の物理ポートのために帯域幅を集約するだけでなく、リンク障害時の冗長性を確保する目的でも使用されます。NX-OS におけるポートチャネル インターフェイスは、ノード内では一意となる、1 ~ 4096 の範囲でユーザが指定した番号によって識別されます。ポートチャネル インターフェイスは、(**interface port-channel** コマンドを使用して) 明示的に設定するか、または (**channel-group** コマンドを使用して) 暗黙的に作成します。ポートチャネル インターフェイスの設定は、ポートチャネルのすべてのメンバー ポートに適用されます。特定の互換性パラメータ（速度など）は、メンバーポートでは設定できません。

ACI モデルでは、ポートチャネルは論理エンティティとして設定され、1 つ以上のリーフ ノードでポートセットに割り当てることができるポリシーのコレクションを表す名前によって識別されます。このような割り当てによって各リーフ ノードにポートチャネル インターフェイスが 1 個作成されます。これは、リーフ ノード内の 1 ~ 4096 の範囲で自動生成される番号によって識別されます。同じポートチャネル名を持つノード間では、番号を同じにすることも、分けることもできます。これらのポートチャネルのメンバーシップも同様に、同じにすることも分けることもできます。FEX ポート上にポートチャネルが作成されるときは、同じポートチャネル名を使用して、リーフ ノードに接続されている各 FEX デバイスに対して 1 つのポートチャネル インターフェイスを作成することができます。したがって、N 個の FEX モジュールに接続されている各リーフ ノードには最大で N+1 個の一意のポートチャネル インターフェイス（自動生成されるポートチャネル番号で識別される）を作成できます。これは以下の例で説明します。FEX ポートのポートチャネルは、*fex-id* とポートチャネル名を指定することによって識別されます（例：interface port-channel foo fex 101）。



- 各リーフが N 個の FEX ノードに接続されているときは、ポートチャネル foo のリーフごとに N+1 個のインスタンスが可能です。
- リーフ ポートおよび FEX ポートを同じポートチャネルインスタンスの一部にすることはできません。
- 各 FEX ノードはポートチャネル foo のインスタンスを 1 つだけ持つことができます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	template port-channel channel-name 例 : apicl(config)# template port-channel foo	新しいポートチャネルを作成するか、既存のポートチャネルを設定します（グローバル コンフィギュレーション）。
ステップ 3	[no] switchport access vlan vlan-id tenant tenant-name application application-name epg epg-name 例 : apicl(config-po-ch-if)# switchport	ポート チャネルを関連付けるすべてのポート上に VLAN を持つ EPG を導入します。

	コマンドまたはアクション	目的
	<code>access vlan 4 tenant ExampleCorp application Web epg webEpg</code>	
ステップ 4	channel-mode active 例 : <pre>apic1(config-po-ch-if) # channel-mode active</pre> (注) 対称ハッシュを有効にするには、 lACP symmetric-hash コマンドを入力します。 <pre>apic1(config-po-ch-if) # lACP symmetric-hash</pre>	(注) channel-mode コマンドは、NX-OS の channel-group コマンドの mode オプションに相当します。ただし、ACI ではこれは (メンバー ポートではなく) ポートチャネルでサポートされます。 対称ハッシュは、次のスイッチではサポートされていません。 • Cisco Nexus 93128TX • Cisco Nexus 9372PX • Cisco Nexus 9372PX-E • Cisco Nexus 9372TX • Cisco Nexus 9372TX-E • Cisco Nexus 9396PX • Cisco Nexus 9396TX
ステップ 5	exit 例 : <pre>apic1(config-po-ch-if) # exit</pre>	設定モードに戻ります。
ステップ 6	leaf node-id 例 : <pre>apic1(config) # leaf 101</pre>	設定するリーフスイッチを指定します。 <i>node-id</i> には、設定の適用対象となる単一のノード ID、または ID の範囲を <i>node-id1-node-id2</i> という形式で指定できます。
ステップ 7	interface type 例 : <pre>apic1(config-leaf) # interface ethernet 1/1-2</pre>	ポートチャネルに設定するインターフェイスまたはインターフェイスの範囲を指定します。
ステップ 8	[no] channel-group channel-name 例 : <pre>apic1(config-leaf-if) # channel-group foo</pre>	インターフェイスまたはインターフェイスの範囲をポートチャネルに割り当てます。ポートチャネルからインターフェイスを削除するには、キーワード no を使用します。インターフェイス上からポートチャネルの割り当てを変更する場合

	コマンドまたはアクション	目的
		は、以前のポートチャネルからインターフェイスを最初に削除することなく channel-group コマンドを入力することができます。
ステップ 9	(任意) lacp port-priority priority 例 : <pre>apicl(config-leaf-if)# lacp port-priority 1000 apicl(config-leaf-if)# lacp rate fast</pre>	この設定とその他のポート単位の LACP プロパティは、この時点でポートチャネルのメンバー ポートに適用できます。 (注) ACI モデルでは、これらのコマンドはポートがポート チャネルのメンバーになった後でのみ使用できます。ポートがポート チャネルから削除された場合、これらのポート単位のプロパティの設定も削除されます。

次の表に、ACI モデルでポート チャネル プロパティのグローバル コンフィギュレーションを行うためのさまざまなコマンドを示します。これらのコマンドは、(config-leaf-if) CLI モードで特定のリーフのポートチャネルのオーバーライドを設定するためにも使用できます。ポートチャネル上から行った設定は、すべてのメンバー ポートに適用されます。

CLI 構文	機能
[no] speed <speedValue>	ポートチャネルの速度を設定します
[no] link debounce time <time>	ポートチャネルのリンク デバウンスを設定します
[no] negotiate auto	ポートチャネルのネゴシエートを設定します
[no] cdp enable	ポートチャネルの CDP を無効または有効にします
[no] mcp enable	ポートチャネルの MCP を無効または有効にします
[no] lldp transmit	ポートチャネルの送信を設定します
[no] lldp receive	ポートチャネルの LLDP 受信を設定します
spanning-tree <bpduguard bpdufilter> <enable disable>	スパンニング ツリー BPDU を設定します
[no] storm-control level <percentage> [burst-rate <percentage>]	ストーム制御 (パーセント) を設定します

CLI 構文	機能
[no] storm-control pps <packet-per-second> burst-rate <packets-per-second>	ストーム制御（秒当たりのパケット）を設定します
[no] channel-mode { active passive on mac-pinning }	ポートチャネルのリンクの LACP モードを設定します
[no] lacp min-links <value>	リンクの最小数を設定します
[no] lacp max-links <value>	リンクの最大数を設定します
[no] lacp fast-select-hot-standby	ホットスタンバイ ポートの LACP 高速セレクトを設定します
[no] lacp graceful-convergence	LACP グレースフル コンバージェンスを設定します
[no] lacp load-defer	LACP ロード遅延メンバー ポートを設定します
[no] lacp suspend-individual	LACP 個別ポートの中断を設定します
[no] lacp port-priority	LACP ポート プライオリティ
[no] lacp rate	LACP レートを設定します

例

ポート チャネル（グローバル コンフィギュレーション）を設定します。速度およびチャネルモードの2つの設定を含むポリシーのコレクションを表す論理エンティティ「foo」を作成します。必要に応じてより多くのプロパティを設定できます。



(注) channel mode コマンドは、NX-OS の channel group コマンドの mode オプションに相当します。ただし、ACI ではこれは（メンバー ポートではなく）ポートチャネルでサポートされます。

```
apic1(config)# template port-channel foo
apic1(config-po-ch-if)# switchport access vlan 4 tenant ExampleCorp application Web ep
webEpg
apic1(config-po-ch-if)# speed 10G
apic1(config-po-ch-if)# channel-mode active
```

FEX のポートチャネルにポートを設定します。この例では、ポート チャネル foo はリーフ ノード 102 に接続されている FEX 101 のポートイーサネット 1/1-2 に割り当てられ、ポートチャネル foo のインスタンスを作成します。リーフ ノードは番号（例えば 1002）を自動生成し、スイッチのポート チャネルを識別します。このポート チャ

ネル番号は、作成されたポート チャネル foo のインスタンス数とは無関係で、リーフ ノード 102 に固有のものです。



- (注) リーフ ノードに FEX モジュールを接続する設定は、FEX ポートを使用してポート チャネルを作成する前に実行する必要があります。

```
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 101/1/1-2
apic1(config-leaf-if)# channel-group foo
```

リーフ 102 では、このポート チャネル インターフェイスを interface port-channel foo FEX 101 で呼ぶこともできます。

```
apic1(config)# leaf 102
apic1(config-leaf)# interface port-channel foo fex 101
apic1(config-leaf)# shut
```

複数のリーフ ノードでポートチャネルにポートを設定します。この例におけるポート チャネル foo は、101 ~ 103 の各リーフ ノード内にあるイーサネット 1/1-2 ポートに割り当てられます。リーフ ノードは各ノードで固有の番号（ノード間で同一または分けることができる）を自動生成し、ポートチャネル インターフェイスとして機能します。

```
apic1(config)# leaf 101-103
apic1(config-leaf)# interface ethernet 1/1-2
apic1(config-leaf-if)# channel-group foo
```

ポートチャネルにメンバーを追加します。この例では、各リーフ ノードのポートチャネルに 2 つのメンバー eth1/3-4 を追加し、各ノードのポートチャネル foo がメンバー eth 1/1-4 を持つようにします。

```
apic1(config)# leaf 101-103
apic1(config-leaf)# interface ethernet 1/3-4
apic1(config-leaf-if)# channel-group foo
```

ポートチャネルからメンバーを削除します。この例は、各リーフ ノードでポートチャネル foo から 2 つのメンバー eth1/2、eth1/4 を削除し、各ノードのポート チャネル foo がメンバー eth 1/1、eth1/3 を持つようにします。

```
apic1(config)# leaf 101-103
apic1(config-leaf)# interface eth 1/2,1/4
apic1(config-leaf-if)# no channel-group foo
```

複数のリーフ ノードで異なるメンバーを持つポートチャネルを設定します。次に、同じポートチャネル foo ポリシーを使用して、リーフごとにメンバー ポートが異なる複数のリーフ ノードでポートチャネル インターフェイスを作成する例を示します。リーフ ノードのポートチャネル番号は、同じポートチャネル foo に対して同一にすることも別々にすることもできます。ただし CLI では、設定は interface port-channel foo で参

照されます。FEX ポートにポートチャネルが設定されている場合は、`interface port-channel foo fex <fex-id>` で参照されます。

```
apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/1-2
apicl(config-leaf-if)# channel-group foo
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
apicl(config)# leaf 102
apicl(config-leaf)# interface ethernet 1/3-4
apicl(config-leaf-if)# channel-group foo
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
apicl(config)# leaf 103
apicl(config-leaf)# interface ethernet 1/5-8
apicl(config-leaf-if)# channel-group foo
apicl(config-leaf-if)# exit
apicl(config-leaf)# interface ethernet 101/1/1-2
apicl(config-leaf-if)# channel-group foo
```

LACP のポート単位のプロパティを設定します。次に、LACP のポート単位のプロパティについてポートチャネルのメンバー ポートを設定する例を示します。



(注) ACI モデルでは、これらのコマンドはポートがポートチャネルのメンバーになった後でのみ使用できます。ポートがポートチャネルから削除された場合、これらポート単位のプロパティ設定も削除されます。

```
apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/1-2
apicl(config-leaf-if)# channel-group foo
apicl(config-leaf-if)# lacp port-priority 1000
apicl(config-leaf-if)# lacp rate fast
```

ポートチャネルの管理状態を設定します。この例におけるポートチャネル `foo` は、`channel-group` コマンドを使用することで、101 ~ 103 の各リーフ ノードに対して設定されます。ポートチャネルの管理状態はポートチャネルインターフェイスを使用してリーフごとに設定できます。ACI モデルでは、ポートチャネルの管理状態をグローバル スコープで設定することはできません。

```
// create port-channel foo in each leaf
apicl(config)# leaf 101-103
apicl(config-leaf)# interface ethernet 1/3-4
apicl(config-leaf-if)# channel-group foo

// configure admin state in specific leaf
apicl(config)# leaf 101
apicl(config-leaf)# interface port-channel foo
apicl(config-leaf-if)# shut
```

オーバーライド設定は、他のプロパティを共有しながら各リーフのポートチャネルインターフェイスに特定の VLAN ドメインを割り当てる場合などにとっても便利です。

```
// configure a port channel global config
apicl(config)# interface port-channel foo
```

```

apic1(config-if)# speed 1G
apic1(config-if)# channel-mode active

// create port-channel foo in each leaf
apic1(config)# leaf 101-103
apic1(config-leaf)# interface ethernet 1/1-2
apic1(config-leaf-if)# channel-group foo

// override port-channel foo in leaf 102
apic1(config)# leaf 102
apic1(config-leaf)# interface port-channel foo
apic1(config-leaf-if)# speed 10G
apic1(config-leaf-if)# channel-mode on
apic1(config-leaf-if)# vlan-domain dom-foo

```

次の例では、**channel-group** コマンドを使用することで、ポートのポートチャンネル割り当てを変更します。他のポートチャンネルに割り当てる前にポートチャンネルのメンバーシップを削除する必要はありません。

```

apic1(config)# leaf 101-103
apic1(config-leaf)# interface ethernet 1/3-4
apic1(config-leaf-if)# channel-group foo
apic1(config-leaf-if)# channel-group bar

```

NX-OS CLI を使用したリーフ ノードおよび FEX デバイスの仮想ポート チャンネルの設定

仮想ポートチャンネル（VPC）は、2つのアップストリームのリーフ ノードへのホストまたはスイッチの接続を可能にして帯域幅の使用率と可用性を向上できる、ポートチャンネルの拡張機能です。NX-OS では、VPC 設定は2つのアップストリーム スイッチ内でそれぞれ行い、スイッチ間のピア リンクを使用して設定が同期されます。



(注) 2つのリーフ スイッチ間での VPC ドメインを作成するとき、同じスイッチの生成を次のいずれかのどちらのスイッチも必要があります。

- 1 - Cisco Nexus N9K スイッチで、スイッチの名前の末尾には、「EX」なしの生成たとえば、N9K 9312TX
- 2: Cisco Nexus N9K スイッチがスイッチ モデル名の最後の「ex」の生成たとえば、N9K-93108TC-EX

スイッチなど、これらの2つが互換性のある VPC ピアではありません。代わりに、同じ世代のスイッチを使用します。

ACI モデルはピア リンクを必要とせず、VPC 設定は両方のアップストリームのリーフ ノードに対してグローバルに行うことができます。ACI では **vpc context** と呼ばれるグローバル設定

モードが導入されており、VPC インターフェイスは両方のリーフ ノードにグローバル設定を適用可能にする **interface vpc** タイプを使用して表されます。

ACI モデルの vPC では、リーフ ポートを使用する vPC と、FEX ポート上の vPC という 2 つの異なるトポロジがサポートされます。リーフ ノードのペア間では多数の VPC インターフェイスを作成することができます。同様に、ストレート型トポロジのリーフ ノード ペアに接続された FEX モジュールのペア間でも多数の VPC インターフェイスを作成できます。

VPC の考慮事項は次のとおりです。

- 使用する VPC 名はリーフ ノードペア間で一意です。たとえば、リーフ ペア（FEX 有り、または無し）ごとに作成できる「corp」という VPC は 1 つだけです。
- リーフ ポートおよび FEX ポートを同じ VPC に混在させることはできません。
- 各 FEX モジュールは、単一の VPC corp インスタンスにのみ含めることができます
- VPC コンテキストにより設定できます。
- VPC context モードでは、特定のリーフ ペアについて、あらゆる VPC の設定を行うことができます。FEX 上の VPC では、次の 2 つの互換性のある例に示すように、*fex-id* ペアを VPC コンテキストに対して、または VPC インターフェイスと共に指定する必要があります。

```
(config)# vpc context leaf 101 102
(config-vpc)# interface vpc Reg fex 101 101
```

または

```
(config)# vpc context leaf 101 102 fex 101 101
(config-vpc)# interface vpc Reg
```

ACI モデルでは、VPC 設定は（以下の例に示すように）次の手順で行います。



(注) VLAN ドメインは、VLAN の範囲で必要です。ポート チャネルのテンプレートに関連付けられている必要があります。

1. VLAN の範囲で VLAN ドメイン構成 (グローバル設定)
2. VPC ドメイン設定 (グローバル設定)
3. ポート チャネルのテンプレートの設定 (グローバル設定)
4. ポート チャネルのテンプレートを VLAN ドメインに関連付ける
5. VPC のポートチャネル設定 (グローバル設定)
6. リーフ ノードの VPC にポートの設定
7. vpc context の VPC の L2、L3 の設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>apic1# configure</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	vlan-domainname[dynamic] [type domain-type] 例 : <pre>apic1(config)# vlan-domain dom1 dynamic</pre>	仮想ポートチャネルの VLAN ドメインの設定 (ポート チャネルのテンプレートとここ)。
ステップ 3	vlanrange 例 : <pre>apic1(config-vlan)# vlan 1000-1999 apic1(config-vlan)# exit</pre>	VLAN ドメインの VLAN の範囲を設定し、configuration mode(設定モード、コンフィギュレーション モード)を終了します。単一の VLAN または複数の VLAN 範囲を設定できます。
ステップ 4	vpc domain explicit domain-id leaf node-id1 node-id2 例 : <pre>apic1(config)# vpc domain explicit 1 leaf 101 102</pre>	リーフ ノードのペア間の VPC ドメインを設定します。リーフ ノードのペアとともに明示モードで VPC ドメイン ID を指定できます。 VPC ドメインを設定するための代替コマンドは次のとおりです。 • vpc domain [consecutive reciprocal] consecutive オプションと reciprocal オプションにより、ACI ファブリックのすべてのリーフ ノードで VPC ドメインの自動設定を行うことができます。 • vpc domain consecutive domain-start leaf start-node end-node このコマンドは、選択した一連のリーフ ノードペアに対して連続的に VPC ドメインを設定します。
ステップ 5	peer-dead-interval interval 例 : <pre>apic1(config-vpc)# peer-dead-interval 10</pre>	リーフ スイッチは、ピアから応答を受信する前に、vPC を復元するまで待機する時間の遅延を設定します。この時間内ピアから応答を受信するはないとリーフ スイッチ、ピアを停止するいると見なすをマスターとしての役割を持

	コマンドまたはアクション	目的
		つ vPC 始動します。ピアから応答を受信するとその時点で、vPC を復元します。範囲は 5 ～ 600 秒です。デフォルトは 200 秒です。
ステップ 6	exit 例 : <pre>apicl(config-vpc) # exit</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 7	template port-channel <i>channel-name</i> 例 : <pre>apicl(config) # template port-channel corp</pre>	新しいポートチャンネルを作成するか、既存のポートチャンネルを設定します (グローバル コンフィギュレーション)。 すべての VPC は各リーフ ペアのポートチャンネルとして設定されます。同じ VPC のリーフ ペアには同じポートチャンネル名を使用する必要があります。このポートチャンネルを使用すると、リーフ ノードの 1 つ以上のペア間の VPC を作成できます。この VPC のインスタンスは各リーフ ノードに 1 つだけになります。
ステップ 8	vlan-domain member <i>vlan-domain-name</i> 例 : <pre>vlan-domain member dom1</pre>	以前に設定された VLAN ドメインには、ポートチャンネルのテンプレートを関連付けます。
ステップ 9	switchport access vlan <i>vlan-id</i> tenant <i>tenant-name</i> application <i>application-name</i> epg <i>epg-name</i> 例 : <pre>apicl(config-po-ch-if) # switchport access vlan 4 tenant ExampleCorp application Web epg webEpg</pre>	ポートチャンネルを関連付けるすべてのポート上に VLAN を持つ EPG を導入します。
ステップ 10	channel-mode active 例 : <pre>apicl(config-po-ch-if) # channel-mode active</pre>	(注) ポートチャンネルは VPC の アクティブチャンネルモードである必要があります。
ステップ 11	exit 例 : <pre>apicl(config-po-ch-if) # exit</pre>	設定モードに戻ります。

	コマンドまたはアクション	目的
ステップ 12	leaf node-id1 node-id2 例 : <pre>apic1(config)# leaf 101-102</pre>	設定するリーフ スイッチのペアを指定します。
ステップ 13	interface typeleaf/interface-range 例 : <pre>apic1(config-leaf)# interface ethernet 1/3-4</pre>	ポート チャネルに設定するインターフェイスまたはインターフェイスの範囲を指定します。
ステップ 14	[no] channel-group channel-name vpc 例 : <pre>apic1(config-leaf-if)# channel-group corp vpc</pre>	インターフェイスまたはインターフェイスの範囲をポートチャネルに割り当てます。ポートチャネルからインターフェイスを削除するには、キーワード no を使用します。インターフェイス上からポートチャネルの割り当てを変更する場合は、以前のポートチャネルからインターフェイスを最初に削除することなく channel-group コマンドを入力することができます。 (注) このコマンドの vpc キーワードによってポートチャネルが VPC になります。VPC がまだ存在しない場合は、VPC ID が自動的に生成され、すべてのメンバリーフ ノードに適用されます。
ステップ 15	exit 例 : <pre>apic1(config-leaf-if)# exit</pre>	
ステップ 16	exit 例 : <pre>apic1(config-leaf)# exit</pre>	
ステップ 17	vpc context leaf node-id1 node-id2 例 : <pre>apic1(config)# vpc context leaf 101 102</pre>	vpc context モードでは、VPC の設定を両方のリーフ ノードペアに適用できます。
ステップ 18	interface vpc channel-name 例 :	

	コマンドまたはアクション	目的
	<code>apicl(config-vpc)# interface vpc blue fex 102 102</code>	
ステップ 19	(任意) <code>[no] shutdown</code> 例 : <code>apicl(config-vpc-if)# no shut</code>	vpc context の管理状態の設定では、両方のリーフノードに対して 1 つのコマンドで VPC の管理状態を変更できます。

例

次に、基本的な VPC を設定する例を示します。

```
apicl# configure
apicl(config)# vlan-domain dom1 dynamic
apicl(config-vlan)# vlan 1000-1999
apicl(config-vlan)# exit
apicl(config)# vpc domain explicit 1 leaf 101 102
apicl(config-vpc)# peer-dead-interval 10
apicl(config-vpc)# exit
apicl(config)# template port-channel corp
apicl(config-po-ch-if)# vlan-domain member dom1
apicl(config-po-ch-if)# channel-mode active
apicl(config-po-ch-if)# exit
apicl(config)# leaf 101-102
apicl(config-leaf)# interface ethernet 1/3-4
apicl(config-leaf-if)# channel-group corp vpc
apicl(config-leaf-if)# exit
apicl(config)# vpc context leaf 101 102
```

次に、FEX ポートを使用して VPC を設定する例を示します。

```
apicl(config-leaf)# interface ethernet 101/1/1-2
apicl(config-leaf-if)# channel-group Reg vpc
apicl(config)# vpc context leaf 101 102
apicl(config-vpc)# interface vpc corp
apicl(config-vpc-if)# exit
apicl(config-vpc)# interface vpc red fex 101 101
apicl(config-vpc-if)# switchport
apicl(config-vpc-if)# exit
apicl(config-vpc)# interface vpc blue fex 102 102
apicl(config-vpc-if)# shut
```

NX-OS スタイル CLI とプロファイルを使用して FEX 接続の設定

NX-OS スタイル CLI を使用してリーフ ノードへの接続を FEX を設定するには、次の手順を使用します。



(注) FEX Id を持つ FEX 接続を構成する 165 に 199 APIC GUI ではサポートされていません。これらの FEX Id のいずれかを使用するには、次のコマンドを使用して、プロファイルを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf-interface-profile name 例 : <code>apic1(config)# leaf-interface-profile fexIntProf1</code>	設定するリーフ インターフェイス プロファイルを指定します。
ステップ 3	leaf-interface-group name 例 : <code>apic1(config-leaf-if-profile)# leaf-interface-group leafIntGrp1</code>	設定するインターフェイス グループを指定します。
ステップ 4	fex associate fex-id [template template-typefex-template-name] 例 : <code>apic1(config-leaf-if-group)# fex associate 101</code>	リーフ ノードに FEX モジュールを接続します。使用するテンプレートを指定するのにオプションのテンプレートのキーワードを使用します。存在しない場合、システムは、名前とタイプが指定したで、テンプレートを作成します。

例

このマージの例では、ID 101 で FEX 接続のリーフ インターフェイス プロファイルを設定します。

```
apic1# configure
apic1(config)# leaf-interface-profile fexIntProf1
```



```
apic1(config-leaf-if-profile)# leaf-interface-group leafIntGrp1
apic1(config-leaf-if-group)# fex associate 101
```

リフレクティブ リレー (802.1Qbg)

リフレクティブ リレーでは、Cisco APIC リリース 2.3(1) でスイッチング オプションの開始時刻です。リフレクティブ リレー: IEEE 標準 802.1Qbg のタグのないアプローチ: ポリシーを適用し、必要に応じて、宛先またはターゲット VM サーバ上にトラフィックを送信する外部のスイッチへのすべてのトラフィックを転送します。ローカルスイッチングはありません。ブロードキャストまたはマルチキャスト トラフィックは、リフレクティブ リレーは、各 VM サーバでローカルにパケットのレプリケーションを提供します。

リフレクティブ リレーの利点の 1 つは、スイッチング機能および管理機能、Vm をサポートするサーバリソースを解放するための外部スイッチを活用しています。リフレクティブ リレーでは、ポリシー、同じサーバ上の Vm の間のトラフィックに適用する Cisco APIC で設定することもできます。

Cisco ACI、入ってきたのと同じポートからオンに戻すにトラフィックを許可する、リフレクティブ リレーを有効にできます。APIC GUI、NX-OS CLI または REST API を使用して、レイヤ 2 インターフェイス ポリシーとして **individual ports**(個々のポート、個別ポート)、ポート チャネルまたは仮想ポート チャネルでリフレクティブ リレーを有効にすることができます。この機能はデフォルトではディセーブルになっています。

用語 仮想イーサネット ポートのためのアグリゲータ 802.1Qbg を説明する (VEPA) が使用されるも機能します。

リフレクティブ リレーのサポート

リフレクティブ リレーには、次のサポートされています。

- IEEE 標準 802.1Qbg タグのないアプローチ、リフレクティブ リレーとも呼ばれます。

Cisco APIC 2.3(1) リリースのリリースは IEE 標準 802.1Qbg をサポートしていませんマルチチャネル テクノロジーと S タグ付きアプローチです。

- 物理ドメイン。

仮想ドメインはサポートしていません。

- 物理ポートは、ポート チャネル (Pc)、および仮想ポート チャネル (Vpc)。

シスコ ファブリック エクステンダ (FEX) とブレードサーバはサポートしていません。リフレクティブ リレーはサポートされていないインターフェイスで有効になっていると、障害が発生すると、最後の有効な設定が保持されます。ポートでリフレクティブ リレーを無効にすると、障害をクリアします。

- Cisco Nexus 9000 シリーズのスイッチと EX または FX 、モデル名の最後にします。

NX-OS は、CLI を使用してリフレクティブ リレーの有効化

; By default(デフォルトで、デフォルトでは) リフレクティブ リレーが無効になっていますただし、スイッチのレイヤ 2 インターフェイス ポリシーとして、ポート、またはポート チャネルまたは仮想ポート チャネルでこれを有効にできます。CLI では、NX-OS テンプレートを使用して、複数のポートでリフレクティブ リレーの有効化または individual ports(個々のポート、個別ポート) で有効にすることができます。

始める前に

この手順では、Cisco Application Centric Infrastructure (ACI) ファブリックを設定し、物理スイッチをインストールしてあることを前提としています。

手順

リフレクティブ リレー 1 つまたは複数のポートで有効にします。

例：

この例では、1 つのポートでリフレクティブ リレーが有効にします。

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/2
apic1(config-leaf-if)# switchport vepa enabled
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

例：

この例では、リフレクティブ リレー、テンプレートを使用して複数のポートで有効にします。

```
apic1(config)# template policy-group grp1
apic1(config-pol-grp-if)# switchport vepa enabled
apic1(config-pol-grp-if)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/2-4
apic1(config-leaf-if)# policy-group grp1
```

例：

この例では、ポート チャネルでリフレクティブ リレーが有効にします。

```
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po2
apic1(config-leaf-if)# switchport vepa enabled
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)#
```

例：

この例では、複数のポート チャネルでリフレクティブ リレーが有効にします。

```
apic1(config)# template port-channel pol
apic1(config-if)# switchport vepa enabled
apic1(config-if)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/3-4
apic1(config-leaf-if)# channel-group pol
```

```
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

例：

この例では、仮想ポート チャンネルでリフレクティブ リレーが有効にします。

```
apic1(config)# vpc domain explicit 1 leaf 101 102
apic1(config-vpc)# exit
apic1(config)# template port-channel po4
apic1(config-if)# exit
apic1(config)# leaf 101-102
apic1(config-leaf)# interface eth 1/11-12
apic1(config-leaf-if)# channel-group po4 vpc
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# interface vpc po4
apic1(config-vpc-if)# switchport vepa enabled
```

インターフェイスのポリシー グループの設定

In data center networks, typically configuration of many interfaces is the same across multiple nodes. ACI ポリシー モデルでは、複数のリーフ ノードにまたがるインターフェイスのグループによって共有されるポリシーグループを作成することでこれを実現できます。The policy-group is identified by a name similar to the port-channel; however, in case of port-channel the policies shared with the group of ports create one logical interface in each leaf while in case of a policy-group, each of the ports sharing the policies are individual physical interfaces. The policy-group concept is very similar to a port-profile in NX-OS.

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	template policy-group <i>policy-group-name</i> 例： apic1(config)# template policy-group pg1	新しいポリシー グループを作成するか、既存のポリシー グループを編集します。
ステップ 3	[no] switchport access vlan <i>vlan-id</i> tenant <i>tenant-name</i> application <i>application-name</i> epg <i>epg-name</i> 例： apic1(config-pol-grp-if)# switchport	

	コマンドまたはアクション	目的
	access vlan 4 tenant ExampleCorp application Web epg webEpg	
ステップ 4	(コンフィギュレーション コマンドの適用) 例 : apicl (config-pol-grp-if) # speed 10G apicl (config-pol-grp-if) # cdp enable	これらの手順の最後の表に、インターフェイスのポリシーグループの設定のためのさまざまなコマンドを示します。
ステップ 5	exit 例 : apicl (config-pol-grp-if) # exit	設定モードに戻ります。
ステップ 6	leaf node-id 例 : apicl (config) # leaf 101-103	設定するリーフを指定します (複数可)。 <i>node-id</i> には、設定の適用対象となる単一のノード ID、または ID の範囲を <i>node-id1-node-id2</i> という形式で指定できます。
ステップ 7	interface type 例 : apicl (config-leaf) # interface ethernet 1/1-24	ポリシー グループを適用するインターフェイスまたはインターフェイスの範囲を指定します。
ステップ 8	[no] policy-group policy-group-name [force] 例 : apicl (config-leaf-if) # policy-group pg1	インターフェイスまたはインターフェイスの範囲にポリシーグループを適用します。Use the keyword no to remove the policy-group from the interface. Use the keyword force to delete any override configurations on the interfaces. 指定したポリシーグループがこのコマンドの前に設定されていない場合、このコマンドは暗黙的にポリシーグループを作成しません。ただし、グローバルスコープでポリシーグループが設定されると、ポリシーグループはインターフェイスに適用されます。 To change the policy-group assignment on an interface, you can enter the policy-group command without first removing the previous policy-group from the interface.

	コマンドまたはアクション	目的
		(注) インターフェイスにポリシーグループを適用してからポートチャンネルにインターフェイスを割り当てる場合、インターフェイスはポリシーグループ設定を失い、ポートチャンネルのポリシーが適用されます。

次の表に、インターフェイスのポリシーグループの設定のためのさまざまなコマンドを示します。

CLI 構文	機能
[no] speed <speedValue>	物理インターフェイスの速度を設定します
[no] link debounce time <time>	物理インターフェイスのリンク でバウンスを設定します
[no] negotiate auto	物理インターフェイスのネゴシエートを設定します
[no] cdp enable	物理インターフェイスの CDP を無効または有効にします
[no] mcp enable	物理インターフェイスの MCP を無効または有効にします
[no] lldp transmit	物理インターフェイスの LLDP 送信を設定します
[no] lldp receive	物理インターフェイスの LLDP 受信を設定します
spanning-tree <bpduguard bpdupfilter> <enable disable>	スパニング ツリー BPDU を設定します
[no] storm-control level <percentage> [burst-rate <percentage>]	ストーム制御（パーセント）を設定します
[no] storm-control pps <packet-per-second> burst-rate <packets-per-second>	ストーム制御（秒当たりのパケット）を設定します

例

次に、ポリシーグループを設定し、リーフ ノード 101 ～ 103 のそれぞれのポートの範囲に適用する例を示します。各リーフでポリシーグループを共有するポートのそれぞれがポリシーグループ `pg1` で定義されているのと同じ設定を持ちます。

```
apic1# configure
apic1(config)# template policy-group pg1
apic1(config-pol-grp-if)# switchport access vlan 4 tenant ExampleCorp application Web
epg webEpg
apic1(config-pol-grp-if)# speed 10G
apic1(config-pol-grp-if)# cdp enable
apic1(config-pol-grp-if)# exit
apic1(config)# leaf 101-103
apic1(config-leaf)# interface ethernet 1/1-24
apic1(config-leaf-if)# policy-group pg1
```

インターフェイスのオーバーライドの設定

ポリシーグループが多数のインターフェイスで使用される場合、割り当てられたポリシーグループ内で設定をオーバーライドできる特定のプロパティに対して、一連のポートを設定するオプションが役立つことがあります。オーバーライド設定は、ポートがポリシーグループに割り当てられている場合にのみ使用できます。オーバーライド設定は、ポートチャネルのメンバーポートでは使用できません。ポートがポートチャネルに追加されると、オーバーライド設定は自動的に削除されます。ただし、オーバーライドが設定されているポートにポリシーグループが割り当てられる限り、オーバーライド設定は自動的に削除されません。ユーザは、必要に応じて `policy-group` コマンドの `force` オプションを使用してオーバーライド設定の削除を決定することができます。

グループポリシーの割り当てがポートから削除されるとき、オーバーライド設定（存在する場合）は変更されません。同様に、ポートが（`force` オプションを使用せずに）別のポリシーグループに割り当てられた場合も、オーバーライド設定は変更されません。オーバーライド設定は設定されると有効になり、ユーザがオーバーライドのきのすべてのプロパティにデフォルト値を割り当てた場合でも削除されません。オーバーライド設定を削除する場合、ユーザは `force` オプションを使用することで、グループポリシーの割り当てを再適用することができます。ただし、`force` オプションは `show running-config` では表示されません。これは ACI モデルでオーバーライド設定を削除するためだけに使用されます。

ACI モデルでは、1 つ以上のプロパティを含むポリシーに対してオーバーライドを設定できます。ポリシーに複数のプロパティがある場合、ポリシー内の 1 つのプロパティだけをオーバーライドすることはできません。CLI フレームワークでは、対応するポリシーに複数のプロパティがあるプロパティをユーザがオーバーライドしようとした際に、曖昧さを回避するため、ポリシー内の他のすべてのプロパティ（オーバーライドプロパティを除く）をオーバーライド設定に暗黙的にコピーします。このような設定の暗黙のコピーは値（デフォルト値を含む）に関係なく `show running-config` の出力に反映されます。また、コピーはオーバーライドポリシー

の設定時に一度だけ行われます。それ以降は、同じポリシー内でプロパティのポリシーグループを変更しても、オーバーライドが設定されているポートには一切影響しません。

オーバーライドの作成時に、ポート割り当て済みのポリシーグループが設定されていない場合、上記のプロパティの暗黙のコピーはできません。代わりに、オーバーライド設定のプロパティ（対応するポリシーに複数のプロパティがある）にはデフォルト値が割り当てられます。ポリシーグループがその後設定されるとき、これらのプロパティはオーバーライド設定では変更されません。ユーザはポリシーグループ自体を設定してからオーバーライドを作成することが推奨されます。すでない場合は、所望の設定を行う際に、（オーバーライドのプロパティに対してデフォルト以外の値を使用してポリシーグループを設定する際に、それに先立ちこれらのプロパティの設定が暗黙的にデフォルトに設定される場合は）ポリシーグループの設定に加えてオーバーライドも設定する必要が発生しかねません。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id 例： <code>apic1(config)# leaf 102</code>	設定するリーフを指定します（複数可）。 <i>node-id</i> には、設定の適用対象となる単一のノードID、またはIDの範囲を <i>node-id1-node-id2</i> という形式で指定できます。
ステップ 3	interface type 例： <code>apic1(config-leaf)# interface ethernet 1/2</code>	オーバーライド設定を使用するインターフェイスまたはインターフェイスの範囲を指定します。
ステップ 4	policy-group policy-group-name force 例： <code>apic1(config-leaf-if)# policy-group pg1 force</code>	ポリシーグループをインターフェイスまたはインターフェイスの範囲に強制適用し、インターフェイスのオーバーライド設定をすべて削除します。

例

次に、ポリシーグループを適用した後、リーフ ノード 101 でポート eth1/1 の速度設定をオーバーライドする例を示します。ACI モデルでは、速度はプロパティ `autoneg` および `link debounce time` も含むポリシーの一部です。このため、これらのプロパティは `pg1` のオーバーライドが設定されるときに速度ポリシーグループからコピーされます。

```
apic1# configure
apic1(config)# interface policy-group pg1
apic1(config-pol-grp-if)# speed 10G
```

```

apic1(config-pol-grp-if)# cdp enable
apic1(config-pol-grp-if)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/1-2
apic1(config-leaf-if)# policy-group pg1
apic1(config-pol-grp-if)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/1
apic1(config-leaf-if)# speed 1G
apic1(config-leaf-if)# show running-config

leaf 101
  interface ethernet 1/1
    policy-group pg1
    speed 1G
    autoneg on
    link debounce time 100

  interface ethernet 1/2
    policy-group pg1

```

次に、リーフ ノード 101 でポート eth1/1 からオーバーライド設定を削除する例を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/1
apic1(config-leaf-if)# policy-group pg1 force
apic1(config-leaf-if)# show running-config

leaf 101
  interface ethernet 1/1
    policy-group pg1

```

エラー訂正の転送について

エラー訂正の転送 (FEC) は、送信元 (トランスミッタ) がエラー訂正コードを使用して冗長的な方法でデータをエンコードしたり、再送信を必要とせずエラーを認識し訂正する場合、信頼できないまたはノイズなチャンネルでデータ転送のエラー制御を行う方法です。使用できるオプションは、次のとおりです。

- CL74-FC-FEC : 25 Gbps の速度をサポートしています。
- CL91-RS-FEC : 25 ~ 100 Gbps の速度をサポートしています。
- Disable-FEC : FEC が無効になります
- Inherit : ポートのトランシーバタイプに基づいてスイッチがFECを使用します。すべての銅線 (CR4) トランシーバでは 25G で FC FEC が有効になっています。100G トランシーバを有するすべてのインターフェイスで RS FEC を有効にします。

デフォルトは「Inherit」です。



(注) FEC は前面ポートでのみ設定かので、ファブリック ポートでは設定できません。

NX-OS スタイル CLI を使用した FEC の設定

手順

	コマンドまたはアクション	目的
ステップ 1	コンフィギュレーション モードを開始します。 例 : <code>apic1# configure</code>	コンフィギュレーション モードを開始します。
ステップ 2	スイッチ モードを開始します。 例 : <code>apic1(config)# leaf 104</code>	スイッチ モードを開始します。
ステップ 3	インターフェイスとポートを指定します。 例 : <code>apic1(config-leaf)# int eth 1/4</code>	インターフェイスとポートを指定します。
ステップ 4	FEC を設定します。 例 : <code>apic1(config-leaf-if)# forward-error-correction c191-rs-fec</code>	RS-FEC を設定します。 (注) デフォルトの <code>forward-error-correction</code> 値は継承されます。
ステップ 5	インターフェイスモードを終了します。 例 : <code>apic1(config-leaf-if)# exit</code>	インターフェイスモードを終了します。



第 3 章

Cisco ACI スマート ライセンス

この章の内容は、次のとおりです。

- [スマート ライセンスについて](#) (39 ページ)

スマート ライセンスについて

Cisco アプリケーション プログラミング インフラストラクチャ コントローラ (APIC) リリース 3.2(1) より、スマート ライセンスが Cisco ACI ファブリックで有効になっており、Cisco スマート ライセンス対応製品として Cisco APIC の拡張機能により有効になっています。Cisco スマート ライセンスは統合ライセンス管理システムであり、Cisco 製品のソフトウェア ライセンスすべてを管理します。

スマート ライセンスには、従来のライセンスで次の利点があります。

- スマート ライセンスの目的に従い、**APIC は ACI コントローラ製品** として場合により参照されます。
- CSSM (Cisco Smart Software Manager) では、中央ポータル ビューを顧客に提供しています。顧客は購入したライセンスすべてと、ライセンスの使用状況およびステータスを表示できます。これにより、ライセンス違反、サブスクリプションベースのライセンスの有効期限切れ、外部コンプライアンスのライセンス違反を防止できます。
- スマート ライセンスをサポートするには、標準の CLI コマンドおよび GUI ビューは異なる Cisco 製品で実装されます。これにより、一貫したユーザーエクスペリエンスを顧客に提供します。
- スマート ライセンスは、ライセンスの管理の複雑さを軽減し、顧客がライセンス関連の問題をトラブルシューティングしやすくします。

次の URL では、スマート ライセンスに関する追加情報を提供します。

- CSSM アカウントにアクセスする顧客ログイン URL : <https://software.cisco.com/>
- Cisco スマート アカウント URL : <https://www.cisco.com/c/en/us/products/software/smart-accounts.html>

次の URL は参照可能なその他のリソースです。

- [トレーニング資料とリソース](#)
- [スマート アカウントおよびスマート ライセンス](#)

スマート ライセンスの使用方法的ガイドラインと制限事項

Follow these Smart Licensing guidelines and limitations:

- 評価期間のカウント ダウン時間は、**apic** 内で保存されます。カウント ダウン時間は、ソフトウェアのダウングレード中に残ります。そのため、お客様は、その APIC ソフトウェアのバージョン 3.2 以降、ダウングレード後に再度をアップグレード、カウントダウン時間は、ダウングレードする前に以前の値から続行します。カウントダウン時間をリセットすることはできません。90 日後にアクションは実行されず、登録するかどうかライセンスのステータスが表示されます評価の期限切れです。
- APIC で有効にする機能のライセンス違反と機能の機能は無効になっていないはありません影響システム機能で。システムは、動作するように続行されますが、ユーザに警告を該当する障害が発生します。最も重大な障害が発生するは **メジャー**。
- 登録に失敗すると、クリックして、**障害** APIC GUI でタブ **システム** > **Smart Licensing** エリア。特定の障害の詳細を表示するには、表示された障害をダブルクリックします。

事前登録の検証

CSSM 設定の検証チェックリスト

以下は、CSSM 内に必要な準備と設定のためのユーザー チェックリストです。

1. 適切なスマート アカウントと仮想アカウントが作成されていることを確認します。
2. Cisco Commerce からスマートが有効なライセンスを購入した場合は、ユーザー購入したライセンスが入力されていることを確認します。
3. APIC スマート ライセンスの登録を開始したら、Cisco TAC エンジニアと協力して適切な CSSM 項目の準備行われていることを確認します。

スマート ライセンスおよび APIC 設定の検証チェックリスト

以下は、APIC 内に必要な準備と設定のためのユーザー チェックリストです。

: DNS 設定は <https://software.cisco.com/> を解決するため APIC で設定される必要があります。

CLI を使用して Smart Licensing に対して登録

CLI を使用して CSSM に直接接続するスマート ライセンスの登録

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーション モードに入ります。
ステップ 2	license smart transport-mode smart-licensing 例 : <code>apic1(config)# license smart transport-mode smart-licensing</code>	スマート ライセンス モードを設定します。
ステップ 3	license smart register idtoken <i>cssm</i> アカウントから <i>id</i> トークン 例 : <code>apic1(config)# license smart register idtoken <id token from cssm account></code>	アカウントからトークンを使用して、CSSM アカウントに登録します。

CLI を使用してトランスポートゲートウェイのスマート ライセンスの登録

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーション モードに入ります。
ステップ 2	license smart transport-mode satellite url <i>http(s)://10.0.0.0:8080/Transportgateway/サービス/DeviceRequestHandler</i> 例 : <code>apic1(config)# license smart transport-mode satellite url http(s)://<ip address hostname of transport gateway>:<http(s) port>/Transportgateway/services/DeviceRequestHandler</code>	トランスポートゲートウェイ モードと URL を設定します。
ステップ 3	license smart register idtoken <i>cssm</i> アカウントから <i>id</i> トークン 例 :	CSSM スマートアカウントまたは CSSM 仮想アカウントからトークンを使用して CSSM に登録します。

Smart Software Manager (SSM) を CLI を使用してサテライトのスマートライセンスの登録

	コマンドまたはアクション	目的
	<code>apic1(config)# license smart register idtoken <id token from cssm account></code>	

Smart Software Manager (SSM) を CLI を使用してサテライトのスマートライセンスの登録

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーション モードに入ります。
ステップ 2	license smart transport-mode satellite url <code>http(s)://10.0.10.1:8080/Transportgateway/サービス/DeviceRequestHandler</code> 例 : <code>apic1(config)# license smart transport-mode satellite url http(s)://<ip address hostname of transport gateway>:<http(s) port>/Transportgateway/services/DeviceRequestHandler</code>	Smart Software Manager (SSM) サテライト モードと URL を設定します。
ステップ 3	license smart register idtoken smart software manager サテライトから id トークン 例 : <code>apic1(config)# license smart register idtoken <id token from smart software manager satellite></code>	Smart Software Manager (SSM) サテライト アカウントからトークンを使用してサテライトに登録します。 (注) 注: は、CSSM アカウントからトークンを使用しないでください。

CLI を使用した HTTP または HTTPS プロキシによるスマートライセンスの登録

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーション モードに入ります。
ステップ 2	license smart transport-mode proxy ip-address ip address port port number 例 : <code>apic1(config)# license smart transport-mode proxy ip-address 10.0.0.248 port 4440</code>	プロキシ モード、IP アドレス、ホスト名および http (s) ポートを設定します。

	コマンドまたはアクション	目的
ステップ 3	license smart register idtoken <i>id token from cssm account</i> 例 : apic1(config)# license smart register idtoken <id token from cssm account>	CSSM スマートアカウントまたはCSSM 仮想アカウントからトークンを使用して、CSSM アカウントを登録します。



第 4 章

APIC 高可用性の設定

- [に関する Cold Standby APIC クラスタの](#) (45 ページ)
- [CLIを使用してスタンバイ apic 内でアクティブな APIC 経由でスイッチング](#) (46 ページ)

に関する Cold Standby APIC クラスタの

Cold Standby APIC クラスタの機能を使用すると、アクティブ/スタンバイ モードでクラスタ apic のパフォーマンスを動作します。Apic 内で、指定したアクティブな apic のパフォーマンス共有負荷をクラスタ化し、指定のスタンバイ apic のパフォーマンスがアクティブ クラスタ apic のパフォーマンスのいずれかの交換用として動作できます。

管理者ユーザとしてを設定することができます、Cold Standby APIC が初めて起動したときに機能します。クラスタ内の少なくとも 3 つ apic のアクティブなパフォーマンスとスタンバイ apic 1 つ以上のパフォーマンスがあることをお勧めします。スタンバイ apic 内でアクティブな APIC を置き換えるには、管理者ユーザとして経由でスイッチを開始できます。

特記事項

- スタンバイ APIC は、アクティブなクラスタと同じファームウェアバージョンでバックアップ APIC を保持するファームウェアのアップデートを自動的に更新されます。
- アップグレードプロセス中に、apic のすべてのアクティブなパフォーマンスをアップグレードすると、スタンバイ APIC もありますが自動的にアップグレードします。
- スタンバイ apic のパフォーマンスには、一時的な Id が割り当てられます。スタンバイ APIC は、アクティブ APIC を経由でスイッチングは、新しい ID が割り当てられます。
- スタンバイ APIC では、Admin login が有効になっていません。トラブルシューティング Cold Standby、として SSH を使用して、スタンバイにログインする必要があります 復旧ユーザ。
- スイッチ オーバー交換のアクティブな APIC 中に電源がオフになって、交換された APIC への接続を防止します。
- 次の条件が失敗する経由でスイッチします。
 - スタンバイ APIC への接続がない場合。

- スタンバイ APIC のファームウェアバージョンがしない場合、アクティブなクラスタのと同じです。
- アクティブにスタンバイ APIC 経由でスイッチング、後にのみのスタンバイ状態である場合、新しいスタンバイを設定する必要があります。
- 上の失敗後にスタンバイ APIC のアドレス アウト オブ バンド維持するため次の制限が適用されます。
 - スタンバイ (新しいアクティブ) APIC が 1 つ以上のアクティブ apic のパフォーマンスがダウンまたは使用できない場合は、アウト バンドアドレスの残されません。
 - スタンバイ (新しいアクティブ) APIC 可能性がありますアクティブ APIC とは異なるサブネットにある場合、そのアウト バンドアドレスの残されません。この制限は APIC 2.x のリリースに対してのみ適用されます。
 - スタンバイ (新しいアクティブ) APIC では、IPv6 アドレス アウト オブ バンドが保持されない可能性があります。この制限は適用されません APIC から開始リリース 3.1 x。
 - スタンバイ (新しいアクティブ) APIC 可能性があります (古い active)] を交換するにはポリシーを以外のスタティック OOB 管理 IP アドレスが設定されている場合、そのアウト バンドアドレスの残されません APIC。



(注) アドレス アウト オブ バンド スタンバイ apic のパフォーマンスを保持するには、制限のいずれかを確認する場合に置換操作が正常に完了した後に、交換された apic 内の OOB ポリシーを手動で変更する必要があります。

- これを置き換えることも、アクティブな apic のパフォーマンスと同じポッドでスタンバイ apic のパフォーマンスを維持することをお勧めします。
- スタンバイ APIC を追加するには 3 つのアクティブな apic のパフォーマンスが必要があります。
- スタンバイ APIC ポリシー設定または管理で参加しません。
- Admin 資格情報を含む、スタンバイ コントローラに複製されるので情報はありません。

CLI を使用してスタンバイ apic 内でアクティブな APIC 経由でスイッチング

スタンバイ apic 内でアクティブな APIC 経由でスイッチするには、次の手順を使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	replace-controller replace <i>ID number</i> <i>Backup serial number</i> 例 : apic1#replace-controller replace 2 FCH1804V27L Do you want to replace APIC 2 with a backup? (Y/n): Y	スタンバイ APIC でアクティブな APIC に置き換えられます。
ステップ 2	replace-controller reset <i>ID number</i> 例 : apic1# replace-controller reset 2 Do you want to reset failover status of APIC 2? (Y/n): Y	アクティブなコントローラのスレーブ をリセットが失敗します。

CLI を使用してスタンバイ apic 内でアクティブな APIC 経由でスイッチング



第 5 章

テナントの設定

- テナント、VRF、およびブリッジ ドメインの作成 (49 ページ)
- ブリッジ ドメイン設定の追加 (52 ページ)
- 適用されるブリッジ ドメインの設定 (54 ページ)
- アプリケーションエンドポイント グループの作成 (56 ページ)
- ブリッジ ドメインのレガシー転送モードの設定 (59 ページ)
- コントラクトの設定 (61 ページ)
- コントラクトの継承 (64 ページ)
- コントラクト優先グループの設定 (74 ページ)
- 他のテナントへのコントラクトのエクスポート (77 ページ)
- コントラクトまたは件名除外の設定 (79 ページ)
- クォータ管理の作成 (82 ページ)

テナント、VRF、およびブリッジ ドメインの作成

このトピックでは、新しいテナントの基本的なプロビジョニングにおける次の手順について説明します。

1. テナントを作成します
2. テナントをセキュリティ ドメインと関連付けます
3. テナントの VRF を作成します
4. テナント内のエンドポイント グループにブリッジ ドメインを作成します

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーションモードに入ります。

	コマンドまたはアクション	目的
ステップ 2	tenant <i>tenant-name</i> 例 : <code>apic1(config)# tenant exampleCorp</code>	テナント (存在しない場合) を作成し、テナントコンフィギュレーションモードを開始します。
ステップ 3	security domain <i>domain-name</i> 例 : <code>apic1(config-tenant)# security domain exampleCorp_dom1</code>	1 つ以上のセキュリティ ドメインとテナントを関連付けます。
ステップ 4	[no] vrf context <i>vrf-name</i> 例 : <code>apic1(config-tenant)# vrf context exampleCorp_v1</code>	テナントのプライベートネットワーク (VRF) を作成します。テナントには 1 つ以上の VRF を設定できます。
ステップ 5	[no] contract { provider consumer } <i>contract-name</i> 例 : <code>apic1(config-tenant-vrf)# contract provider web</code>	VRF 下ですべての EPG のコントラクトを提供または使用します。
ステップ 6	exit 例 : <code>apic1(config-tenant-vrf)# exit</code>	テナントコンフィギュレーションモードに戻ります。
ステップ 7	[no] bridge-domain <i>bd-name</i> 例 : <code>apic1(config-tenant)# bridge-domain exampleCorp_b1</code>	テナント下でブリッジドメインを作成または削除します。ブリッジドメインコンフィギュレーションモードを開始します。
ステップ 8	[no] vrf member <i>vrf-name</i> 例 : <code>apic1(config-tenant-bd)# vrf member exampleCorp_v1</code>	VRF にブリッジドメインを割り当てます。
ステップ 9	exit 例 : <code>apic1(config-tenant-bd)# exit</code>	テナントコンフィギュレーションモードに戻ります。
ステップ 10	interface bridge-domain <i>bd-name</i> 例 : <code>apic1(config-tenant)# interface bridge-domain exampleCorp_b1</code>	テナント インターフェイス コンフィギュレーションモードを開始して、ブリッジドメインにインターフェイスを適用します。
ステップ 11	[no]{ip ipv6} address アドレス/マスク 長[scope { private public }] [secondary]	ブリッジドメインのゲートウェイ IP アドレスを割り当てまたは削除し、IP

	コマンドまたはアクション	目的
	例 : <pre>apicl(config-tenant-if)# ip address 172.1.1.1/24 apicl(config-tenant-if)# ipv6 address 2001:1:1::1/64</pre>	アドレスモードを開始し、オプションの IP アドレス プロパティを設定します。 ゲートウェイアドレスの範囲は、次のいずれかです。 • Public : ルーティング プロトコル (BGP、OSPF、EIGRP) により外部レイヤ 3 ネットワークにアドバタイズできます。 • Private : 外部レイヤ 3 ネットワークにアドバタイズされません。 The optional secondary keyword allows you to configure a secondary gateway address.

例

次に、セキュリティ ドメインへの割り当て、コントラクトと VRF の作成、ブリッジドメインの作成を含むテナントの基本設定の例を示します。

```
apicl# configure
apicl(config)# tenant exampleCorp
apicl(config-tenant)# security domain exampleCorp_dom1
apicl(config-tenant)# vrf context exampleCorp_v1
apicl(config-tenant-vrf)# contract enforce
apicl(config-tenant-vrf)# contract provider web
apicl(config-tenant-vrf)# contract consumer db
apicl(config-tenant-vrf)# contract provider icmp
apicl(config-tenant-vrf)# contract consumer icmp
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# bridge-domain exampleCorp_b1
apicl(config-tenant-bd)# vrf member exampleCorp_v1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# interface bridge-domain exampleCorp_b1
apicl(config-tenant-interface)# ip address 172.1.1.1/24
apicl(config-tenant-interface)# ipv6 address 2001:1:1::1/64
apicl(config-tenant-interface)# exit
```

次に、リーフに特有の VRF 設定の例を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# vrf context exampleCorp_v1 tenant exampleCorp
apicl(config-leaf-vrf)# ip route 1.2.3.4 5.6.7.8
```

次に、リーフ インターフェイスに特有の VRF 設定の例を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# int eth 1/1
apic1(config-leaf-if)# vrf member exampleCorp_v1 tenant exampleCorp

```

次のタスク

アプリケーションプロファイルを追加し、アプリケーションエンドポイントグループ (EPG) を作成し、ブリッジ ドメインに EPG を関連付けます。

ブリッジ ドメイン設定の追加

この項では、ブリッジ ドメインの次の設定について説明します。

- MAC アドレスの設定
- DHCP リレー アドレスの設定
- 共有サービスのルート漏出の設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic# configure	コンフィギュレーション モードに入ります。
ステップ 2	tenant tenant-name 例 : apic(config-tenant)# tenant exampleCorp	テナント コンフィギュレーション モードを開始します。
ステップ 3	interface bridge-domain bd-name 例 : apic(config-tenant)# interface bridge-domain exampleCorp_bd1	テナント インターフェイス コンフィギュレーションモードを開始してブリッジ ドメインを設定します。
ステップ 4	(任意) mac-address mac-address 例 : apic(config-tenant-interface)# mac-address 1234.5678.abcd	広く使用されているゲートウェイ機能の ARP 応答で使用する MAC アドレスを設定します。
ステップ 5	(任意) no mac-address 例 : apic(config-tenant-interface)# no mac-address	MAC アドレスをデフォルトに変更します。

	コマンドまたはアクション	目的
ステップ 6	(任意) [no] ip dhcp relay address tenant <i>tenant-name dhcp-address {application</i> <i>app-name epg epg-name external-l2</i> <i>l2-epg-name external-l3 l3-epg-name}</i> 例 : <pre>apic(config-tenant-interface)# ip dhcp relay address 192.0.20.1 tenant exampleCorp application appl epg epg1</pre>	サポートされているすべてのオプションとともに、ブリッジドメインの DHCP リレー アドレスを設定または削除します。
ステップ 7	(任意) [no]{ip ipv6} shared address アドレス/マスク長 provider application アプリケーション名 epg epg 例 : <pre>apic(config-tenant-interface)# ip shared address 7.8.9.1/24 provider application app2 epg epg2</pre>	ルート漏出は、DHCP、複数テナント VRF の DNS のような共通サービスを提供するために VRF 全体に許可されます。共有サービスは、サブネットをプロバイダーまたはコンシューマのサブネットとしてマークし、共有サービスを提供する EPG を指定することによって実現されます。
ステップ 8	(任意) [no]{ip ipv6} shared epg consumer application アドレス/マスク any 長 addressany 例 : <pre>apic(config-tenant-interface)# ip shared address 3.2.3.4/24 consumer application any epg any</pre>	上のステップを参照してください。

例

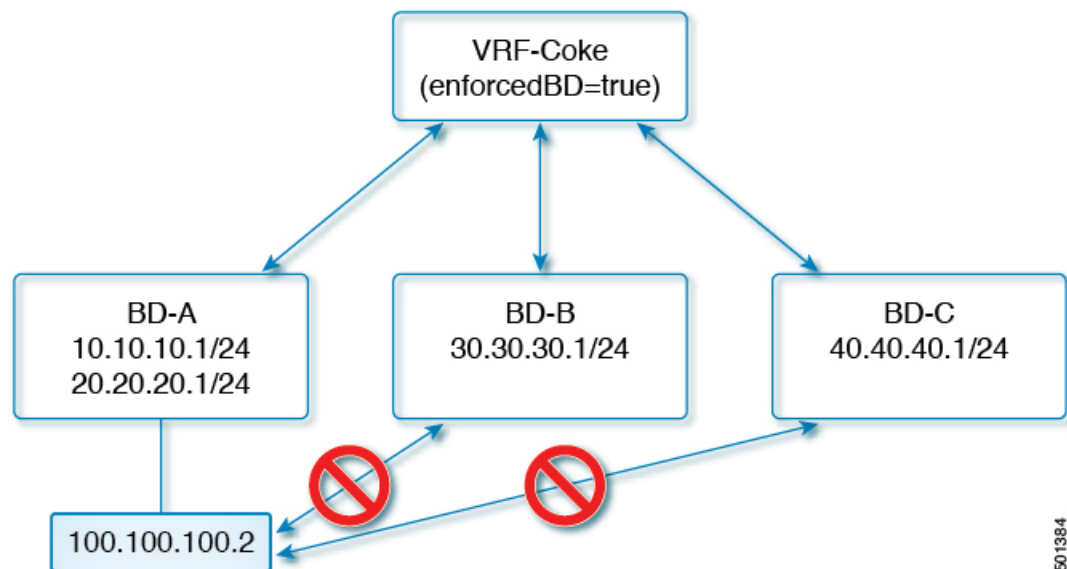
```
apic1# configure
apic1(config)# tenant exampleCorp
apic1(config-tenant)# interface bridge-domain exampleCorp_bd1
apic1(config-tenant-interface)# mac-address 1234.5678.abcd
apic1(config-tenant-interface)# ip dhcp relay address 192.0.20.1 tenant exampleCorp
application appl epg epg1
apic1(config-tenant-interface)# ip shared address 1.2.3.4/24 provider application any
apic1(config-tenant-interface)# ip shared address 3.2.3.4/24 consumer application any
epg any
apic1(config-tenant-interface)# exit
apic1(config-tenant)# exit
apic1(config)# tenant my_dhcp_provider
apic1(config-tenant)# interface bridge-domain bd_dhcp
apic1(config-tenant-interface)# ip shared address 7.8.9.1/24 provider application app2
epg epg2
```

適用されるブリッジドメインの設定

適用されるブリッジドメイン（BD）設定では、関連付けられているブリッジドメイン内のサブネットゲートウェイを ping できるサブジェクトエンドポイントグループ（EPG）で、エンドポイントを作成する必要があります。

この設定では、任意のサブネットゲートウェイに ping を送信できる IP アドレスのグローバル例外リストを作成できます。

図 1: 適用されるブリッジドメイン



- (注)
- 例外の IP アドレスは、すべての VRF 上ですべての BD ゲートウェイの ping を実行できます。
 - L3 アウトに設定されているループバック インターフェイスは、対象とするループバック インターフェイスに設定されている IP アドレスへの到達可能性を強制しません。
 - eBGP ピア IP アドレスが、L3out インターフェイスのサブネットとは異なるサブネットに存在している場合、許容される例外サブネットにピアサブネットを追加する必要があります。

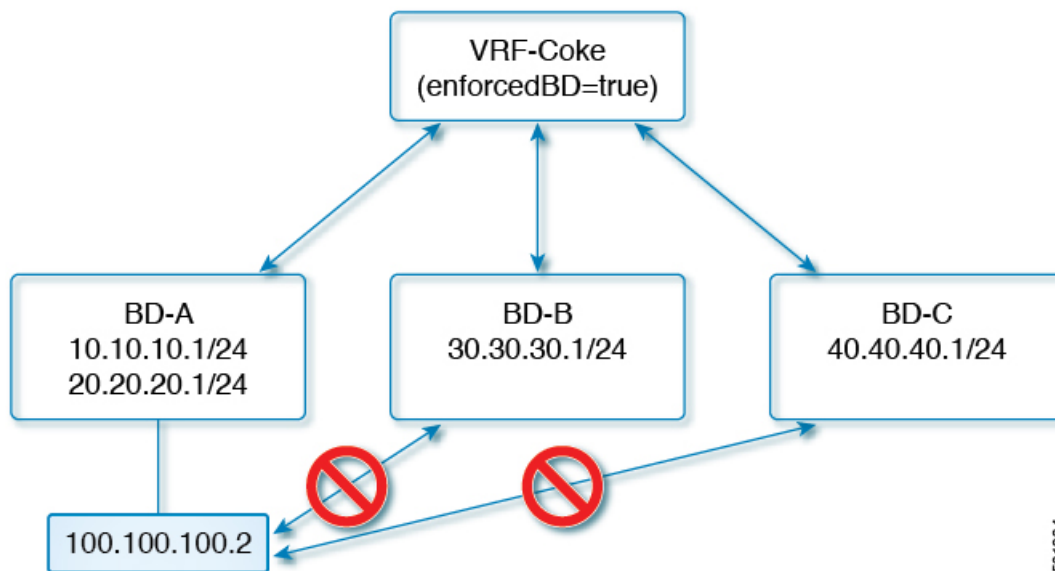
そうしないと、送信元 IP アドレスが L3out インターフェイスのサブネットとは異なるサブネットに存在するため、eBGP トラフィックがブロックされます。

適用されるブリッジドメインの設定

適用されるブリッジドメイン（BD）設定では、関連付けられているブリッジドメイン内のサブネットゲートウェイを ping できるサブジェクトエンドポイントグループ（EPG）で、エンドポイントを作成する必要があります。

この設定では、任意のサブネットゲートウェイに ping を送信できる IP アドレスのグローバル例外リストを作成できます。

図 2: 適用されるブリッジドメイン



- (注)
- 例外の IP アドレスは、すべての VRF 上ですべての BD ゲートウェイの ping を実行できます。
 - L3 アウトに設定されているループバック インターフェイスは、対象とするループバック インターフェイスに設定されている IP アドレスへの到達可能性を強制しません。
 - eBGP ピア IP アドレスが、L3out インターフェイスのサブネットとは異なるサブネットに存在している場合、許容される例外サブネットにピアサブネットを追加する必要があります。
- そうしないと、送信元 IP アドレスが L3out インターフェイスのサブネットとは異なるサブネットに存在するため、eBGP トラフィックがブロックされます。

NX-OS スタイル CLI を使用した適用されるブリッジドメインの設定

このセクションでは、NX-OS スタイル コマンドライン インターフェイス (CLI) を使用して、適用されるブリッジドメインを設定する方法について説明します。

手順

ステップ 1 テナントを作成し有効にします。

例 :

次の例 (「**cokeVrf**」) が作成され有効になっています。

```
apic1(config-tenant)# vrf context cokeVrf
apic1(config-tenant-vrf)# bd-enforce enable
apic1(config-tenant-vrf)# exit
apic1(config-tenant)#exit
```

ステップ 2 例外リストに、サブネットを追加します。

例 :

```
apic1(config)#bd-enf-exp-ip add1.2.3.4/24
apic1(config)#exit
```

適用されるブリッジドメインは次のようなコマンドを使用して動作可能かどうかを確認できます。

```
apic1# show running-config all | grep bd-enf
bd-enforce enable
bd-enf-exp-ip add 1.2.3.4/24
```

例

次のコマンドでは、除外リストからサブネットを削除します。

```
apic1(config)# no bd-enf-exp-ip 1.2.3.4/24
apic1(config)#tenant coke
apic1(config-tenant)#vrf context cokeVrf
```

次のタスク

適用されるブリッジドメインを無効にするには、次のコマンドを実行します。

```
apic1(config-tenant-vrf)# no bd-enforce enable
```

アプリケーション エンドポイント グループの作成

このトピックでは、スタティック アプリケーション EPG の基本的なプロビジョニングにおける次の手順について説明します。

1. テナント内のアプリケーション プロファイルを作成します

2. アプリケーション プロファイルに EPG を作成します
3. EPG にブリッジ ドメインを割り当てます
4. EPG をレイヤ 2 インターフェイスに導入します

始める前に

アプリケーション プロファイルおよびアプリケーション エンドポイント グループ (EPG) を作成する前に、VLAN ドメイン、テナント、VRF、およびブリッジ ドメインを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	コンフィギュレーションモードに入ります。
ステップ 2	tenant tenant-name 例 : apicl(config)# tenant exampleCorp	テナント コンフィギュレーション モードを開始します。
ステップ 3	[no] application app-name 例 : apicl(config-tenant)# application OnlineStore	アプリケーション プロファイルを作成し、アプリケーション プロファイル コンフィギュレーション モードを開始します。
ステップ 4	[no] epg epg-name 例 : apicl(config-tenant-app)# epg exampleCorp_webepg1	アプリケーション プロファイルの EPG を作成 (または削除) し、EPG コンフィギュレーション モードを開始します。
ステップ 5	[no] bridge-domain member epg-name 例 : apicl(config-tenant-app-epg)# bridge-domain member exampleCorp_b1	ブリッジ ドメインに EPG を関連付けます。EPG はすべて BD に属している必要があります。
ステップ 6	exit 例 : apicl(config-tenant-app-epg)# exit	テナント アプリケーション コンフィギュレーション モードに戻ります。
ステップ 7	exit 例 : apicl(config-tenant-app)# exit	テナント コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 8	exit 例 : <code>apic1(config-tenant)# exit</code>	グローバル コンフィギュレーション モードに戻ります。
ステップ 9	leaf node-id 例 : <code>apic1(config)# leaf 101</code>	設定するリーフを指定します。
ステップ 10	interface type 例 : <code>apic1(config-leaf)# interface eth 1/2</code>	設定するインターフェイスを指定します。イーサネット ポートの場合は、「ethernet slot / port」を使用します。
ステップ 11	(任意) switchport 例 : <code>apic1(config-leaf-if)# switchport</code>	レイヤ 2 はポートのデフォルトの状態であるため、このコマンドはポートをレイヤ 3 設定から変換する必要があるときにのみ必要です。
ステップ 12	vlan-domain member domain-name 例 : <code>apic1(config-leaf-if)# vlan-domain member dom1</code>	インターフェイスを VLAN ドメインに関連付けます。
ステップ 13	switchport trunk allowed vlan vlan-id tenant tenant-name app app-name epg epg-name 例 : <code>apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp application OnlineStore epg exampleCorp_webepg1</code>	インターフェイスに EPG を導入し、EPG と VLAN のマッピングを通じて EPG を特定します。この設定は、静的な EPG の導入にのみ適用されます。 VLAN が別の EPG または外部 SVI で使用されている場合、この EPG に使用する前に VLAN 設定を削除する必要があります。 (注) インターフェイスが VLAN ドメインと関連付けられている必要があります、そうでない場合このコマンドは拒否されます。

例

次に、レイヤ 2 ポートに導入されるアプリケーション EPG を作成する例を示します。

```
apic1# configure
apic1(config)# tenant exampleCorp
apic1(config-tenant)# application OnlineStore
```

```
apic1(config-tenant-app)# epg exampleCorp_webepg1
apic1(config-tenant-app-epg)# bridge-domain member exampleCorp_b1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit

apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/2
apic1(config-leaf-if)# switchport
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp application
OnlineStore epg exampleCorp_webepg1
```

次に、ポート チャンネルに EPG を導入する例を示します。

```
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# switchport
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp application
OnlineStore epg exampleCorp_webepg1
```

次のタスク

ポートの VLAN を EPG にマッピングします。

ブリッジ ドメインのレガシー転送モードの設定

レガシーの転送モードでは、コントラクトまたは EPG を使用せずにスイッチングとルーティングを行うことができます。このモードでは、ポートの VLAN はブリッジドメインに直接マッピングされます。The **legacy forwarding vlan** command automatically creates all necessary objects so that no EPG-related configuration is required.

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : configure	コンフィギュレーション モードに入ります。
ステップ 2	tenant tenant-name 例 : apic1(config)# tenant exampleCorp	テナント コンフィギュレーション モードを開始します。
ステップ 3	bridge-domain bd-name 例 : apic1(config-tenant)# bridge-domain exampleCorp_b1	テナント インターフェイス コンフィギュレーションモードを開始してブリッジ ドメインを設定します。

	コマンドまたはアクション	目的
ステップ 4	[no] legacy forwarding vlan <i>vlan-id</i> vlan-domain <i>vlan-domain-name</i> 例 : apicl(config-tenant-bd) # legacy-forwarding vlan 50 vlan-domain dom1	ブリッジドメインに VLAN をマッピングします。
ステップ 5	exit 例 : apicl(config-tenant-bd) # exit	テナント コンフィギュレーション モードに戻ります。
ステップ 6	exit 例 : apicl(config-tenant) # exit	グローバル コンフィギュレーション モードに戻ります。
ステップ 7	leaf <i>node-id</i> 例 : apicl(config) # leaf 101	設定するリーフを指定します。
ステップ 8	interface <i>type</i> 例 : apicl(config-leaf) # interface eth 1/1	設定するインターフェイスを指定します。For an Ethernet port, use ethernet slot/port .
ステップ 9	[no] switchport trunk allowed vlan <i>vlan-id</i> tenant <i>tenant-name</i> legacy-forwarding 例 : apicl(config-leaf-if) # switchport trunk allowed vlan 50 tenant exampleCorp legacy-forwarding	インターフェイスで VLAN を有効にして、レガシーの転送モードで VLAN を使用するテナントブリッジドメインにこれを関連付けます。

例

次に、ブリッジドメイン間の転送にレガシーの転送モードを設定する例を示します。

```

apicl# configure
apicl(config)# tenant exampleCorp
apicl(config-tenant)# bridge-domain exampleCorp_b1
apicl(config-tenant-bd)# legacy-forwarding vlan 50 vlan-domain dom1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# bridge-domain exampleCorp_b2
apicl(config-tenant-bd)# legacy-forwarding vlan 60 vlan-domain dom1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# exit

apicl(config)# leaf 101
apicl(config-leaf)# interface eth 1/1
apicl(config-leaf-if)# vlan-domain member dom1

```

```
apicl(config-leaf-if)# switchport trunk allowed vlan 50 tenant exampleCorp
legacy-forwarding
apicl(config-leaf-if)# exit
apicl(config-leaf)# interface eth 1/2
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# switchport trunk allowed vlan 60 tenant exampleCorp
legacy-forwarding
```

コントラクトの設定

コントラクトは次のタスクでテナントの下に設定します。

- アクセス リストとしてフィルタを定義します
- コントラクトおよびサブジェクトを定義します
- EPG にコントラクトをリンクします

タスクは、この順序に従う必要はありません。たとえば、コントラクトを定義する前に、EPG にコントラクト名をリンクすることができます。



(注) APIC のフィルタ (ACL) では、従来の NX-OS ACL の **permit | deny** の代わりに **match** が使用されます。フィルタ エントリの目的は、特定のトラフィック フローを一致させることです。トラフィックは、ACL にコントラクトまたはタブー コントラクトが適用されると、許可または拒否されます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	コンフィギュレーションモードに入ります。
ステップ 2	tenant tenant-name 例 : tenant exampleCorp	テナント (存在しない場合) を作成し、テナント コンフィギュレーション モードを開始します。
ステップ 3	access-list acl-name 例 : apicl(config-tenant)# access-list http_acl	コントラクトで利用できるアクセス リスト (フィルタ) を作成します。
ステップ 4	(任意) match {arp icmp ip} 例 :	選択したプロトコルのトラフィックに一致するルールを作成します。

	コマンドまたはアクション	目的
	<code>apicl (config-tenant-acl) # match arp</code>	
ステップ 5	(任意) match {tcp udp} [src from[-to]] [dest from[-to]] 例 : <pre>apicl (config-tenant-acl) # match tcp dest 80 apicl (config-tenant-acl) # match tcp dest 443</pre>	TCP または UDP トラフィックに一致するルールを作成します。
ステップ 6	(任意) match raw options 例 : <pre>apicl (config-tenant-acl) #</pre>	Raw vEntry に一致するルールを作成します。
ステップ 7	exit 例 : <pre>apicl (config-tenant-acl) # exit</pre>	テナントコンフィギュレーションモードに戻ります。
ステップ 8	contract contract-name 例 : <pre>apicl (config-tenant) # contract web80</pre>	コントラクトを作成し、コントラクトコンフィギュレーションモードを開始します。
ステップ 9	subject subject-name 例 : <pre>apicl (config-tenant-contract) # subject web80</pre>	コントラクトサブジェクトを作成し、サブジェクトコンフィギュレーションモードを開始します。
ステップ 10	(任意) [no] access-group acl-name [in out both] 例 : <pre>apicl (config-tenant-contract-subj) # access-group http_acl both</pre>	一致するトラフィックの方向を指定し、コントラクトからアクセスリストを追加 (削除) します。
ステップ 11	(任意) [no] label name label-name {provider consumer} 例 : <pre>apicl (config-tenant-contract-subj) #</pre>	サブジェクトにプロバイダーまたはコンシューマのラベルを追加 (削除) します。
ステップ 12	(任意) [no] label match {provider consumer} [any one all none] 例 : <pre>apicl (config-tenant-contract-subj) #</pre>	次のプロバイダーまたはコンシューマのラベルの一致タイプを指定します。 • any : 任意のラベルにコントラクト関係がある場合の一致のこと。 • one : 1つのラベルにコントラクト関係がある場合の一致のこと。

	コマンドまたはアクション	目的
		• all : すべてのラベルにコントラクト関係がある場合の一致のこと。 • none : ラベルにコントラクト関係がない場合の一致のこと。
ステップ 13	exit 例 : <pre>apicl(config-tenant-contract-subj) # exit</pre>	コントラクトコンフィギュレーションモードに戻ります。
ステップ 14	exit 例 : <pre>apicl(config-tenant-contract) # exit</pre>	テナントコンフィギュレーションモードに戻ります。
ステップ 15	application app-name 例 : <pre>apicl(config-tenant) # application OnlineStore</pre>	アプリケーションコンフィギュレーションモードを開始します。
ステップ 16	epg epg-name 例 : <pre>apicl(config-tenant-app) # epg exampleCorp_webepg1</pre>	コントラクトにリンクする EPG のコンフィギュレーションモードを開始します。
ステップ 17	bridge-domain member bd-name 例 : <pre>apicl(config-tenant-app-epg) # bridge-domain member exampleCorp_bdl</pre>	この EPG のブリッジドメインを指定します。
ステップ 18	contract provider provider-contract-name 例 : <pre>apicl(config-tenant-app-epg) # contract provider web80</pre>	この EPG のプロバイダーコントラクトを指定します。この EPG との通信は、このプロバイダーコントラクトに従う通信である限り、その他の EPG から開始することができます。
ステップ 19	contract consumer consumer-contract-name 例 : <pre>apicl(config-tenant-app-epg) # contract consumer rmi99</pre>	この EPG のコンシューマコントラクトを指定します。この EPG のエンドポイントは、このコントラクトを提供する EPG の任意のエンドポイントとの通信を開始することができます。

例

この例では、EPG にコントラクトを作成し適用する方法を示します。

```

apic1# configure
apic1(config)# tenant exampleCorp

# CREATE FILTERS
apic1(config-tenant)# access-list http_acl
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# match tcp dest 443
apic1(config-tenant-acl)# exit

# CREATE CONTRACT WITH FILTERS
apic1(config-tenant)# contract web80
apic1(config-tenant-contract)# subject web80
apic1(config-tenant-contract-subj)# access-group http_acl both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit

# ASSOCIATE CONTRACTS TO EPG
apic1(config-tenant)# application OnlineStore
apic1(config-tenant-app)# epg exampleCorp_webepg1
apic1(config-tenant-app-epg)# bridge-domain member exampleCorp_bd1
apic1(config-tenant-app-epg)# contract consumer rmi99
apic1(config-tenant-app-epg)# contract provider web80
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)#exit
apic1(config-tenant)#exit

# ASSOCIATE PORT AND VLAN TO EPG
apic1(config)#leaf 101
apic1(config-leaf)# interface ethernet 1/4
apic1(config-leaf-if)# switchport trunk allowed vlan 102 tenant exampleCorp application
OnlineStore epg exampleCorp_webepg1

```

この例では、コントラクト自体のフィルタインラインを宣言してコントラクトを定義するためのシンプルな方法を示します。

```

apic1# configure
apic1(config)# tenant exampleCorp
apic1(config-tenant)# contract web80
apic1(config-tenant-contract)# match tcp 80
apic1(config-tenant-contract)# match tcp 443

```

コントラクトの継承

コントラクト継承について

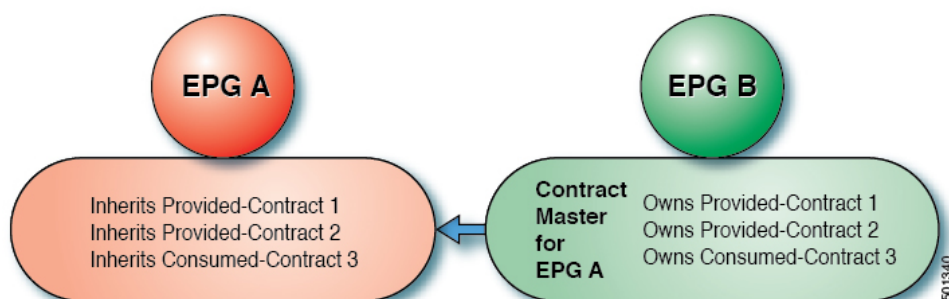
関連する契約を新しい EPG に統合するため、EPG を有効にして同じテナントの別の EPG に直接関連する契約すべて（提供済み/消費済み）を継承できます。コントラクトの継承は、アプ

リケーション EPG、マイクロセグメント EPG、L2Out EPG、および L3Out EPG に設定できます。

リリース 3.x では、EPG 間の提供済み/消費済みの両方の契約に、契約を継承する設定も可能です。EPG 間契約が、モデル名や後発のモデルの最後に EX または EX が付く、Cisco Nexus 9000 シリーズ スイッチでサポートされています。

EPG を有効にし、APIC GUI、NX-OS スタイル CLI、REST API を使用して、別の EPG に直接関連する契約すべてを継承できます。

図 3: コントラクトの継承



上の図で、EPG A は EPG B から（EPG A の契約マスター）提供済みの契約 1 および 2、消費済みの契約 3 を継承するように設定されています。

コントラクト継承を設定する際は、次のガイドラインに従ってください。

- コントラクト継承は、アプリケーション EPG、マイクロセグメント（uSeg）EPG、外部 L2Out EPG、および外部 L3Out EPG 用に設定できます。コントラクト関係は同じタイプの EPG 間で確立する必要があります。
- 関係が確立されると、提供するコントラクトと消費するコントラクトの両方がコントラクトマスターから継承されます。
- コントラクトマスターとコントラクトを継承する EPG は同じテナント内にある必要があります。
- マスター契約への変更は、すべての継承に伝播されます。新しい契約がマスターに追加される場合、継承先にも追加されます。
- EPG は、複数のコントラクトマスターからコントラクトを継承することができます。
- コントラクト継承は単一のレベルでのみサポートされ（連結できない）、コントラクトマスターがコントラクトを継承することはできません。
- コントラクトサブジェクトラベルおよび EPG ラベルの継承がサポートされています。EPG A が EPG B から契約を継承する場合、EPG A と EPG B で異なるサブジェクトラベルが設定されていると、APIC は EPG B で設定されているサブジェクトラベルのみを使用し、どちらの EPG からでもラベルを収集しません。
- EPG が契約に直接関連付けられている、または契約を継承しているかどうかに関わらず、TCAM 内のエントリが消費されます。したがって契約スケールガイドラインが引き続き

適用されます。詳細については、お使いのリリースの「検証されたスケーラビリティガイド」を参照してください。

- vzAny セキュリティ コントラクトとタブー コントラクトはサポートされません。

契約の継承設定および継承済みおよびスタンドアロン契約を表示することに関する詳細は、「Cisco APIC の基本設定ガイドを参照してください。

NX-OS スタイルの CLI を使用したアプリケーションまたは uSeg EPG のコントラクト継承の設定

アプリケーション EPG または uSeg EPG のコントラクト継承を設定するには、次のコマンドを使用します。

始める前に

EPG が使用するテナント、アプリケーション プロファイル、およびブリッジ ドメインを設定します。

VRF レベルで EPG が共有するコントラクトを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーションモードに入ります。
ステップ 2	tenant tenant-name 例 : apic1# (config) tenant Tn1	設定するテナントを作成または指定し、テナントコンフィギュレーションモードを開始します。
ステップ 3	application application-name 例 : apic1(config-tenant)# application AP1	アプリケーションを作成または指定し、アプリケーションモードを開始します。
ステップ 4	epg epg-name [type micro-segmented] 例 : apic1(config-tenant-app)# epg AEPg403	設定するアプリケーション EPG または uSeg EPG を作成または指定し、EPG コンフィギュレーションモードを開始します。uSeg EPG の場合はタイプを追加します。 この例では、アプリケーション EPG のコントラクト マスターです。

	コマンドまたはアクション	目的
ステップ 5	bridge-domain member <i>bd-name</i> 例 : apicl(config-tenant-app-epg)# bridge-domain member T1BD1	ブリッジ ドメインに EPG を関連付けます。
ステップ 6	contract consumer <i>contract-name</i> 例 : apicl(config-tenant-app-epg)# contract consumer cctr5	この EPG が消費するコントラクトを追加します。
ステップ 7	contract provider [<i>label label</i>] 例 : apicl(config-tenant-app-epg)# contract provider T1ctrl_cif	サブジェクトまたは EPG ラベルのオプションリストなど（事前に設定済みである必要があります）、この EPG が提供するコントラクトを追加します。
ステップ 8	exit 例 : apicl(config-tenant-app-epg)# exit	コンフィギュレーションモードを終了します。
ステップ 9	epg <i>epg-name</i> [type micro-segmented] 例 : apicl(config-tenant-app)# epg AEPg404	設定するアプリケーション EPG または uSeg EPG を作成または指定し、EPG コンフィギュレーションモードを開始します。uSeg EPG の場合はタイプを追加します。 この例では、コントラクトを継承する EPG です。
ステップ 10	bridge-domain member <i>bd-name</i> 例 : apicl(config-tenant-app-epg)# bridge-domain member T1BD1	ブリッジ ドメインに EPG を関連付けます。
ステップ 11	inherit-from-epg application <i>application-name</i>epg <i>EPG-contract-master-name</i>] 例 : apicl(config-tenant-app-epg)# inherit-from-epg application AP1 epg AEPg403	この EPG が EPG コントラクトマスターからコントラクトを継承するように設定します。
ステップ 12	exit 例 : apicl(config-tenant-app-epg)# exit	コンフィギュレーションモードを終了します。

	コマンドまたはアクション	目的
ステップ 13	epg epg-name [type micro-segmented] 例 : <pre>apic1(config-tenant-app)# epg uSeg1_403_10 type micro-segmented</pre>	設定するアプリケーション EPG または uSeg EPG を作成または指定し、EPG コンフィギュレーションモードを開始します。 この例では、uSeg EPG のコントラクトマスターです。
ステップ 14	bridge-domain member bd-name 例 : <pre>apic1(config-tenant-app-epg)# bridge-domain member T1BD1</pre>	ブリッジ ドメインに EPG を関連付けます。
ステップ 15	contract provider [label label] 例 : <pre>apic1(config-tenant-app-epg)# contract provider T1ctrl_uSeg_l3out</pre>	サブジェクトまたは EPG ラベルのオプションリストなど（事前に設定済みである必要があります）、この EPG が提供するコントラクトを追加します。
ステップ 16	attribute-logical-expression logical-expression 例 : <pre>apic1(config-tenant-app-epg)# attribute-logical-expression 'ip equals 192.168.103.10 force'</pre>	一致基準として論理式を uSeg EPG に追加します。
ステップ 17	exit 例 : <pre>apic1(config-tenant-app-epg)# exit</pre>	コンフィギュレーションモードを終了します。
ステップ 18	epg epg-name [type micro-segmented] 例 : <pre>apic1(config-tenant-app)# epg uSeg1_403_30 type micro-segmented</pre>	設定するアプリケーション EPG または uSeg EPG を作成または指定し、EPG コンフィギュレーションモードを開始します。 この例では、EPG コントラクト マスターからコントラクトを継承する uSeg EPG です。
ステップ 19	bridge-domain member bd-name 例 : <pre>apic1(config-tenant-app-epg)# bridge-domain member T1BD1</pre>	ブリッジ ドメインに EPG を関連付けます。
ステップ 20	attribute-logical-expression logical-expression 例 :	基準として論理式を uSeg EPG に追加します。

	コマンドまたはアクション	目的
	apicl(config-tenant-app-epg)# attribute-logical-expression 'ip equals 192.168.103.30 force'	
ステップ 21	inherit-from-epg application <i>application-name epg</i> <i>EPG-contract-master-name</i> 例 : apicl(config-tenant-app-epg)# inherit-from-epg application AP1 epg uSeg1_403_10	この EPG が EPG コントラクト マスターからコントラクトを継承するように設定します。
ステップ 22	exit 例 : apicl(config-tenant-app-epg)# exit	コンフィギュレーションモードを終了します。
ステップ 23	exit 例 : apicl(config-tenant-app)# exit	コンフィギュレーションモードを終了します。
ステップ 24	exit 例 : apicl(config-tenant)# exit	コンフィギュレーションモードを終了します。
ステップ 25	exit 例 : apicl(config)# exit	コンフィギュレーションモードを終了します。

例

```

ifav90-ifc1# show running-config tenant Tn1 application AP1
# Command: show running-config tenant Tn1 application AP1
# Time: Fri Apr 28 17:28:32 2017
tenant Tn1
  application AP1
    epg AEPg403
      bridge-domain member T1BD1
      contract consumer cctr5 imported
      contract provider T1ctrl1_cif
    exit
    epg AEPg404
      bridge-domain member T1BD1
      inherit-from-epg application AP1 epg AEPg403
    exit
    epg uSeg1_403_10 type micro-segmented
      bridge-domain member T1BD1
      contract provider T1Ctrl1_uSeg_l3out
      attribute-logical-expression 'ip equals 192.168.103.10 force'
    exit
    epg uSeg1_403_30 type micro-segmented
      bridge-domain member T1BD1

```

```

attribute-logical-expression 'ip equals 192.168.103.30 force'
inherit-from-epg application AP1 epg uSeg1_403_10
exit
exit
exit

```

NX-OS スタイルの CLI を使用した L2Out EPG のコントラクト継承の設定

外部 L2Out EPG のコントラクト継承を設定するには、次のコマンドを使用します。

始める前に

EPG が使用するテナント、VRF、およびブリッジ ドメインを設定します。

EPG が使用するレイヤ 2 外部ネットワーク（L2Out）を設定します。

VRF レベルで EPG が共有するコントラクトを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apic1# configure	コンフィギュレーションモードに入ります。
ステップ 2	tenant tenant-name 例： apic1(config)# tenant Tn1	設定するテナントを作成または指定し、テナントコンフィギュレーションモードを開始します。
ステップ 3	external-l2 epg external-l2-epg-name 例： apic1(config-tenant)# external-l2 epg l2out1:l2Ext1	外部 L2Out EPG を設定または指定します。この例では、L2out コントラクトマスターです。
ステップ 4	bridge-domain member bd-name 例： apic1(config-tenant-l2ext-epg)# bridge-domain member T1BD1	ブリッジドメインに L2Out EPG を関連付けます。
ステップ 5	contract provider contract-name [label label] 例： apic1(config-tenant-l2ext-epg)# contract provider T1ctr_tcp	この EPG が提供するコントラクトを追加します。
ステップ 6	exit 例：	コンフィギュレーションモードを終了します。

	コマンドまたはアクション	目的
	<code>apicl(config-tenant-l2ext-epg) # exit</code>	
ステップ 7	external-l2 epg <i>external-l2-epg-name</i> 例 : <code>apicl(config-tenant) # external-l2 epg L2out12:l2Ext12</code>	外部 L2Out EPG を設定します。この例では、L2out コントラクト マスターからコントラクトを継承する EPG です。
ステップ 8	bridge-domain member <i>bd-name</i> 例 : <code>apicl(config-tenant-l2ext-epg) # bridge-domain member T1BD1</code>	ブリッジ ドメインに L2out EPG を関連付けます。
ステップ 9	inherit-from-epg <i>L2Out-contract-master-name</i> 例 : <code>apicl(config-tenant-l2ext-epg) # inherit-from-epg epg l2out1:l2Ext1</code>	この EPG が L2Out コントラクト マスターからコントラクトを継承するように設定します。
ステップ 10	exit 例 : <code>apicl(config-tenant-l2ext-epg) # exit</code>	コンフィギュレーションモードを終了します。

例

上記の手順は次の例からの抜粋です。

```
apicl# show running-config tenant Tn1 external-l2
# Command: show running-config tenant Tn1 external-l2
# Time: Thu May 11 13:10:14 2017
tenant Tn1
  external-l2 epg l2out1:l2Ext1
    bridge-domain member T1BD1
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out10:l2Ext10
    bridge-domain member T1BD10
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out11:l2Ext11
    bridge-domain member T1BD11
    contract provider T1ctr_udp
    exit
  external-l2 epg l2out12:l2Ext12
    bridge-domain member T1BD12
    inherit-from-epg epg l2out1:l2Ext1
    inherit-from-epg epg l2out10:l2Ext10
    inherit-from-epg epg l2out11:l2Ext11
    inherit-from-epg epg l2out2:l2Ext2
    inherit-from-epg epg l2out3:l2Ext3
    inherit-from-epg epg l2out4:l2Ext4
    inherit-from-epg epg l2out5:l2Ext5
    inherit-from-epg epg l2out6:l2Ext6
    inherit-from-epg epg l2out7:l2Ext7
```

```

inherit-from-epg epg l2out8:l2Ext8
inherit-from-epg epg l2out9:l2Ext9
exit
external-l2 epg l2out2:l2Ext2
bridge-domain member T1BD2
contract provider T1ctr_tcp
exit
external-l2 epg l2out3:l2Ext3
bridge-domain member T1BD3
contract provider T1ctr_tcp
exit
external-l2 epg l2out4:l2Ext4
bridge-domain member T1BD4
contract provider T1ctr_tcp
exit
external-l2 epg l2out5:l2Ext5
bridge-domain member T1BD5
contract provider T1ctr_tcp
exit
external-l2 epg l2out6:l2Ext6
bridge-domain member T1BD6
contract provider T1ctr_tcp
exit
external-l2 epg l2out7:l2Ext7
bridge-domain member T1BD7
contract provider T1ctr_tcp
exit
external-l2 epg l2out8:l2Ext8
bridge-domain member T1BD8
contract provider T1ctr_tcp
exit
external-l2 epg l2out9:l2Ext9
bridge-domain member T1BD9
contract provider T1ctr_tcp
exit
exit

```

NX-OS スタイルの CLI を使用した外部 L3Out EPG のコントラクト継承の設定

外部 L3Out EPG のコントラクト継承を設定するには、次のコマンドを使用します。

始める前に

EPG が使用するテナント、VRF、およびブリッジドメインを設定します。

EPG が使用するレイヤ 3 外部ネットワーク（L3Out）を設定します。

VRF レベルで EPG が共有するコントラクトを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 :	コンフィギュレーションモードに入ります。

	コマンドまたはアクション	目的
	<code>apicl# configure</code>	
ステップ 2	tenant <i>tenant-name</i> 例 : <code>apicl(config)# tenant Tn1</code>	設定するテナントを作成または指定し、テナントコンフィギュレーションモードを開始します。
ステップ 3	external-l3 epg <i>external-l3-epg-name l3out l3out-name</i> 例 : <code>apicl(config-tenant-app)# external-l3 epg l3Ext108 l3out T1L3out1</code>	外部 L3Out EPG を設定します。この例では、L3out コントラクト マスターです。
ステップ 4	vrf member <i>vrf-name</i> 例 : <code>apicl(tenant-l3out)# vrf member T1ctx1</code>	L3out を VRF に関連付けます。
ステップ 5	match ip <i>ip-address-and-mask</i> 例 : <code>apicl(config-tenant-l3ext-epg)# match ip 192.168.110.0/24 shared</code>	EPG の一部としてホストを識別するサブネットを追加し、そのサブネットのオプションの共有範囲を追加します。
ステップ 6	contract provider <i>contract-name [label label]</i> 例 : <code>apicl(config-tenant-l3ext-epg)# contract provider T1ctrl-L3out</code>	この EPG が提供するコントラクトを追加します。
ステップ 7	exit 例 : <code>apicl(config-tenant-l3ext-epg)# exit</code>	コンフィギュレーションモードを終了します。
ステップ 8	external-l3 epg <i>external-l3-epg-name l3out l3out-name</i> 例 : <code>apicl(config-tenant-app)# external-l3 epg l3Ext110 l3out T1L3out1</code>	外部 L3Out EPG を設定します。この例では、L3out コントラクト マスターからコントラクトを継承する EPG です。
ステップ 9	vrf member <i>vrf-name</i> 例 : <code>apicl(tenant-l3out)# vrf member T1ctx1</code>	L3out を VRF に関連付けます。
ステップ 10	match ip <i>ip-address-and-mask</i> 例 : <code>apicl(config-tenant-l3ext-epg)# match ip 192.168.112.0/24 shared</code>	EPG の一部としてホストを識別するサブネットを追加し、そのサブネットのオプションの共有範囲を追加します。

	コマンドまたはアクション	目的
ステップ 11	inherit-from-epg <i>L3Out-contract-master-name</i> 例 : <pre>apic1(config-tenant-l3ext-epg)# inherit-from-epg l3Ext108</pre>	この EPG が L3Out コントラクト マスターからコントラクトを継承するように設定します。
ステップ 12	exit 例 : <pre>apic1(config-tenant-l3ext-epg)# exit</pre>	コンフィギュレーションモードを終了します。

例

```
ifav90-ifc1# show running-config tenant Tn1 external-l3 epg l3Ext110
# Command: show running-config tenant Tn1 external-l3 epg l3Ext110
# Time: Fri Apr 28 17:36:15 2017
tenant Tn1
  external-l3 epg l3Ext108 l3out TlL3out1
    vrf member Tlctx1
    match ip 192.168.110.0/24 shared
    contract provider Tlctrl-L3out
    exit
  external-l3 epg l3Ext110 l3out TlL3out1
    vrf member Tlctx1
    match ip 192.168.112.0/24 shared
    inherit-from-epg epg l3Ext108
    exit
exit
```

コントラクト優先グループの設定

契約優先グループについて

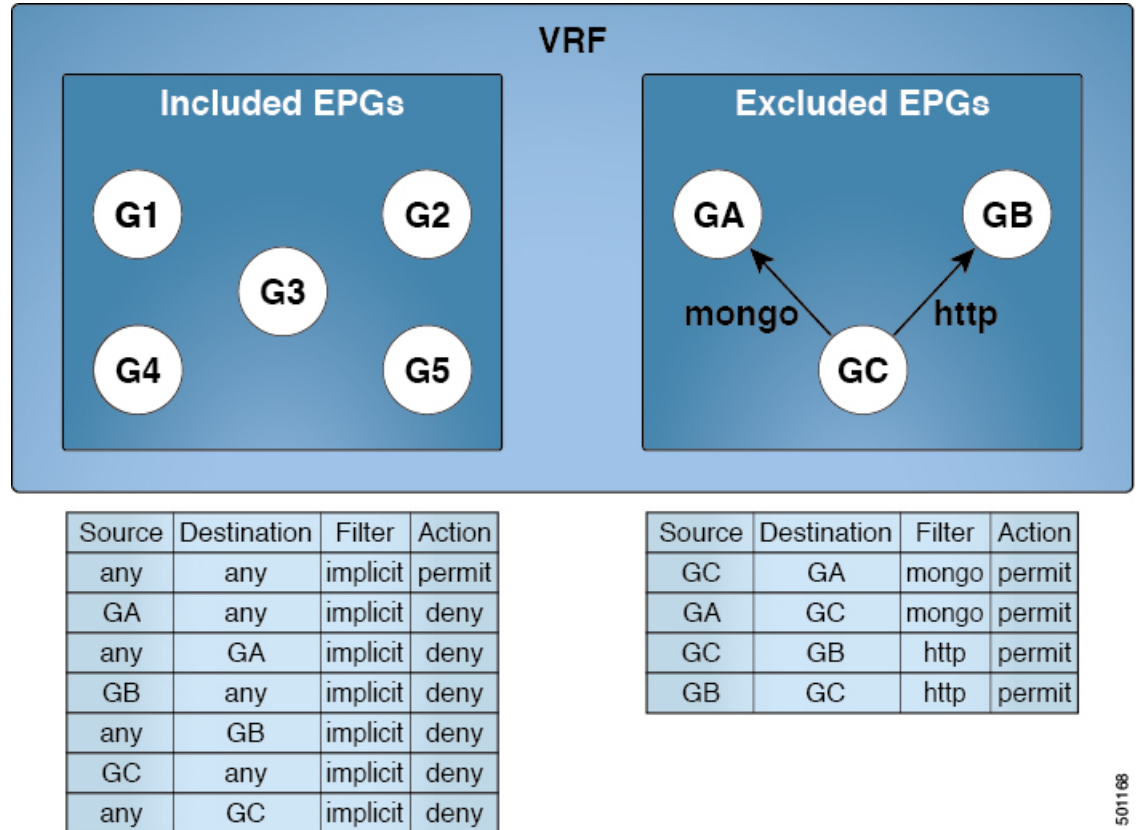
契約優先グループが設定されている VRF で、EPG に利用可能なポリシー適用には 2 種類あります。

- EPG を含む：EPG が契約優先グループのメンバーシップを持っている場合、EPG は契約をせずにお互いに自由に通信できます。これは、source-any-destination-any-permit デフォルトルールに基づくものです。
- EPG を除外：優先グループのメンバーではない EPG は、相互に通信するために契約が必要です。そうしない場合、デフォルトの source-any-destination-any-deny ルールが適用されます。

契約優先グループ機能では、VRF で EPG 間のより高度な通信の制御が可能です。VRF の EPG のほとんどはオープン通信ですが、一部には他の EPG との制限がある場合、契約優先グループとフィルタ付きの契約の組み合わせを設定し、EPG 内の通信を正確に制御できます。

優先グループから除外されている EPG は、source-any-destination-any-deny デフォルトルールを上書きする契約がある場合にのみ、他 EPG と通信できます。

図 4: 契約優先グループの概要



501168

制限事項

以下の制限が契約優先グループに適用されます。

- L3Out およびアプリケーション EPG が契約優先グループで設定されており、EPG が VPC でのみ展開されているトポロジで、VPC の 1 つのリーフ スイッチのみに L3Out のプレフィックス エントリがあることがわかります。この場合、VPC の他のリーフ スイッチにはエントリがなく、そのためトラフィックをドロップします。

この問題を回避するには、次のいずれかを行います。

- VRF の契約グループを無効および再度有効にします。
- L3Out EPG のプレフィックス エントリを削除し再度作成します。
- また、サービス グラフ契約のプロバイダまたはコンシューマ EPG が契約グループに含まれる場合、シャドウ EPG は契約グループから除外できません。シャドウ EPG は契約グループで許可されますが、シャドウ EPG が展開されているノードで契約グループポリシーの展開をトリガしません。ノードに契約グループポリシーをダウンロードするには、契約グループ内にダミー EPG を展開します。

NX-OS スタイル CLI を使用したコントラクト優先グループの設定

APIC NX-OS スタイル CLI を使用して、コントラクト優先グループを設定することができます。この例では、VRF のコントラクト優先グループが設定されています。VRF を使用する EPG のひとつは、優先グループに含まれます。

始める前に

コントラクト優先グループで消費されるテナント、VRF、EPG を作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>apic1# configure apic1(config)#</pre>	設定モードを開始します。
ステップ 2	tenant <i>tenant-name</i> 例 : <pre>apic1(config)# tenant tenant64</pre>	テナントを作成するか、テナント設定モードを開始します
ステップ 3	vrf context <i>vrf-name</i> 例 : <pre>apic1(config-tenant)# vrf context vrf64</pre>	VRF を作成するか、VRF 設定モードを開始します
ステップ 4	whitelist-blacklist-mix 例 : <pre>apic1(config-tenant-vrf)# whitelist-blacklist-mix apic1(config-tenant-vrf)# exit</pre>	VRF のコントラクト優先グループを有効にし、テナント設定モードに戻ります。
ステップ 5	bridge-domain <i>bd-name</i> 例 : <pre>apic1(config-tenant)# bridge-domain bd64</pre>	VRF のブリッジドメインを作成するか、BD 設定モードを開始します。
ステップ 6	vrf member <i>vrf-name</i> 例 : <pre>apic1(config-tenant-bd)# vrf member vrf64 apic1(config-tenant-bd)# exit</pre>	ブリッジドメインと VRF を関連付け、テナント設定モードに戻ります。

	コマンドまたはアクション	目的
ステップ 7	application <i>app-name</i> 例 : apicl(config-tenant)# application app-ldap	アプリケーションを作成するか、アプリケーション設定モードを開始します。
ステップ 8	epg <i>epg-name</i> 例 : apicl(config-tenant-app)# epg epg-ldap	EPG を作成するか、EPG テナントアプリケーション EPG 設定モードを開始します。
ステップ 9	bridge-domain member <i>bd-name</i> 例 : apicl(config-tenant-app-epg)# bridge-domain member bd64	ブリッジ-ドメインに EPG を関連付けます。
ステップ 10	vrf-blacklist-mode 例 : apicl(config-tenant-app-epg)# vrf-blacklist-mode	コントラクト優先グループに含まれるこの EPG を設定します。

例

次の例では、vrf64 のコントラクト優先グループを作成し、epg-ldap を含めます。

```

apicl# configure
apicl(config)# tenant tenant64
apicl(config-tenant)# vrf context vrf64
apicl(config-tenant-vrf)# whitelist-blacklist-mix
apicl(config-tenant-vrf)# exit

apicl(config-tenant)# bridge-domain bd64
apicl(config-tenant-bd)# vrf member vrf64
apicl(config-tenant-bd)# exit

apicl(config-tenant)# application app-ldap
apicl(config-tenant-app)# epg epg-ldap
apicl(config-tenant-app-epg)# bridge-domain member bd64
apicl(config-tenant-app-epg)# vrf-blacklist-mode

```

他のテナントへのコントラクトのエクスポート

1 つのテナントからコントラクトをエクスポートし、別のテナントにインポートできます。コントラクトをインポートするテナントでは、コントラクトはコンシューマコントラクトとしてのみ適用できます。コントラクトはエクスポート時に名前を変更できます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーションモードに入ります。
ステップ 2	tenant <i>tenant-name</i> 例 : <code>apic1(config)# tenant RedCorp</code>	エクスポートするテナントのテナントコンフィギュレーションモードを開始します。
ステップ 3	contract <i>contract-name</i> 例 : <code>apic1(config-tenant)# contract web80</code>	エクスポートするコントラクトのコントラクトコンフィギュレーションモードを開始します。
ステップ 4	scope {<i>application</i> <i>exportable</i> <i>tenant</i> <i>vrf</i>} 例 : <code>apic1(config-tenant-contract)# scope exportable</code>	コントラクトの共有方法を設定します。スコープは次のようになります。 • application—同じアプリケーションの EPG で共有可能 • exportable—テナントで共有可能 • tenant—同じテナントの EPG で共有可能 • vrf—同じ VRF の EPG で共有可能
ステップ 5	export to tenant <i>other-tenant-name</i> as <i>new-contract-name</i> 例 : <code>apic1(config-tenant-contract)# export to tenant BlueCorp as webContract1</code>	他のテナントにコントラクトをエクスポートします。同じコントラクト名を使用することも、名前を変更することもできます。
ステップ 6	exit 例 : <code>apic1(config-tenant-contract)# exit</code>	テナントコンフィギュレーションモードに戻ります。
ステップ 7	exit 例 : <code>apic1(config-tenant)# exit</code>	グローバル コンフィギュレーションモードに戻ります。
ステップ 8	tenant <i>tenant-name</i> 例 : <code>tenant BlueCorp</code>	インポートするテナントのテナントコンフィギュレーションモードを開始します。

	コマンドまたはアクション	目的
ステップ 9	application <i>app-name</i> 例 : apicl(config-tenant) # application BlueStore	アプリケーション コンフィギュレーション モードを開始します。
ステップ 10	epg <i>epg-name</i> 例 : apicl(config-tenant-app) # epg BlueWeb	コントラクトにリンクする EPG のコンフィギュレーションモードを開始します。
ステップ 11	contract consumer <i>consumer-contract-name</i> imported 例 : apicl(config-tenant-app-epg) # contract consumer webContract1 imported	この EPG にインポートされたコンシューマコントラクトを指定します。この EPG のエンドポイントは、このコントラクトを提供する EPG の任意のエンドポイントとの通信を開始することができます。

例

次に、テナント RedCorp から、コンシューマ コントラクトになるテナント BlueCorp にコントラクトをエクスポートする例を示します。

```

apic# configure
apicl(config) # tenant RedCorp
apicl(config-tenant) # contract web80
apicl(config-tenant-contract) # scope exportable
apicl(config-tenant-contract) # export to tenant BlueCorp as webContract1
apicl(config-tenant-contract) # exit
apicl(config-tenant) # exit
apicl(config) # tenant BlueCorp
apicl(config-tenant) # application BlueStore
apicl(config-tenant-application) # epg BlueWeb
apicl(config-tenant-application-epg) # contract consumer webContract1 imported

```

コントラクトまたは件名除外の設定

コントラクトまたはコントラクトの件名の例外の設定

Cisco APIC リリース 3.2(1) では、EPG 間のコントラクトが拡張され、コントラクトに参加しているコントラクトプロバイダまたはコンシューマのサブネットを拒否できます。インター EPG コントラクトおよび内部 EPG コントラクトは、この機能でサポートされます。

プロバイダ EPG の件名を有効にして、件名またはコントラクトの例外で一致基準が設定されているものを除くすべてのコンシューマ EPG との通信が可能になります。たとえば、サブセッ

トを除く、テナントのすべての EPG にサービスを提供するために EPG を有効にする場合、これら EPG を除外できます。これを設定するには、コントラクトまたはそのコントラクトの件名のいずれかで例外を作成します。サブセットがコントラクトの提供または消費のアクセスを拒否します。

ラベル、カウンタ、許可および拒否ログは、コントラクトおよび件名の例外でサポートされています。

コントラクトのすべての件名に例外を適用するには、コントラクトに例外を追加します。コントラクトの単一の件名にのみ例外を適用する場合、件名に例外を追加します。

件名にフィルタを追加する場合、フィルタのアクションを設定できます（フィルタ条件に一致するオブジェクトを許可または拒否する）。また、**[拒否]** フィルタについては、フィルタの優先順位を設定することができます。**[許可]** フィルタは常にデフォルトの優先順位があります。自動拒否の件名-フィルタ関係をマーキングすると、件名に一致している場合、各 EPG のペアに適用されます。コントラクトと件名には、複数の件名-フィルタ関係を含むことができます。これは、フィルタに一致するオブジェクトを許可または拒否するように独自に設定できます。

例外タイプ

コントラクトと件名の例外は次のタイプに基づき、* ワイルドカードなどの正規表現を含むことができます。

例外の条件は、[コンシューマ正規表現] および [プロバイダ正規表現] のフィールドで定義されているように、これらのオブジェクトを除外します。	例	説明
テナント	<pre><vzException consRegex="common" field= "Tenant" name= "excep03" provRegex="t1" /></pre>	この例では、common テナントを使用して、EPG が t1 テナントにより提供されるコントラクトを消費しないように除外します。
VRF	<pre><vzException consRegex="ctx1" field= "Ctx" name= "excep05" provRegex= "ctx1" /></pre>	この例では、ctx1 のメンバーが同じ VRF から提供されるサービスを使用しないように除外します。
EPG	<pre><vzException consRegex="EPgPa*" field= "EPg" name= "excep03" provRegex= "EPg03" /></pre>	この例では、名前が EPgPa から始まる複数の EPG が存在すると仮定し、EPg03 により提供されているコントラクトのコンシューマとしてすべて拒否する必要があります。

例外の条件は、[コンシューマ正規表現] および[プロバイダ正規表現]のフィールドで定義されているように、これらのオブジェクトを除外します。	例	説明
Dn	<pre><vzException consRegex= "uni/tn-t36/ap-customer/epg-epg193" field= "Dn" name="excep04" provRegex= "uni/tn-t36/ap-customer/epg-epg200" /></pre>	この例では、epg193 が epg200 により提供されたコントラクトを消費しないように除外します。
タグ	<pre><vzException consRegex= "red" field= "Tag" name= "excep01" provRegex= "green" /></pre>	例では、red タグでマークされているオブジェクトが消費することと、green タグでマークされているオブジェクトがコントラクトに参加しないように除外します。

NX-OS スタイルの CLI を使用したコントラクトまたはコントラクトの件名除外の設定

このタスクでは、ほとんどの EPG の通信を許可するコントラクトを設定しますが、それらのサブネットへのアクセスを拒否します。契約またはサブジェクトには、複数の例外を追加することができます。

始める前に

テナント、VRF、アプリケーション プロファイル、EPG を設定して、コントラクトを提供し消費します。

手順

ステップ 1 次の例のようにコマンドを使用して、HTTP および HTTPS のフィルタを設定します。

例：

```
apicl(config)# tenant t2
apicl(config-tenant)# access-list acl
apicl(config-tenant-acl)# match ip
apicl(config-tenant-acl)# match tcp dest 80
apicl(config-tenant-acl)# exit
apicl(config-tenant)# access-list ac2
apicl(config-tenant-acl)# match ip
apicl(config-tenant-acl)# match tcp dest 443
```

ステップ 2 EPG01 の消費と EPG03 の提供を除外するコントラクトを設定します。

例：

```
apic1(config-tenant)# contract webCtct
apic1(config-tenant-contract)# subject https-subject
apic1(config-tenant-contract-subj)# exception name EPG consumer-regexp EPg01 field EPg
provider-regexp EPg03
apic1(config-tenant-contract-subj)# access-group ac1 in blacklist
apic1(config-tenant-contract-subj)# access-group ac2 in whitelist
```

クォータ管理の作成

APIC クォータ管理の設定について

Cisco で開始 Application Policy Infrastructure Controller (APIC) 2.3(1) のリリースでは、テナント管理者を設定できるオブジェクトの数に制限します。これにより、特定のテナントでまたはテナント間でグローバルに追加できるどのような管理対象オブジェクトを制限する管理できます。APIC) 2.3(1) のリリースでは、テナント管理者を設定できるオブジェクトの数に制限します。これにより、特定のテナントでまたはテナント間でグローバルに追加できるどのような管理対象オブジェクトを制限する管理できます。

この機能は、任意のテナントまたは承認からテナントのグループを制限するときに便利な ACI リーフ単位またはまたは不当可能性がある他のテナントで同じファブリックに影響する、使用可能なリソースの大部分を使用するファブリックあたりの最大値。

NX-OS スタイル CLI を使用したクォータ設定の作成

この手順では、NX-OS スタイル CLI を使用してクォータ管理構成を作成する方法について説明します。

手順

NX-OS CLI を使用して、クォータ管理設定を作成します。

例：

```
apic1# conf t
apic1(config)# quota fvBD max 100 scope uni/tn-green exceed-action fault
apic1(config)# quota fvBD max 1000 scope uni exceed-action fail
apic1(config)# quota fvBD max 34 tenant red
```

構文：

```
[no] quota <className> max <maxValue> [exceed-action {fail|fault}] \
[scope <containerDn>| tenant <tenantName> \
[{bridge-domain <bd>|application <ap> [epg <epgName>]}]]]
```

<className> は、**fvBD** や **fvCtx** などの管理対象オブジェクトクラス名です。モデルのクォータ フラグの存在に従ってすべての対象となるクラスは受け入れられます。

<maxValue> は、<exceed-action> が適用された後の値です。

<maxValue> が超過した後 <exceed-action> がアクションを起こす場合、次のいずれかになります。

- **失敗**：制限を超えたトランザクションが失敗することを希望する場合。
- **障害**：障害が発生します。

ここで、<containerDn> は、制限が適用されるツリーです。「uni」はポリシーモデル全体ACI にわたって行われ、「tenant green」は、グリーンのテナントに対して行われます。



第 6 章

レイヤ 2 外部接続の設定

- [レイヤ 2 外部接続の設定 \(85 ページ\)](#)
- [VLAN ドメインの設定 \(89 ページ\)](#)
- [EPG の Q-in-Q カプセル化マッピングの設定 \(97 ページ\)](#)
- [ACI ファブリックでイーサネット トラフィックを介したファイバ チャネルのサポート \(100 ページ\)](#)
- [ファイバ チャネル NPV \(115 ページ\)](#)
- [802.1Q トンネルの設定 \(121 ページ\)](#)
- [ダイナミック ブレークアウト ポートの設定 \(127 ページ\)](#)
- [ポート プロファイルの設定 \(132 ページ\)](#)
- [仮想スイッチ上のマイクロセグメンテーション \(140 ページ\)](#)
- [ベア メタル上のマイクロセグメンテーションの設定 \(143 ページ\)](#)
- [レイヤ 2 IGMP スヌープ マルチキャストの設定 \(145 ページ\)](#)
- [ポート セキュリティの設定 \(154 ページ\)](#)
- [プロキシ ARP の設定 \(161 ページ\)](#)
- [MACsec の設定 \(171 ページ\)](#)

レイヤ 2 外部接続の設定

レイヤ 2 外部接続とは、ACI リーフスイッチ（別名境界リーフ）と外部ルータ間のスイッチング ネットワークのことを表します。外部 L2 ネットワークを表す VLAN はファブリック内のブリッジドメインの 1 つにマッピングされ、ブリッジドメインにレイヤ 2 拡張を提供し、ブリッジドメインを使用する EPG を外部ネットワークと通信できるようにします。外部ネットワークは EPG にマッピングされるため、ノード全体でさまざまな内部アプリケーションとさまざまな L2 外部 VLAN 間のコントラクトを認識するのに役立ちます。



注意 APIC でインターフェイスごとの設定を行う際に、GUI と CLI を混在させないでください。GUI で行われた設定が、NX-OS CLI では部分的にしか機能しない可能性があります。

たとえば、GUI の **[Tenants] > [tenant-name] > [Application Profiles] > [application-profile-name] > [Application EPGs] > [EPG-name] > [Static Ports] > [Deploy Static EPG on PC, VPC, or Interface]** でスイッチ ポートを設定したと仮定します。

次に NX-OS スタイルの CLI で `show running-config` コマンドを使用すると、以下のような出力を受信します。

```
leaf 102
interface ethernet 1/15
switchport trunk allowed vlan 201 tenant t1 application ap1 ep1
exit
exit
```

NX-OS スタイルの CLI でこれらのコマンドを使用してスタティック ポートを設定すると、次のエラーが発生します。

```
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 1/15
apic1(config-leaf-if)# switchport trunk allowed vlan 201 tenant t1 application ap1 ep1
No vlan-domain associated to node 102 interface ethernet1/15 encap vlan-201
```

これは、CLI に APIC GUI では実行されない検証があることが原因です。show running-config コマンドによって出力されたコマンドが NX-OS CLI で機能するためには、VLAN ドメインが事前に設定されている必要があります。設定の順序は GUI に適用されません。

レイヤ 2 外部接続の設定はスタティック アプリケーション EPG と似ていて、ノード、ポートの VLAN を EPG にマッピングし、EPG をブリッジドメインにマッピングして、コントラクトを提供または使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	設定モードにアクセスします。 例： <code>apic1# configure</code>	
ステップ 2	テナント設定モードを開始します。 例： <code>apic1(config)# tenant exampleCorp</code>	
ステップ 3	<code>[no] external-l2 ep1 ep1-name</code> 例： <code>apic1(config-tenant)# external-l2 ep1 extendBD1</code>	外部レイヤ 2 EPG を作成（または削除）します。

	コマンドまたはアクション	目的
ステップ 4	EPG にブリッジドメインを割り当てます。 例： <code>apicl(config-tenant-extl2epg) # bridge-domain member bd1</code>	
ステップ 5	テナント設定モードに戻ります。 例： <code>apicl(config-tenant-extl2epg) # exit</code>	
ステップ 6	グローバル コンフィギュレーションモードに戻ります。 例： <code>apicl(config-tenant) # exit</code>	
ステップ 7	設定するリーフを指定します。 例： <code>apicl(config) # leaf 101</code>	
ステップ 8	外部 EPG のポートを指定します。 例： <code>apicl(config-leaf) # interface eth 1/2</code>	
ステップ 9	デフォルトでは、ポートはレイヤ 2 トランク モードです。ポートがレイヤ 3 モードである場合、このコマンドを使用してレイヤ 2 トランク モードに変換する必要があります。 例： <code>apicl(config-leaf-if) # switchport</code>	
ステップ 10	インターフェイスを VLAN ドメインに関連付けます。 例： <code>apicl(config-leaf-if) # vlan-domain member dom1</code>	
ステップ 11	Assigns a VLAN on the leaf port and maps the VLAN to a layer 2 external EPG, with the switchport trunk allowed vlan <i>vlan-id</i> tenant <i>tenant-name</i> external-l2 <i>epg</i> <i>epg-name</i> command. 例：	(注) インターフェイスが VLAN ドメインと関連付けられている必要があります、そうでない場合このコマンドは拒否されます。

	コマンドまたはアクション	目的
	<pre>apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp external-12 epg extendBD1</pre>	
ステップ 12	リーフポートに vlan を割り当て、switchport {trunk allowed を使用して vlan を外部 svi にマッピングします。trunk native vlan tenant external-svi } vlan id テナント名 コマンド。 access 例 : <pre>apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp external-svi</pre>	(注) インターフェイスが VLAN ドメインと関連付けられている必要があり、そうでない場合このコマンドは拒否されます。

例

次に、外部接続用にレイヤ 2 ポートを導入する例を示します。

```
apic1# configure
apic1(config)# tenant exampleCorp
apic1(config-tenant)# external-12 epg extendBD1
apic1(config-tenant-ext12epg)# bridge-domain member bd1
apic1(config-tenant-ext12epg)# exit
apic1(config-tenant)# exit

apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/2
apic1(config-leaf-if)# switchport
apic1(config-leaf-if)# switchport mode trunk
apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp external-12
epg extendBD1
```

次に、外部接続用にレイヤ 2 ポート チャンネルまたは vPC を導入する例を示します。

```
...

apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp external-12
epg extendBD1
```

次に、外部接続用にレイヤ 2 インターフェイスで SVI を設定する例を示します。

```
apic1(config)# leaf 101

apic1(config-leaf)# interface ethernet 1/5
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp external-svi
apic1(config-leaf-if)# no switchport trunk allowed vlan 10 tenant exampleCorp external-svi

apic1(config-leaf)# interface ethernet 1/37
```



```
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport access vlan 11 tenant exampleCorp external-svi
apic1(config-leaf-if)# no switchport access vlan 11 tenant exampleCorp external-svi

apic1(config-leaf)# interface port-channel po34
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk native vlan 12 tenant exampleCorp external-svi
apic1(config-leaf-if)# no switchport trunk native vlan 12 tenant exampleCorp external-svi
```

VLAN ドメインの設定

VLAN ドメインについて

ACI ファブリックは、4K VLAN のグループに分割することにより、複数のテナントで使用できる、ファブリック全体の大規模なレイヤ 2 ドメインを利用可能にします。VLAN ドメインはノードおよびポートのグループ上で設定できる一連の VLAN を表します。VLAN ドメインは、複数のテナントを共有し、ノード、ポートおよび VLAN などの共通のリソースを個別に管理できるようになりました。テナントは 1 つ以上の VLAN ドメインにアクセスできます。VLAN プールの詳細については、「*Cisco Application セントリック インフラストラクチャ基礎*」の「*ACI ポリシー モード I*」章の「エンドポイント グループ」を参照してください。

これらの VLAN ドメインは、スタティックまたはダイナミックに設定できます。スタティック VLAN ドメインは、スタティック VLAN プールをサポートしますが、ダイナミック VLAN ドメインは、スタティックとダイナミックの両方の VLAN プールをサポートできます。スタティック プールの VLAN は、ユーザによって管理され、ベア メタル ホストへの接続などのアプリケーションに使用されます。ダイナミック プールの VLAN は、ユーザの介入なしに APIC によって割り当てかつ管理され、VMM などのアプリケーションに使用されます。VLAN ドメインおよびドメイン内の VLAN プールのデフォルト タイプはスタティックです。

ファブリック管理者は、テナントが L2/L3 設定のファブリック リソースの使用を開始する前に、次のステップを実行します。

1. VLAN ドメインを作成し、各 VLAN ドメイン内の VLAN を割り当てます。
2. 1 つ以上の VLAN ドメインにリーフ スイッチの外部向けポートを割り当てます。
3. Convert a port to L2/L3 by using the **[no] switchport** command. ポートのデフォルト状態はトランク モードでは、L2（スイッチポート）です。
4. L2 ポートの場合、グローバルまたはローカル ポートになるポート上の VLAN の範囲を設定します。デフォルトは **global** です。

ファブリック管理者は、VLAN ドメインがテナントに割り当てられ、テナントアプリケーションによって使用された後でも、これらの手順の設定を更新することができます。

スパニングツリー および VLAN ドメインに関する注意事項

スパニングツリーに ACI ファブリックが参加していない場合でも、アクセス ポリシー設定に基づきスパニングツリー ドメインをパーティション化できます。ACI は、スパニングツリー ドメインを判断するのにブリッジドメインまたはその設定に依存しません。代わりに、VLAN プールが EPG ドメインに割り当てられている場合、リーフ スイッチは同じ VLAN のカプセル化内の BPDU をフラッドします。EPG ドメインに割り当てられている VLAN プールは、最終的にスパニングツリー ドメインとして機能します。

異なる VLAN プールに関連付けられた複数の EPG ドメインを使用する場合、すべて同じ VLAN ID を使用していても、BPDU がエンドポイント プロパティ全体でフラッドすることを許可しません。EPG ドメインのタイプ（物理またはレイヤ 2 の外部）では、この動作は変更されません。

ACI ファブリックはスパニングツリー ドメイン内のすべてのデバイスからのすべての BPDU をフラッドするため、これにより各インターフェイスの MAC アドレスなど、外部デバイスの BPDU 情報を確認する動作をトリガする可能性があります。アクティブになる機能の例は、IOS デバイスの「スパニングツリー EtherChannel の不正な設定ガード」です。これらの機能は、レイヤ 2 トンネルとして ACI を利用する際に考慮する必要があります。



- (注) ポート VLAN ごとの機能 (localPort 範囲に同じ VLAN ID を使用してリーフ スイッチで複数の EPG を設定する) で設定されたインターフェイスでは、マルチ スパニング ツリー (MST) はサポートされません。

基本的な VLAN ドメイン設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] vlan-domain domain-name [dynamic] 例 : <code>apic1(config)# vlan-domain dom2 dynamic</code>	VLAN ドメインを作成するか、既存のドメインを編集します。Include the dynamic keyword to create a dynamic VLAN pool. デフォルトは static です。
ステップ 3	[no] vlan range [dynamic] 例 : <code>apic1(config-vlan)# vlan 1000-1999,4001</code>	VLAN ドメインに VLAN の範囲またはカンマ区切りのリストを割り当てます。 VLAN は、スタティックまたはダイナミックのいずれかになります。スタティック VLAN は、たとえば、ホスト

	コマンドまたはアクション	目的
		から外部のスイッチドネットワークへの接続を提供するなど、ユーザによって設定されますが、ダイナミック レンジの VLAN は、VMM または L4～L7 サービスなどの APIC アプリケーションによって内部的に設定されます。デフォルトタイプはスタティックです。 (注) スタティック ドメインは、ダイナミック VLAN を含めることはできません。 特定のポートの VLAN は、1 つの VLAN ドメインのみにマッピングできます。これは、設定時に適用されます。

例

この例では、基本的な VLAN ドメインを設定する方法を示します。

```

apic1# configure
apic1(config)# vlan-domain dom1
apic1(config-vlan)# vlan 1000-1999,4001
apic1(config-vlan)# exit
apic1(config)# vlan-domain dom2 dynamic
apic1(config-vlan)# vlan 101-200
apic1(config-vlan)# vlan 301-400 dynamic

```

高度な VLAN ドメイン設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] vlan-domain domain-name [dynamic] [type {phys l2ext l3ext}] 例 :	VLAN ドメインを作成するか、既存のドメインを編集します。Include the dynamic keyword to create a dynamic VLAN pool. デフォルトは static です。

	コマンドまたはアクション	目的
	<pre>apic1(config)# vlan-domain dom1 type phys</pre>	The type option is visible and mandatory if one or more of the following conditions exist: • 3つのすべてのVLAN ドメインのタイプが、このドメイン名に存在しない場合 • 3つのVLAN ドメインのタイプが、異なるVLAN プールを持つ場合 • 3つのVLAN ドメインのタイプが、同じVLAN プールを共有するが、プール名はVLAN ドメインの名前と異なる場合
ステップ 3	[no] vlan-pool <i>vlan-pool-name</i> 例 : <pre>apic1(config-leaf)# vlan-pool myVlanPool3</pre>	VLAN プールを作成します。This command is available only when the type option is present in the vlan-domain command. You must declare the VLAN pool before adding VLANs with the vlan command.
ステップ 4	[no] vlan <i>range</i> [dynamic] 例 : <pre>apic1(config-vlan-domain)# vlan 1000-1999,4001</pre>	VLAN ドメインにVLAN の範囲またはカンマ区切りのリストを割り当てます。 VLAN は、スタティックまたはダイナミックのいずれかになります。スタティック VLAN は、たとえば、ホストから外部のスイッチドネットワークへの接続を提供するなど、ユーザによって設定されますが、ダイナミック レンジのVLANは、VMMまたはL4～L7サービスなどのAPICアプリケーションによって内部的に設定されます。デフォルトタイプはスタティックです。 (注) スタティック ドメインは、ダイナミック VLAN を含めることはできません。 特定のポートのVLANは、1つのVLAN ドメインのみにマッピングできます。これは、設定時に適用されます。

	コマンドまたはアクション	目的
ステップ 5	show vlan-domain [name domain-name] [vlan vlan-id] [leaf leaf-id] 例 : <pre>apic1(config-vlan-domain)# show vlan-domain name dom1 vlan 1002 leaf 102</pre>	アプリケーション EPG、サブインターフェイス、外部 SVI および外部 L2 などのアプリケーションの VLAN ドメインの使用状況を表示します。

例

次の例では、VLAN プールを使用する VLAN ドメインを設定する方法を示します。

```
apic1# configure
(config)# vlan-domain dom1 type phys
(config-vlan-domain)# vlan-pool myVlanPool3
(config-vlan-domain)# vlan 1000-1999, 4001
```

ポートへの VLAN ドメインの関連付け

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>apic1# configure</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id1-node-id2 例 : <pre>apic1(config)# leaf 101-102</pre>	設定するリーフのペアを指定します。
ステップ 3	interface type 例 : <pre>apic1(config-leaf)# int eth 1/1-24</pre>	VLAN ドメインに関連付けられたポートまたはポートの範囲を指定します。
ステップ 4	[no] vlan-domain member domain-name 例 : <pre>apic1(config-leaf-if)# vlan-domain member dom1</pre>	VLAN ドメインに指定したポートを割り当てます。
ステップ 5	[no] switchport 例 : <pre>apic1(config-leaf-if)# switchport</pre>	デフォルトでは、ポートはレイヤ 2 トランク モードです。ポートがレイヤ 3 モードである場合、このコマンドを使用し

	コマンドまたはアクション	目的
		て、レイヤ 2 トランク モードに変換する必要があります。
ステップ 6	(任意) [no] switchport vlan scope local 例 : <pre>apic1(config-leaf-if) # switchport vlan scope local</pre>	デフォルトでは、VLAN の範囲はノードに対してグローバルです。1 つの VLAN は、ノードの 1 つだけの EPG にマッピングできます。VLAN の範囲がポートに対してローカルの場合、VLAN からの EPG へのマッピングは、同じノード上の別のポートでは異なります。 To return the scope to global, use the no command prefix.
ステップ 7	show vlan-domain [name domain-name] [vlan vlan-id] [leaf leaf-id] 例 : <pre>apic1(config-leaf-if) # show vlan-domain name dom1 vlan 1002 leaf 102</pre>	アプリケーション EPG、外部 SVI および外部 L2 などのアプリケーションの VLAN ドメインの使用状況を表示します。

例

この例では、ポートに VLAN ドメインを関連付ける方法を示します。

```
apic1# configure
(config) # leaf 101-102
(config-leaf) # int eth 1/1-24
(config-leaf-if) # vlan-domain member dom1

(config-leaf) # int eth 1/1-12
(config-leaf-if) # no switchport
(config-leaf) # int eth 1/13-24
(config-leaf-if) # switchport

(config) # leaf 101-102
(config-leaf) # int eth 1/1-12
(config-leaf-if) # switchport vlan scope local

(config-leaf) # int eth 1/13
(config-leaf-if) # no switchport vlan scope local
```

ポート チャネルへの VLAN ドメインの関連付け

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id1-node-id2 例 : <code>apic1(config)# leaf 101-102</code>	設定するリーフのペアを指定します。
ステップ 3	interface port-channel port-channel-name 例 : <code>apic1(config-leaf)# int port-channel pc1</code>	VLAN ドメインに関連付けられるポート チャネルを指定します。
ステップ 4	[no] vlan-domain member domain-name 例 : <code>apic1(config-leaf-if)# vlan-domain member dom1</code>	VLAN ドメインに指定したポート チャネルを割り当てます。

例

```
apic1# configure
apic1(config)# leaf 101-102
apic1(config-leaf)# int port-channel pc1
apic1(config-leaf-if)# vlan-domain member dom1
```

テンプレート ポリシー グループへの VLAN ドメインの関連付け

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	template policy-group policy-group-name 例 :	設定するテンプレート ポリシー グループを指定します。

	コマンドまたはアクション	目的
	<code>apic1(config)# template policy-group myPolGp5</code>	
ステップ 3	[no] vlan-domain member domain-name 例 : <code>apic1(config-pol-grp-if)# vlan-domain member dom1</code>	VLAN ドメインに指定したテンプレート ポリシー グループを割り当てます。

例

```
apic1# configure
apic1(config)# template policy-group myPolGp5
apic1(config-pol-grp-if)# vlan-domain member dom1
```

テンプレート ポート チャネルへの VLAN ドメインの関連付け

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	template port-channel policy-group-name 例 : <code>apic1(config)# template port-channel myPC7</code>	設定するテンプレート ポート チャネルを指定します。
ステップ 3	[no] vlan-domain member domain-name 例 : <code>apic1(config-if)# vlan-domain member dom1</code>	VLAN ドメインに指定したテンプレート ポート チャネルを割り当てます。

例

```
apic1# configure
apic1(config)# template port-channel myPC7
apic1(config-po-ch-if)# vlan-domain member dom1
```


バーチャル ポート チャネルへの VLAN ドメインの関連付け

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	vpcontext fex id1 node-id2 [fex-id1 fex-id2] leaf 例 : <code>apic1(config)# vpc context leaf 101 102</code>	設定する VPC およびリーフを指定します。
ステップ 3	interface vpc vpc-name [fex fex-id1 fex-id2] 例 : <code>apic1(config-vpc)# int vpc vpc1</code>	VLAN ドメインに関連付けられるポート チャネルを指定します。
ステップ 4	[no] vlan-domain member domain-name 例 : <code>apic1(config-vpc-if)# vlan-domain member dom1</code>	VLAN ドメインに指定した VPC を割り当てます。

例

```
apic1# configure
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# int vpc vpc1
apic1(config-vpc-if)# vlan-domain member dom1
```

EPG の Q-in-Q カプセル化マッピングの設定

EPG の Q-in-Q カプセル化マッピング

Cisco APIC を使用して、通常のインターフェイスまたは VPC で二重タグ付き VLAN トラフィック入力を EPG にマッピングできます。この機能が有効になっている場合、二重タグ付きトラフィックが EPG のネットワークに入るとき、両方のタグがファブリックで個別に処理され、ACI スイッチを入力するとき二重タグに復元されます。単一タグおよびタグなしのトラフィックの入力はドロップします。

この機能は、Nexus 9300-FX プラットフォーム スイッチでのみサポートされています。

外側と内側の両方のタグは、EtherType 0x8100 である必要があります。

MAC ラーニングおよびルーティングは、カプセル化のアクセスではなく、EPG ポート、sclass、VRF に基づいています。

QoS 優先度設定がサポートされ、入力の外側のタグから派生し、出力の両方のタグに書き換えられます。

EPG はリーフ スイッチの他のインターフェイスに同時に関連付けることができ、単一タグの VLAN に設定されます。

サービス グラフは、Q-in-Q カプセル化したインターフェイスにマッピングされているプロバイダとコンシューマ EPG をサポートしています。サービス ノードの入力および出力トラフィックが単一タグのカプセル化フレームにある限り、サービス グラフを挿入することができます。

この機能では、次の機能とオプションがサポートされていません。

- ポートごとの VLAN 機能
- FEX 接続
- 混合モードはサポートされていません。たとえば、Q-in-Q カプセル化モードのインターフェイスでは、通常の VLAN のカプセル化ではなく、二重タグ付きカプセルのみを持つ EPG にバインディングされている静的パスを有します。
- STP および「カプセル化でフラッドング」オプション
- タグなしおよび 802.1p モード
- マルチポッドと複数サイト
- レガシブリッジ ドメイン
- L2Out および L3Out 接続
- VMM の統合
- Q-in-Q カプセル化モードにルーティングされるポート モードの変更はサポートされていません
- Q-in-Q カプセル化モードのポートと通常のトランク モードのポート間では、各 Vlan MCP はサポートされていません。
- VPC ポートが Q-in-Q カプセル化モードを有効にしている場合、VLAN の整合性チェックは行われません。

NX-OS スタイル CLI を使用した Q-in-Q カプセル化リーフ インターフェイスへの EPG のマッピング

Q-in-Q カプセル化のインターフェイスを有効にし、EPG にインターフェイスを関連付けます。

始める前に

Q-in-Q モードに設定されているインターフェイスでマッピングされるテナント、アプリケーション プロファイル、アプリケーション EPG を作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	Configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf number 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	interface ethernet slot/port 例 : apic1 (config-leaf)# interface ethernet 1/25	設定するインターフェイスを指定します。
ステップ 4	switchport mode dot1q-tunnel doubleQtagPort 例 : apic1(config-leaf-if)# switchport mode dot1q-tunnel doubleQtagPort	Q-in-Q カプセル化のインターフェイスを有効にします。
ステップ 5	switchport trunk qinq outer-vlan vlan-number inner-vlan vlan-number tenant tenant-name application application-name epg epg-name 例 : apic1(config-leaf-if)# switchport trunk qinq outer-vlan 202 inner-vlan 203 tenant tenant64 application AP64 epg EPG64	インターフェイスを EPG に関連付けます。

例

次の例では、リーフ インターフェイス 101/1/25 で Q-in-Q カプセル化を有効にして（VLAN ID 201 外部および VLAN ID 203 内部）、EPG64 にインターフェイスを関連付けます。

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/25
apic1(config-leaf-if)# switchport mode dot1q-tunnel doubleQtagPort
apic1(config-leaf-if)# switchport trunk qinq outer-vlan 202 inner-vlan 203 tenant tenant64 application AP64 epg EPG64
```

ACI ファブリックでイーサネットトラフィックを介したファイバチャネルのサポート

ACI ファブリック上のイーサネットトラフィックによるファイバチャネルをサポートする

ACIを使用すると、設定し、(FCoE)トラフィック ACI ファブリックの Fibre Channel over Ethernet のサポートを管理できます。

FCoE は、ファイバチャネル SAN からイーサネットネットワークにシームレスに移動するストレージトラフィックを有効になり、イーサネットパケット内のファイバチャネル (FC) パケットをカプセル化するプロトコルです。

FCoE プロトコルのサポート ACI ファブリックでの一般的な実装には、FC ネットワーク上にある SAN ストレージデバイスとの通信に ACI ファブリックのイーサネットベースの上に存在するホストが有効になります。ホストは、ACI Leaf スイッチに導入仮想の F ポートを介して接続しています。仮想 F ポートが同じ ACI Leaf スイッチに導入する仮想 NP ポートを介して ACI ファブリックへの FCF ブリッジでは、SAN ストレージデバイスおよび FC ネットワークが接続されています。仮想 NP ポートおよび仮想 F ポートも汎用的に仮想ファイバチャネル (vFC) ポートと呼ばれます。



(注) リリースバージョン 2.0(1)、においては、FCoE サポートはハードウェアです N9K-C93180YC-EX と N9K-C93108TC-EX に制限されます。リリースバージョン 2.2(1) と N9K-C93180LC-EX 40 ギガビットイーサネット (GE) ポートは F または NP ポートとして使用できます。ただし、FCoE を許可されている場合は、40GE ポートブレイクアウトを対応することはできません。FCoE はブレイクアウトポートではサポートされていません。

リリースバージョン 2.2(2) からは、N9K-C93180YC-FX および N9K-C93108TC-FX ハードウェアは、FCoE をサポートします。リリース 2.3(1) では、ハードウェア N9K-C93180YC-FX および N9K-C93108TC-FX Fex ポート上での FCoE サポートは使用できます。

リリースバージョン 2.2(x)、において FCoE は、次の FEX Nexus デバイスではサポートも。

- 10 ギガ-ビット C2348UPQ N2K
- 10 ギガ-ビット C2348TQ N2K
- N2K-C2232PP-10GE
- N2K-B22DELL-P
- N2K-B22HP-P
- N2K-B22IBM-P
- N2K B22DELL P FI

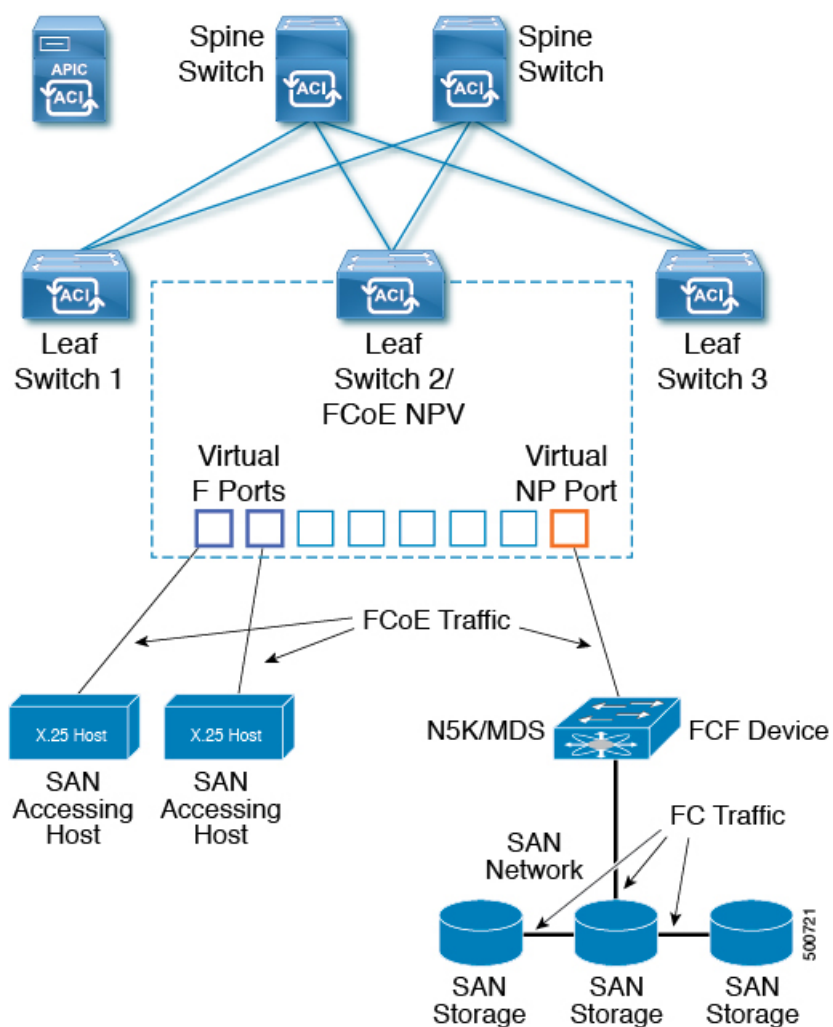


- (注) FCoE の使用されている vlan には、グローバルに設定 `vlanScope` 必要があります。FCoE の `vlanScope portLocal` に設定はサポートされていません。値は、L2 インターフェイス ポリシー `l2IfPol` 経由で設定されます。

FCoE ACI を通過するトラフィックをサポートするトポロジ

ACI ファブリック over FCoE トラフィックをサポートする一般的な設定のトポロジは、次のコンポーネントで構成されています。

図 5: ACI トポロジサポート FCoE トラフィック



- [FC SAN を介して NPV バックボーンとして機能するためのポリシーに設定された 1 つ以上の ACI Leaf スイッチ
- F ポートとして機能するように設定 NPV 設定のリーフ スイッチ上のインターフェイスを選択します。

F ポートでは、FCoE トラフィックと SAN 管理または SAN を消費しているアプリケーションを実行しているホストの対応します。

- NP ポートとして機能する NPV 設定のリーフ スイッチ上のインターフェイスを選択します。

NP ポートでは、FCoE トラフィックと FCF ブリッジの対応します。

FCF ブリッジは、通常は SAN ストレージデバイスを接続するファイバチャネルのリンクから FC トラフィックを受信し、SAN 管理または SAN のデータを消費しているホストに ACI ファブリック上を伝送するための FCoE フレームに FC パケットをカプセル化します。FCoE トラフィックを受信し、ファイバチャネルネットワーク経由で伝送するための FC に戻る再します。



(注) 上記の ACI トポロジでは、FCoE トラフィックのサポートには、ホストと F ポートおよび FCF デバイスと NP ポート間の直接接続間の直接接続が必要です。

APIC サーバを設定し、APIC gui FCoE トラフィックをモニタ演算子を有効にする、APIC NX-OS スタイルの CLI、またはアプリケーションで、APIC REST API をコールします。

FCoE の初期化をサポートするトポロジ

FCoE トラフィックフローの説明に従ってを行われるするためには、する必要もあります SAN ホストが F ポートとして有効になっているインターフェイスを検出する、FCoE Initialization protocol (FIP) パケットがブロードキャストされる別の VLAN 接続を設定します。

vFC インターフェイス設定ルール

VFC のネットワークと APIC GUI、NX-OS スタイル CLI、または REST API を通じて EPG の導入を設定するかどうかは、プラットフォーム、次の一般的なルールが適用されます。

- F ポートモードは、vFC ポートのデフォルトモードです。NP ポートモードは、インターフェイス ポリシーで具体的に設定する必要があります。
- デフォルトのロードバランシングモードはリーフ スイッチ、またはインターフェイスレベル vFC 設定が src dst ox id。
- ブリッジドメインごとに 1 つの VSAN 割り当てがサポートされます。
- VSAN プールおよび VLAN プールの割り当てモードは、常にスタティックである必要があります。
- vFC のポートには、VSAN (ファイバチャネルドメインとも呼ばれる) が含まれるドメインの VLAN にマッピングされた VSAN との関連付けが必要です。

FCoE NX-OS スタイル CLI 設定

NX-OSスタイルCLIを使用したポリシーまたはプロファイルのないFCoE接続の設定

次の例の NX-OS スタイル CLI シーケンス EPG の FCoE 接続を設定する **e1** テナントで **t1** 設定またはスイッチ レベルとインターフェイス レベル ポリシーとプロファイルを適用せず。

手順

	コマンドまたはアクション	目的
ステップ 1	ターゲットテナントの下には、FCoE トラフィックをサポートするブリッジ ドメインを設定します。 例 : <pre>apic1(config)# tenant t1 apic1(config-tenant)# vrf context v1 apic1(config-tenant-vrf)# exit apic1(config-tenant)# bridge-domain b1 apic1(config-tenant-bd)# fc apic1(config-tenant-bd)# vrf member v1 apic1(config-tenant-bd)# exit apic1(config-tenant)# exit</pre>	サンプル コマンド シーケンスはブリッジ ドメインを作成 b1 テナントで t1 FCoE 接続をサポートするように設定します。
ステップ 2	同じのテナントの下には、FCoE に設定されたブリッジ ドメインとターゲット EPG を関連付けます。 例 : <pre>apic1(config)# tenant t1 apic1(config-tenant)# application a1 apic1(config-tenant-app)# epg e1 apic1(config-tenant-app-epg)# bridge-domain member b1 apic1(config-tenant-app-epg)# exit apic1(config-tenant-app)# exit apic1(config-tenant)# exit</pre>	サンプル コマンド シーケンス作成 EPG e1 し、FCoE に設定されたブリッジ ドメインにその EPG を関連付けます b1。
ステップ 3	VLAN マッピングに VSAN ドメイン、VSAN プール、VLAN プール、VSAN を作成します。 例 : A <pre>apic1(config)# vsan-domain dom1 apic1(config-vsan)# vsan 1-10 apic1(config-vsan)# vlan 1-10 apic1(config-vsan)# fcoe vsan 1 vlan 1 loadbalancing src-dst-ox-id</pre>	例 A 、サンプル コマンド シーケンスは、VSAN ドメインを作成 dom1 VSAN プールと VLAN プール、VSAN 1 を VLAN 1 にマッピングされ、VLAN 2 に VSAN 2 をマップ 例 B 、代替サンプル コマンド シーケンスは再利用可能な VSAN 属性テンプレートを作成 pol1 VSAN ドメインを作成し、dom1 、そのテンプレートから属性とマッピングを継承します。

	コマンドまたはアクション	目的
	<pre>apicl(config-vsant)# fcoe vsan 2 vlan 2</pre> 例 : B <pre>apicl(config)# template vsan-attribute poll apicl(config-vsant-attr)# fcoe vsan 2 vlan 12 loadbalancing src-dst-ox-id apicl(config-vsant-attr)# fcoe vsan 3 vlan 13 loadbalancing src-dst-ox-id apicl(config-vsant-attr)# exit apicl(config)# vsan-domain dom1 apicl(config-vsant)# vsan 1-10 apicl(config-vsant)# vlan 1-10 apicl(config-vsant)# inherit vsan-attribute poll apicl(config-vsant)# exit</pre>	
ステップ 4	FCoE Initialization (FIP) プロセスをサポートする物理ドメインを作成します。 例 : <pre>apicl(config)# vlan-domain fipVlanDom apicl(config-vlan)# vlan 120 apicl(config-vlan)# exit</pre>	例では、コマンドシーケンスは、通常の VLAN ドメインを作成 fipVlanDom 、VLAN を含む 120 FIP プロセスをサポートします。
ステップ 5	ターゲットテナントの下には、定期的なブリッジドメインを設定します。 例 : <pre>apicl(config)# tenant t1 apicl(config-tenant)# vrf context v2 apicl(config-tenant-vrf)# exit apicl(config-tenant)# bridge-domain fip-bd apicl(config-tenant-bd)# vrf member v2 apicl(config-tenant-bd)# exit apicl(config-tenant)# exit</pre>	コマンドシーケンスがブリッジドメインを作成例では、 fip bd 。
ステップ 6	同じテナントの下には、設定されている定期的なブリッジドメインでこの EPG を関連付けます。 例 : <pre>apicl(config)# tenant t1 apicl(config-tenant)# application a1 apicl(config-tenant-app)# epg epg-fip apicl(config-tenant-app-epg)# bridge-domain member fip-bd apicl(config-tenant-app-epg)# exit apicl(config-tenant-app)# exit apicl(config-tenant)# exit</pre>	例では、コマンドシーケンス関連付けます EPG epg fip ブリッジドメインを fip bd 。

	コマンドまたはアクション	目的
ステップ 7	VFC インターフェイスを F モードで設定します。 例 : A <pre> apic1(config)# leaf 101 apic1(config-leaf)# interface ethernet 1/2 apic1(config-leaf-if)# vlan-domain member fipVlanDom apic1(config-leaf-if)# switchport trunk native vlan 120 tenant t1 application a1 epg epg-fip apic1(config-leaf-if)# exit apic1(config-leaf)# exit apic1(config-leaf)# interface vfc 1/2 apic1(config-leaf-if)# switchport mode f apic1(config-leaf-if)# vsan-domain member dom1 apic1(config-leaf-if)# switchport vsan 2 tenant t1 application a1 epg e1 apic1(config-leaf-if)# switchport trunk allowed vsan 3 tenant t1 application a1 epg e2 apic1(config-leaf-if)# exit </pre> 例 : B <pre> apic1(config)# vpc context leaf 101 102 apic1(config-vpc)# interface vpc vpc1 apic1(config-vpc-if)# vlan-domain member vfdom100 apic1(config-vpc-if)# vsan-domain member dom1 apic1(config-vpc-if)# #For FIP discovery apic1(config-vpc-if)# switchport trunk native vlan 120 tenant t1 application a1 epg epg-fip apic1(config-vpc-if)# switchport vsan 2 tenant t1 application a1 epg e1 apic1(config-vpc-if)# exit apic1(config-vpc)# exit apic1(config)# leaf 101-102 apic1(config-leaf)# interface ethernet 1/3 apic1(config-leaf-if)# channel-group vpc1 vpc apic1(config-leaf-if)# exit apic1(config-leaf)# exit </pre> 例 : C	例では A コマンドシーケンスは、インターフェイスを有効に 1/2 リーフ スイッチで 101 として機能する、F ポートおよびインターフェイスの VSAN のドメインに関連 dom1 。 ネイティブ モードで 1 つ (と 1 つだけ) の VSAN 対象のインターフェイスの各割り当てる必要があります。各インターフェイスには、通常モードで 1 つ以上の追加 Vsan を割り当てることができます。 サンプル コマンドシーケンスは、対象のインターフェイスを関連付けます 1/2 と。 • VLAN 120 FIP ディスカバリの EPG に関連付けます epg fip およびアプリケーション a1 テナントで t1 。 • VSAN 2 ネイティブ VSAN として、EPG に関連付けます e1 およびアプリケーション a1 テナントで t1 。 • VSAN 3 定期的な VSAN として。 例では B 、コマンドシーケンスは、両方のログ上で同じ、VSAN で VPCover vFC を設定します。CLI からログごとに異なる Vsan を指定することはできません。代替設定は、GUI を高度な apic 内で実行できます。

	コマンドまたはアクション	目的
	<pre> apicl(config)# leaf 101 apicl(config-leaf)# interface vfc-po pcl apicl(config-leaf-if)# vsan-domain member dom1 apicl(config-leaf-if)# switchport vsan 2 tenant t1 application a1 epg e1 apicl(config-leaf-if)# exit apicl(config-leaf)# interface ethernet 1/2 apicl(config-leaf-if)# channel-group pcl apicl(config-leaf-if)# exit apicl(config-leaf)# exit </pre>	
ステップ 8	VFC インターフェイスを NP モードで設定します。 例 : <pre> apicl(config)# leaf 101 apicl(config-leaf)# interface vfc 1/4 apicl(config-leaf-if)# switchport mode np apicl(config-leaf-if)# vsan-domain member dom1 </pre>	サンプル コマンド シーケンスは、インターフェイスを有効に 1/4 リーフ スイッチで 101 として機能する、NP ポートおよびインターフェイスの VSAN のドメインに関連 dom1。
ステップ 9	VSAN を対象となる FCoE 対応インターフェイスに割り当てます。 例 : <pre> apicl(config-leaf-if)# switchport trunk allowed vsan 1 tenant t1 application a1 epg e1 apicl(config-leaf-if)# switchport vsan 2 tenant t4 application a4 epg e4 </pre>	ネイティブ モードで 1 つ (と 1 つだけ) の VSAN 対象のインターフェイスの各割り当てする必要があります。各インターフェイスには、通常モードで 1 つ以上の追加 Vsan を割り当てることができません。 サンプル コマンド シーケンスは、ターゲット インターフェイスを VSAN 1 に割り当て、それを EPG e1 とアプリケーション a1 にテナント t1 の下に関連付けます。「trunk allowed」は、VSAN 1 に通常モードのステータスを割り当てます。コマンド シーケンスも割り当てます、インターフェイス、必要な ネイティブ モード VSAN 2。次の例に示すは、同一のインターフェイスを異なるテナント アクセスで実行されているさまざまな Epg を提供するためにさまざまな Vsan の動作を渡します。

NX-OSスタイルCLIを使用したポリシーまたはプロファイルがあるFCoE接続の設定

次の例 NX-OS スタイル CLI のシーケンスを作成し、EPG の FCoE 接続を設定するポリシーを使用して **e1** テナントで **t1**。

手順

	コマンドまたはアクション	目的
ステップ 1	ターゲット テナントの下には、FCoE トラフィックをサポートするブリッジ ドメインを設定します。 例 : <pre>apicl# configure apicl(config)# tenant t1 apicl(config-tenant)# vrf context v1 apicl(config-tenant-vrf)# exit apicl(config-tenant)# bridge-domain b1 apicl(config-tenant-bd)# fc apicl(config-tenant-bd)# vrf member v1 apicl(config-tenant-bd)# exit apicl(config-tenant)# exit apicl(config)#</pre>	サンプル コマンド シーケンスはブリッジ ドメインを作成 b1 テナントで t1 FCoE接続をサポートするように設定します。
ステップ 2	同じのテナントの下には、設定されている FCoE ブリッジ ドメインと、ターゲット EPG を関連付けます。 例 : <pre>apicl(config)# tenant t1 apicl(config-tenant)# application a1 apicl(config-tenant-app)# epg e1 apicl(config-tenant-app-epg)# bridge-domain member b1 apicl(config-tenant-app-epg)# exit apicl(config-tenant-app)# exit apicl(config-tenant)# exit apicl(config)#</pre>	サンプル コマンド シーケンス作成 EPG e1 その EPG の FCoE に設定されたブリッジ ドメイン関連付け b1 。
ステップ 3	VLAN マッピングに VSAN ドメイン、VSAN プール、VLAN プール、VSAN を作成します。 例 : A <pre>apicl(config)# vsan-domain dom1 apicl(config-vsan)# vsan 1-10 apicl(config-vsan)# vlan 1-10 apicl(config-vsan)# fcoe vsan 1 vlan</pre>	例 A、サンプル コマンド シーケンスは、VSAN ドメインを作成 dom1 VSAN プールと VLAN プール、マップ VSAN 1 VLAN 1 と VLAN 2 に VSAN 2 をマップ 例 B、代替サンプル コマンド シーケンスは再利用可能な vsan 属性テンプレートを作成 pol1 VSAN ドメインを

	コマンドまたはアクション	目的
	<pre> 1 loadbalancing src-dst-ox-id apic1(config-vsan)# fcoe vsan 2 vlan 2 例 : B apic1(config)# template vsan-attribute poll apic1(config-vsan-attr)# fcoe vsan 2 vlan 12 loadbalancing src-dst-ox-id apic1(config-vsan-attr)# fcoe vsan 3 vlan 13 loadbalancing src-dst-ox-id apic1(config-vsan-attr)# exit apic1(config)# vsan-domain dom1 apic1(config-vsan)# inherit vsan-attribute poll apic1(config-vsan)# exit </pre>	作成し、 dom1 、そのテンプレートから属性とマッピングを継承します。
ステップ 4	FCoE Initialization (FIP) プロセスをサポートする物理ドメインを作成します。 例 : <pre> apic1(config)# vlan-domain fipVlanDom apic1(config)# vlan-pool fipVlanPool </pre>	
ステップ 5	ファイバチャネル SAN ポリシーを設定します。 例 : <pre> apic1# apic1# configure apic1(config)# template fc-fabric-policy ffp1 apic1(config-fc-fabric-policy)# fctimer e-d-tov 1111 apic1(config-fc-fabric-policy)# fctimer r-a-tov 2222 apic1(config-fc-fabric-policy)# fcoe fcmap 0E:FC:01 apic1(config-fc-fabric-policy)# exit </pre>	サンプルコマンドシーケンスは、SAN のファイバチャネル ポリシーを作成 ffp1 の組み合わせを指定するエラー検出タイムアウト値 (EDTOV)、resource allocation(リソース割り当て、リソースの割り当て)タイムアウト値(RATOV)、およびターゲット リーフ上の FCoE 対応のインターフェイスのデフォルト FC マップ値スイッチです。
ステップ 6	ファイバチャネル ノード ポリシーを作成します。 例 : <pre> apic1(config)# template fc-leaf-policy flp1 apic1(config-fc-leaf-policy)# fcoe fka-adv-period 44 apic1(config-fc-leaf-policy)# exit </pre>	サンプルコマンドシーケンスは、ファイバチャネルノードのポリシーを作成 flp1 を中断のロードバランシングの有効化と FIP キープアライブ値の組み合わせを指定します。これらの値は、ターゲット リーフスイッチ上のすべて

	コマンドまたはアクション	目的
		の FCoE 対応インターフェイスにも適用されます。
ステップ 7	ノード ポリシー グループを作成します。 例 : <pre> apic1(config)# template leaf-policy-group lpg1 apic1(config-leaf-policy-group)# inherit fc-fabric-policy ffp1 apic1(config-leaf-policy-group)# inherit fc-leaf-policy flp1 apic1(config-leaf-policy-group)# exit apic1(config)# exit apic1# </pre>	サンプルコマンドシーケンスはノード ポリシー グループを作成 lpg1、SAN のファイバチャネルポリシーの値を結合する ffp1 とファイバチャネル ノードのポリシー、 flp1。このノード ポリシーグループの合計値は、後で設定されているノードのプロファイルに適用できます。
ステップ 8	ノード プロファイルを作成します。 例 : <pre> apic1(config)# leaf-profile lp1 apic1(config-leaf-profile)# leaf-group lg1 apic1(config-leaf-group)# leaf 101 apic1(config-leaf-group)# leaf-policy-group lpg1 </pre>	サンプルコマンドシーケンスがノードのプロファイルを作成 lp1 ノード ポリシー グループと関連付けます lpg1、ノードグループ lg1、およびリーフ スイッチ 101。
ステップ 9	F ポート インターフェイスのインターフェイス ポリシー グループを作成します。 例 : <pre> apic1(config)# template policy-group ipg1 apic1(config-pol-grp-if)# priority-flow-control mode auto apic1(config-pol-grp-if)# switchport mode f apic1(config-pol-grp-if)# slow-drain pause timeout 111 apic1(config-pol-grp-if)# slow-drain congestion-timeout count 55 apic1(config-pol-grp-if)# slow-drain congestion-timeout action log </pre>	サンプルコマンドシーケンスは、インターフェイスグループのポリシーを作成 ipg1 し、プライオリティ フロー制御の有効化、F ポートの有効化、およびこのポリシー グループに適用されているすべてのインターフェイスに対して低速ドレインポリシーの値を決定する値の組み合わせを割り当てます。
ステップ 10	NP ポート インターフェイスのインターフェイス ポリシー グループを作成します。 例 : <pre> apic1(config)# template policy-group ipg2 apic1(config-pol-grp-if)# priority-flow-control mode auto apic1(config-pol-grp-if)# switchport mode np </pre>	サンプルコマンドシーケンスは、インターフェイスグループ ポリシー ipg2 を作成し、このポリシーグループに適用されているすべてのインターフェイスに対して、優先順位フロー制御の有効化、NP ポートの有効化、低速ドレインポリシーの値を決定する値の組み合わせを割り当てます。

	コマンドまたはアクション	目的
	<pre>apic1(config-pol-grp-if)# slow-drain pause timeout 111 apic1(config-pol-grp-if)# slow-drain congestion-timeout count 55 apic1(config-pol-grp-if)# slow-drain congestion-timeout action log</pre>	
ステップ 11	F ポート インターフェイスのインターフェイス プロファイルを作成します。 例 : <pre>apic1# configure apic1(config)# leaf-interface-profile lip1 apic1(config-leaf-if-profile)# description 'test description lip1' apic1(config-leaf-if-profile)# leaf-interface-group lig1 apic1(config-leaf-if-group)# description 'test description lig1' apic1(config-leaf-if-group)# policy-group ipg1 apic1(config-leaf-if-group)# interface ethernet 1/2-6, 1/9-13</pre>	サンプルコマンドシーケンスは、インターフェイス プロファイルを作成 lip1 F ポートのインターフェイスの F ポートの特定のインターフェイス ポリシーグループ プロファイルに関連付けます ipg1、このインターフェイスを指定し プロファイルとその関連するポリシー。適用されます。
ステップ 12	NP ポート インターフェイスのインターフェイス プロファイルを作成します。 例 : <pre>apic1# configure apic1(config)# leaf-interface-profile lip2 apic1(config-leaf-if-profile)# description 'test description lip2' apic1(config-leaf-if-profile)# leaf-interface-group lig2 apic1(config-leaf-if-group)# description 'test description lig2' apic1(config-leaf-if-group)# policy-group ipg2 apic1(config-leaf-if-group)# interface ethernet 1/14</pre>	サンプルコマンドシーケンスは、インターフェイス プロファイルを作成 lip2 NP ポート インターフェイス、NP ポートの特定のインターフェイス ポリシーグループ プロファイルに関連付けます ipg2、このインターフェイスを指定し、プロファイルとその関連するポリシー適用されます。
ステップ 13	レベル 1 の QoS クラス ポリシーを設定します。 例 : <pre>apic1(config)# qos parameters level1 apic1(config-qos)# pause no-drop cos 3</pre>	サンプルコマンドシーケンスは、FCoE トラフィック プライオリティ フロー制御ポリシーを適用することがおよび非ドロップパケットのクラスのサービスレベル 3 の処理を一時停止の QoS レベルを指定します。

NX-OS スタイル CLI を使用して FCoE オーバー FEX の設定

FEX ポートは、ポート Vsan として設定されます。

手順

ステップ 1 テナントと VSAN のドメインを設定します。

例：

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain b1
apic1(config-tenant-bd)# fc
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# application a1
apic1(config-tenant-app)# epg e1
apic1(config-tenant-app-epg)# bridge-domain member b1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit

apic1(config)# vsan-domain dom1
apic1(config-vsan)# vlan 1-100
apic1(config-vsan)# vsan 1-100
apic1(config-vsan)# fcoe vsan 2 vlan 2 loadbalancing src-dst-ox-id
apic1(config-vsan)# fcoe vsan 3 vlan 3 loadbalancing src-dst-ox-id
apic1(config-vsan)# fcoe vsan 5 vlan 5
apic1(config-vsan)# exit
```

ステップ 2 FEX をインターフェイスに関連付けます。

例：

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/12
apic1(config-leaf-if)# fex associate 111
apic1(config-leaf-if)# exit
```

ステップ 3 ポート、ポート チャネル、および VPC あたり FEX を介して FCoE を設定します。

例：

```
apic1(config-leaf)# interface vfc 111/1/2
apic1(config-leaf-if)# vsan-domain member dom1
apic1(config-leaf-if)# switchport vsan 2 tenant t1 application a1 epg e1
apic1(config-leaf-if)# exit

apic1(config-leaf)# interface vfc-po pc1 fex 111
apic1(config-leaf-if)# vsan-domain member dom1
apic1(config-leaf-if)# switchport vsan 2 tenant t1 application a1 epg e1
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 111/1/3
apic1(config-leaf-if)# channel-group pc1
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

```

apic1(config)# vpc domain explicit 12 leaf 101 102
apic1(config-vpc)# exit
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# interface vpc vpc1 fex 111 111
apic1(config-vpc-if)# vsan-domain member dom1
apic1(config-vpc-if)# switchport vsan 2 tenant t1 application a1 epg e1
apic1(config-vpc-if)# exit
apic1(config-vpc)# exit
apic1(config)# leaf 101-102
apic1(config-leaf)# interface ethernet 1/2
apic1(config-leaf-if)# fex associate 111
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 111/1/2
apic1(config-leaf-if)# channel-group vpc1 vpc
apic1(config-leaf-if)# exit

```

ステップ 4 設定を確認するには、次のコマンドを実行します。

例：

```

apic1(config-vpc)# show vsan-domain detail
vsan-domain : dom1

```

```

vsan : 1-100

```

```

vlan : 1-100

```

Leaf	Interface	Vsan	Vlan	Vsan-Mode	Port-Mode	Usage
Operational	State					
101	vfc111/1/2	2	2	Native		Tenant: t1
Deployed						App: a1 Epg: e1
101	PC:pc1	5	5	Native		Tenant: t1
Deployed						App: a1 Epg: e1
101	vfc111/1/3	3	3	Native	F	Tenant: t1
Deployed						App: a1 Epg: e1

NX-OS スタイルの CLI を使用した FCoE 設定の検証

次 **show** コマンドは、リーフ スイッチ ポートで FCoE の設定を確認します。

手順

使用して、**vsan ドメインを表示** コマンドをターゲット スイッチで FCoE が有効になっていることを確認します。

コマンドの例では、FCoE がリストされているリーフ スイッチおよび接続の詳細を FCF で有効になっていることを確認します。

例：

```
ifav-isim8-ifc1# show vsan-domain detail
```

```
vsan-domain : iPostfcoeDomP1
```

```
vsan : 1-20 51-52 100-102 104-110 200 1999 3100-3101 3133
      2000
```

```
vlan : 1-20 51-52 100-102 104-110 200 1999 3100-3101 3133
      2000
```

Leaf	Interface	Vsan	Vlan	Vsan Mode	Port Mode	Usage	Operational State
101	vfc1/11	1	1	Regular	F	Tenant: iPost101 App: iPost1 Epg: iPost1	Deployed
101	vfc1/12	1	1	Regular	NP	Tenant: iPost101 App: iPost1 Epg: iPost1	Deployed
101	PC:infraAccBnd1 Grp_pc01	4	4	Regular	NP	Tenant: iPost101 App: iPost4 Epg: iPost4	Deployed
101	vfc1/30	2000		Native		Tenant: t1 App: a1 Epg: e1	Not deployed (invalid-path)

NX-OS スタイル CLI を使用した FCoE 要素の展開解除

ACI ファブリックから FCoE 接続を導入解除に移動してもでは、いくつかのレベルで FCoE コンポーネントを削除する必要があります。

手順

- ステップ 1** リーフ ポート インターフェイスの属性のリスト、そのモードの設定をデフォルトに設定し、その EPG の導入とドメインの関連付けを削除します。

インターフェイス **vfc** のポート モードの設定を設定する例 **1/2** のデフォルトに [EPG の導入を削除 **e1** と VSAN ドメインに関連付け **dom1** そのインターフェイスから。

例 :

```
apic1(config)# leaf 101
apic1(config-leaf)# interface vfc 1/2
apic1(config-leaf-if)# show run
# Command: show running-config leaf 101 interface vfc 1 / 2
# Time: Tue Jul 26 09:41:11 2016
  leaf 101
    interface vfc 1/2
      vsan-domain member dom1
      switchport vsan 2 tenant t1 application a1 epg e1
    exit
  exit
apic1(config-leaf-if)# no switchport mode
apic1(config-leaf-if)# no switchport vsan 2 tenant t1 application a1 epg e1
apic1(config-leaf-if)# no vsan-domain member dom1
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

- ステップ 2** 一覧表示し、VSAN マッピングおよび VLAN と VSAN のプールを削除します。

例の VSAN の VLAN マッピングが削除されます **vsan 2**、VLAN プール **1-10**、および VSAN プール **1-10** VSAN ドメインから **dom1**。

例 :

```
apic1(config)# vsan-domain dom1
apic1(config-vsan)# show run
# Command: show running-config vsan-domain dom1
# Time: Tue Jul 26 09:43:47 2016
  vsan-domain dom1
    vsan 1-10
    vlan 1-10
    fcoe vsan 2 vlan 2
  exit
apic1(config-vsan)# no fcoe vsan 2
apic1(config-vsan)# no vlan 1-10
apic1(config-vsan)# no vsan 1-10
apic1(config-vsan)# exit

#####
NOTE: To remove a template-based VSAN to VLAN mapping use an alternate sequence:
#####

apic1(config)# template vsan-attribute <template_name>
apic1(config-vsan-attr)# no fcoe vsan 2
```

- ステップ 3** VSAN ドメインを削除します。

例は、ドメインの VSAN を削除する **dom1**。

例 :

```
apic1(config)# no vsan-domain dom1
```

ステップ 4 必要はないかどうかは、関連付けられているテナント、EPG、およびセクタを削除できます。

ファイバチャネル NPV

ファイバチャネル接続の概要

スイッチは、NPV を有効にした後は NPV モードになります。NPV モードはスイッチ全体に適用されます。NPV モードのスイッチに接続するすべてのエンド デバイスは、N ポートとしてログインし、この機能を使用する必要があります（ループ接続デバイスはサポートされていません）。（NPV モードの）エッジスイッチから NPV コア スイッチへのすべてのリンクは、（E ポートではなく）NP ポートとして確立されます。このポートは、通常のスイッチ間リンクに使用されます。

FC NPV の利点

FC NPV では次の機能を提供します。

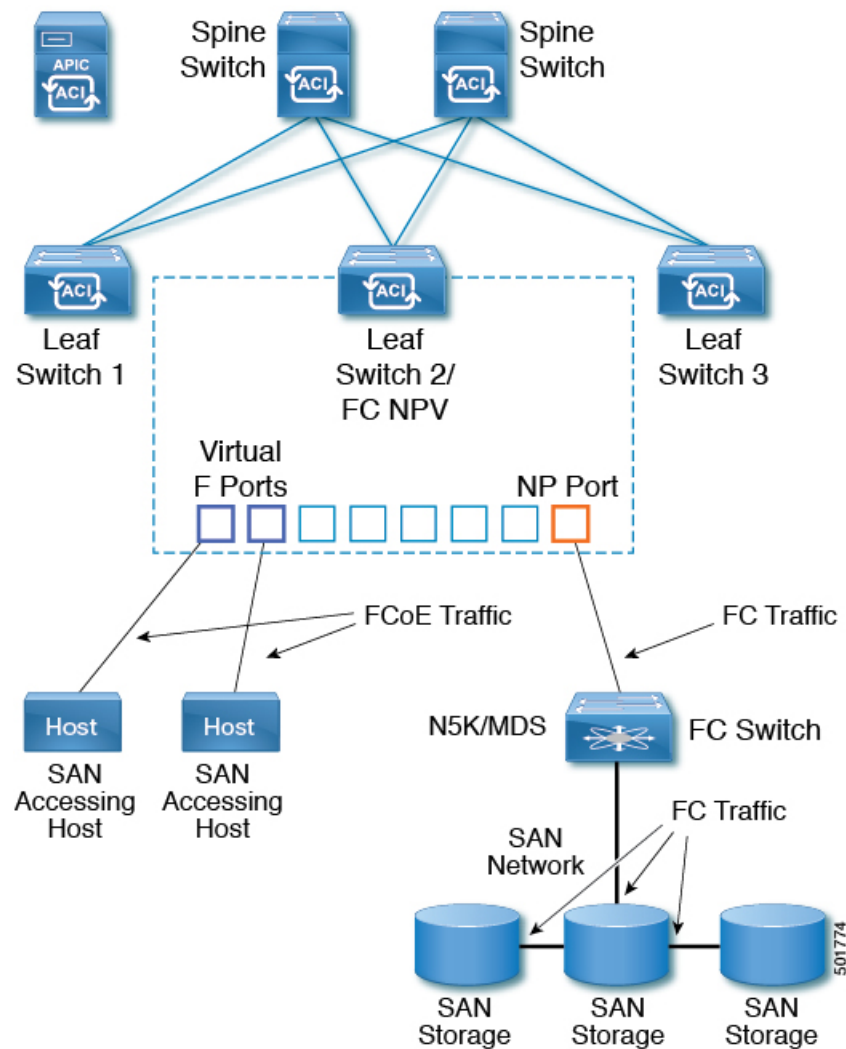
- ファブリックのドメイン ID を追加することがなく、ファブリックに接続しているホスト数の増加する
- FC および FCoE ホストとターゲットの FC インターフェイスを使用した SAN ファブリックへの接続
- 自動トラフィック マッピング
- スタティック トラフィック マッピング
- 自動ロード バランシングの中断

FC NPV モード

ACI の Feature-set fcoe-npv は、最初に FCoE/FC 設定がプッシュされるときに、デフォルトで自動的に有効になります。

FC トポロジ

ACI ファブリックで FC トラフィックをサポートする一般的な設定のトポロジは、次のコンポーネントで構成されています。



- リーフは、FCoE NP ポートまたはネイティブ FC NP ポートを使用して FC スイッチに接続できます。
- ACI リーフは、FCoE リンクを使用してサーバ/ストレージと直接接続することができます。
- FC/FCoE トラフィックは、ファブリック/スパインには送信されません。リーフ スイッチでは、FCoE トラフィックのローカル スイッチングは実行しません。スイッチングは、FC/FCoE NPV リンク経由でリーフ スイッチと接続されているコア スイッチによって行われます。
- Flogi に続く複数の FDISC は、FCoE ホストと FC/FCoE NP リンクによりサポートされます。

FC NPV の注意事項と制約事項

FC NPV を設定する場合、次の注意事項および制限事項に注意してください。

- FC 設定に使用できるポートの範囲は 1 ～ 48 です。ポート 49 ～ 54 までは FC ポートを変換できません。
- F ポートとしての FC ポート モードはサポートされていません。
- SAN ポート チャンネルがサポートされていません。
- FC NPV リンクで FEX 経由の FCoE ホストはサポートされていません。
- FC 速度の自動化および 32 G はサポートされていません。
- ACI リーフ 93180YC-FX ポートが 8 G 速度で設定されているとき、Brocade Port Blade FC16-32 への FC Uplink (NP) 接続はサポートされていません。
- FC トランク モード **ON** および **自動化** はサポートされていません。
- FC 塗りつぶしパターン ARBFF はサポートされていません。
- FC は 40 G およびブレイク アウト ポートではサポートされていません。
- FC ポートでは、FEX 起動はサポートされていません。
- FC NPV のサポートは、リリース 3.2(1) において N9K-C93180YC-FX に限られます。
- イーサネットから FC (またはその逆) のポート変換には、スイッチのリロードが必要です。現在 1 個のポートの連続範囲のみ FC ポートに変換可能で、この範囲はあり 4 の倍数で終わるポート番号である必要があります。例：1～4、1～8、または 21-24。

NX-OS CLI を使用したポリシーまたはプロファイルのない FC 接続の設定

次の例コマンドシーケンスは、FCoE 接続をサポートするように設定テナント t1 でブリッジドメイン b1 を作成します。

始める前に

- ターゲットテナントの下には、FCoE トラフィックをサポートするブリッジドメインを設定します。

手順

ステップ 1 FCoE 接続をブリッジドメインを作成します。

例：

```

apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain b1
apic1(config-tenant-bd)# fc
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# exit

```

ステップ 2 同じのテナントの下には、FCoE に設定されたブリッジドメインとターゲット EPG を関連付けます。次のコマンドシーケンスの例では、EPG e1 を作成し、FCoE に設定されたブリッジドメイン b1 にその EPG を関連付けます。

例：

```

apic1(config)# tenant t1
apic1(config-tenant)# application a1
apic1(config-tenant-app)# epg e1
apic1(config-tenant-app-epg)# bridge-domain member b1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit

```

ステップ 3 次の例では、1 ～ 10 の VSAN で VSAN ドメイン dom1 が作成されます。

例：

```

apic1(config)# vsan-domain dom1
apic1(config-vsan)# vsan 1-10

```

ステップ 4 ポートの範囲をイーサネットから FC モードに変換します。次の例では、スイッチのポート 1/1-4 を 101 から FC に変換します。

例：

```

apic1# config
apic1(config)# leaf 101
apic1(config-leaf)# slot 1
apic1(config-leaf-slot)# port 1 4 type fc
apic1(config-leaf-slot)# exit
apic1(config-leaf)# exit

```

(注) イーサネットから FC (およびその逆) へのポート変換には、スイッチのリロードが必要です。

ステップ 5 FC インターフェイスを NP モードに設定します。次の例では、インターフェイス fc 1/10 にさまざまなインターフェイスプロパティを設定し、VSAN ドメイン dom1 にそのインターフェイスを関連付けます。対象のインターフェイスのそれぞれが、ネイティブモードで 1 個 (そして 1 個のみ) の VSAN を割り当てる必要があります。サンプル コマンドシーケンスは、対象のインターフェイス 1/10 をネイティブ VSAN として VSAN 10 に関連付け、テナント t1 で EPG e1 およびアプリケーション a1 に関連付けます。

例：

```

apic1(config-leaf)# interface fc 1/10
apic1(config-leaf-fc-if)# switchport mode [f | np]
apic1(config-leaf-fc-if)# switchport rxbbcredit <16-64>
apic1(config-leaf-fc-if)# switchport speed [16G | 32G | 4G | 8G | auto | unknown]
apic1(config-leaf-fc-if)# vsan-domain member dom1
apic1(config-leaf-fc-if)# switchport vsan 10 tenant t1 application a1 epg e1

```

ステップ 6 サーバ ポートをアップリンク ポートにピンするためトラフィック マップを作成します。次の例では、FC 1/7 アップリンク インターフェイスにピンされている vFC 1/47 サーバ インターフェイスにトラフィック マップを作成します。

例：

```
apic1# config
apic1(config)# leaf 101
apic1(config-leaf)# npv traffic-map server-interface vfc 1/47 label label1 tenant tenant1
  application appl epg epg1
apic1(config-leaf)# npv traffic-map external-interface fc 1/7 tenant tenant1 label label1
```

ポリシーまたは NX-OS は、CLI を使用したプロファイルと FC 接続の設定

次の例コマンド シーケンスは、FCoE 接続をサポートするように設定テナント t1 でブリッジ ドメイン b1 を作成します。

始める前に

- ターゲットテナントの下には、FCoE トラフィックをサポートするブリッジ ドメインを設定します。

手順

ステップ 1 FCoE 接続をブリッジ ドメインを作成します。

例：

```
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain b1
apic1(config-tenant-bd)# fc
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# exit
```

ステップ 2 同じのテナントの下には、FCoE に設定されたブリッジ ドメインとターゲット EPG を関連付けます。次の例コマンド シーケンスは、EPG e1 を作成し、FCoE に設定されたブリッジ ドメイン b1 にその EPG を関連付けます。

例：

```
apic1(config)# tenant t1
apic1(config-tenant)# application a1
apic1(config-tenant-app)# epg e1
apic1(config-tenant-app-epg)# bridge-domain member b1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
```

ステップ 3 VSAN ドメインを作成します。次の例では、1~10 の vsan の vsan ドメイン dom1 が作成されます。

例：

```
apic1(config)# vsan-domain dom1
apic1(config-vsan)# vsan 1-10
```

ステップ 4 NP ポート インターフェイスのインターフェイス ポリシー グループを作成します。サンプル コマンド シーケンスは、FC インターフェイス ポリシー グループ ipg2 を作成し、このポリシー グループに適用されているインターフェイスの値を決定する値の組み合わせを割り当てます。

例：

```
apic1(config)# template fc-policy-group ipg1
apic1(config-fc-pol-grp-if)# switchport ?
  fill-pattern  Configure fill pattern for fc interface
  mode          Configure port mode for fc interface
  rxbbcredit    Configure rxBBCredit for fc interface
  speed         Configure speed for fc interface
  trunk-mode    Configure trunk-mode for fc interface
apic1(config-fc-pol-grp-if)# switchport fill-pattern [ARBFF | IDLE]
apic1(config-fc-pol-grp-if)# switchport mode [f | np]
apic1(config-fc-pol-grp-if)# switchport rxbbcredit <16-64>
apic1(config-fc-pol-grp-if)# switchport speed [16G | 32G | 4G | 8G | auto | unknown]
apic1(config-fc-pol-grp-if)# vsan-domain member dom1
```

ステップ 5 FC ポート インターフェイスのインターフェイス プロファイルを作成します。サンプル コマンド シーケンスの FC ポート インターフェイスのインターフェイス プロファイル lip1 を作成するには、プロファイルの関連付け FC インターフェイス ポリシー グループ ipg1、およびプロファイルとその関連するポリシーが適用このインターフェイスを指定します。

例：

```
apic1# configure
apic1(config)# leaf-interface-profile lip1
apic1(config-leaf-if-profile)# description 'test description lip1'
apic1(config-leaf-if-profile)# leaf-interface-group lig1
apic1(config-leaf-if-group)# description 'test description lig1'
apic1(config-leaf-if-group)# fc-policy-group ipg1
apic1(config-leaf-if-group)# interface fc 1/1-4
```

ステップ 6 リーフ プロファイルを作成し、そのリーフ プロファイルにリーフ インターフェイス プロファイルを割り当てて、そのプロファイルの適用先となるリーフ ID を割り当てます：

例：

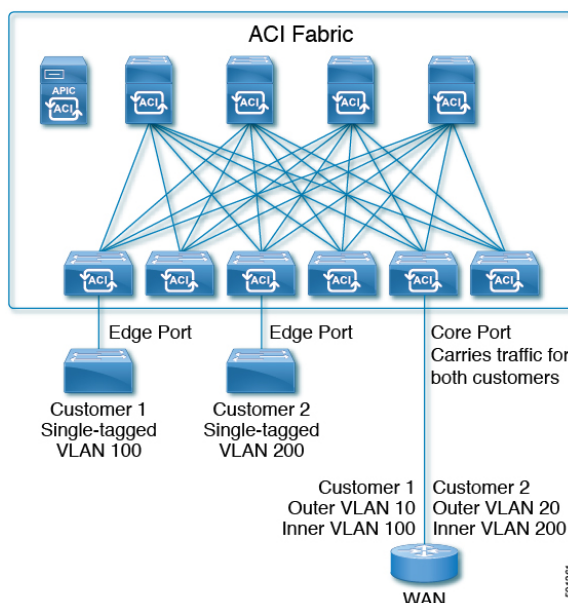
```
apic1(config)# leaf-profile lp103
apic1(config-leaf-profile)# leaf-interface-profile lip1
apic1(config-leaf-profile)# leaf-group range
apic1(config-leaf-group)# leaf 103
apic1(config-leaf-group)#
```

(注) リーフ インターフェイス プロファイル リーフに関連付けると、リーフのリロードが FC ポートとしてこれらのポートを起動する必要があります。

802.1Q トンネルの設定

ACI 802.1 q トンネルについて

図 6: ACI 802.1 q トンネル



Cisco ACI および Cisco APIC リリース 2.2(1x) 以降では、エッジ（トンネル）ポートで 802.1Q トンネルを設定して、QoS（QoS）の優先順位設定とともに、ファブリックのイーサネットフレームの point-to-multi-point トンネリングを有効にできます。**Dot1q トンネル** は、タグ付き、802.1Q タグ付き、802.1ad 二重タグ付きフレームを、ファブリックでそのまま送信します。各トンネルでは、単一の顧客からのトラフィックを伝送し、単一のブリッジドメインに関連付けられています。ACI 前面パネルポートは、**Dot1q トンネル** の一部である可能性があります。レイヤ 2 スイッチングは宛先 MAC（DMAC）に基づいて行われ、通常の MAC ラーニングはトンネルで行われます。エッジポート **Dot1q トンネル** は、第二世代以降のスイッチのモデル名の最後に「EX」がつく Cisco Nexus 9000 シリーズスイッチでサポートされます。

Cisco ACI および Cisco APIC リリース 2.3(x) では、同じコアポートで複数の 802.1 q トンネルを設定可能で、複数の顧客から二重タグ付きトラフィックを伝送し、これは 802.1 q 用に設定されたアクセスのカプセル化を各自識別できます。802.1Q トンネルでは、MAC アドレス学習を無効にすることもできます。エッジポートとコアポートの両方を、アクセスカプセル化が設定され、MAC アドレス学習が無効にされた 802.1Q トンネルに所属させることができます。**Dot1q トンネル** のエッジポートとコアポートの両方は、Cisco Nexus 9000 シリーズスイッチの第三世代の、スイッチモデル名の末尾に「FX」が付いている機種でサポートされます。

このドキュメントで使用する用語は、**Cisco Nexus 9000 シリーズ** のドキュメントとは異なっている場合があります。

表 11: 802.1Q トンネルの用語

ACI のドキュメント	Cisco Nexus 9000 シリーズのドキュメント
エッジ ポート	トンネル ポート
コア ポート	トランク ポート

次の注意事項および制約事項が適用されます:

- VTP、CDP、LACP、LLDP、および STP プロトコルのレイヤ 2 トンネリングは、次の制限付きでサポートされます。
 - リンク集約制御プロトコル (LACP) トンネリングは、個々のリーフ インターフェイスを使用する、ポイントツーポイントトンネルでのみ、予想通りに機能します。ポートチャネル (PC) または仮想ポート チャネル (vPC) ではサポートされません。
 - PC または vPC を持つ CDP および LLDP トンネリングは確定的ではありません。これは、トラフィックの宛先として選択するリンクによって異なります。
 - レイヤ 2 プロトコル トンネリングに VTP を使用するには、CDP をトンネル上で有効にする必要があります。
 - レイヤ 2 プロトコルのトンネリングが有効になっており、Dot1q トンネルのコア ポートにブリッジ ドメインが展開されている場合、STP は 802.1q トンネルブリッジ ドメインではサポートされません。
 - ACI リーフ スイッチは、トンネルブリッジ ドメインのエンドポイントで点滅し、ブリッジ ドメインでフラッドイングして STP TCN パケットに反応します。
 - 2 個上のインターフェイスを持つ CDP および LLDP トンネリングが、すべてのインターフェイスでパケットをフラッドイングします。
 - Cisco APIC リリース 2.3(x) 以降では、エッジ ポートからコア ポートにトンネリングしているレイヤ 2 プロトコル パケットの宛先 MAC アドレスは、01-00-0c-cd-cd-d0 に書き換えられ、コア ポートからエッジ ポートにトンネリングしているレイヤ 2 プロトコルパケットの宛先 MAC アドレスは、プロトコルに対して標準のデフォルト MAC アドレスに書き換えられます。
 - PC または VPC が **Dot1q トンネル** の唯一のインターフェイスであり、**削除**され再度設定される場合、**Dot1q トンネル** に対する PC/VPC の関連付けを削除してから再設定します。
 - Cisco APIC リリース 2.2(x) では、二重タグつきフレームのイーサタイプは 0x8100 の後に 0x9100 が続く必要があります。
- ただし、Cisco APIC リリース 2.3(x) 以降の場合、この制限は第三世代のスイッチモデル名の最後に「FX」がつく Cisco Nexus switches のエッジ ポートには適用されません。

- コア ポートについては、二重タグつきフレームのイーサタイプは、0x8100 の後に 0x8100 が続く必要があります。
- 複数のエッジ ポートおよびコア ポート（リーフ スイッチ上も）を **Dot1q トンネル** に含むことができます。
- エッジ ポートは 1 つのトンネルの一部にのみ属することが可能ですが、コア ポートは複数の Dot1q トンネルに属することができます。
- Cisco APIC リリース 2.3(x) 以降では、通常の EPG が 802.1 q で使用されるコア ポートに展開できます。
- L3Outs は **Dot1q トンネル** に有効になっているインターフェイスではサポートされていません。
- FEX は **Dot1q トンネル** のメンバとしてはサポートされていません。
- ブレークアウト ポートとして設定されているインターフェイスは、802.1 q をサポートしていません。
- インターフェイス レベルの統計情報は **Dot1q トンネル** のインターフェイスでサポートされていますが、トンネル レベルの統計情報はサポートされていません。

NX-OS スタイル CLI を使用した 802.1Q トンネルの設定



- (注) **Dot1q トンネル** に含まれるインターフェイスのポート、ポート チャネル、仮想ポート チャネルを使用できます。手順の詳細にはポートの設定が含まれます。エッジおよびコア ポート チャネルと仮想ポート チャネルを設定するコマンドについては、下の例を参照してください。

次の手順で、**Dot1q トンネル** を作成し、NX-OS スタイル CLI を使用してトンネルで使用するインターフェイスを設定します。



- (注) **Dot1q トンネル** には 2 個以上のインターフェイスを含める必要があります。手順を繰り返し（または 2 個のインターフェイスをまとめて設定）、**Dot1q トンネル** で使用する各インターフェイスをマークします。この例で、2 個のインターフェイスは単一の顧客で使用されているエッジ スイッチ ポートとして設定されます。

次の手順を使用して、設定を次の手順を使用して、NX-OS スタイル CLI を使用して **Dot1q トンネル** を設定します。

1. トンネルで使用するインターフェイスを最低 2 個設定します。
2. **Dot1q トンネル** を作成します。
3. トンネルとすべてのインターフェイスを関連付けます。

始める前に

Dot1q トンネル を使用するテナントを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーションモードに入ります。
ステップ 2	次の手順により 802.1Q で使用するための 2 個のインターフェイスを設定します。	
ステップ 3	leaf ID 例 : apic1(config)# leaf 101	Dot1q トンネル のインターフェイスが配置されるリーフを特定します。
ステップ 4	interface ethernet slot/port 例 : apic1(config-leaf)# interface ethernet 1/13-14	トンネルのポートとしてマークされるインターフェイスを特定します。
ステップ 5	switchport mode dot1q-tunnel {edgePort corePort} 例 : apic1(config-leaf-if)# switchport mode dot1q-tunnel edgePort apic1(config-leaf-if)# exit apic1(config-leaf)# exit apic1(config)# exit	802.1Q トンネルで使用するインターフェイスをマークして、設定モードをそのままにします。 この例では、エッジポートを使用するためにいくつかのインターフェイス設定を示します。トンネルに複数のインターフェイスを設定するには、手順 3 ~ 5 を繰り返します。
ステップ 6	次の手順で 802.1q トンネルを作成します。	
ステップ 7	leaf ID 例 : apic1(config)# leaf 101	インターフェイスが配置されているリーフに戻ります。
ステップ 8	interface ethernet slot/port 例 : apic1(config-leaf)# interface ethernet 1/13-14	トンネルに含まれるインターフェイスに戻ります。

	コマンドまたはアクション	目的
ステップ 9	switchport tenant <i>tenant-namedot1q-tunnel</i> <i>tunnel-name</i> 例 : <pre>apic1(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_edgetunnel apic1(config-leaf-if)# exit</pre>	トンネルにインターフェイスに関連付け、設定モードを終了します。
ステップ 10	トンネルとその他のインターフェイスを関連付けるには、ステップ 7～10 を繰り返します。	

例：NX-OS スタイル CLI でポートを使用する 802.1Q トンネルを設定する

この例では、2つのポートを Dot1q トンネルで使用されるエッジポートインターフェイスとしてマークし、さらに2つのポートをコアポートインターフェイスで使用されるものとしてマークし、トンネルを作成して、ポートをトンネルに関連付けます。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/13-14
apic1(config-leaf-if)# switchport mode dot1q-tunnel edgePort
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 1/10, 1/21
apic1(config-leaf-if)# switchport mode dot1q-tunnel corePort
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# tenant tenant64
apic1(config-tenant)# dot1q-tunnel vrf64_tunnel
apic1(config-tenant-tunnel)# l2protocol-tunnel cdp
apic1(config-tenant-tunnel)# l2protocol-tunnel lldp
apic1(config-tenant-tunnel)# access-encap 200
apic1(config-tenant-tunnel)# mac-learning disable
apic1(config-tenant-tunnel)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/13-14
apic1(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_tunnel
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 1/10, 1/21
apic1(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_tunnel
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

例：NX-OS スタイル CLI でポート チャネルを使用する 802.1Q トンネルを設定する

例：NX-OS スタイル CLI でポート チャネルを使用する 802.1Q トンネルを設定する

例では、このエッジポート 802.1q インターフェイスとして 2 つのポート チャネルにマークし、2 つ以上のポート チャネルをコア ポート 802.1q インターフェイスとしてマークして、Dotq トンネルを作成し、トンネルとポート チャネルを関連付けます。

```
apic1# configure
apic1(config)# tenant tenant64
apic1(config-tenant)# dot1q-tunnel vrf64_tunnel
apic1(config-tenant-tunnel)# l2protocol-tunnel cdp
apic1(config-tenant-tunnel)# l2protocol-tunnel lldp
apic1(config-tenant-tunnel)# access-encap 200
apic1(config-tenant-tunnel)# mac-learning disable
apic1(config-tenant-tunnel)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel pc1
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/2-3
apic1(config-leaf-if)# channel-group pc1
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel pc1
apic1(config-leaf-if)# switchport mode dot1q-tunnel edgePort
apic1(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_tunnel
apic1(config-tenant-tunnel)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 102
apic1(config-leaf)# interface port-channel pc2
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/4-5
apic1(config-leaf-if)# channel-group pc2
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel pc2
apic1(config-leaf-if)# switchport mode dot1q-tunnel corePort
apic1(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_tunnel
```

例：NX-OS スタイル CLI で仮想ポート チャネルを使用する 802.1Q トンネルを設定する

この例では、2 つの仮想ポート チャネル (VPC) を Dot1q トンネルのエッジポート 802.1Q インターフェイスとしてマークし、さらに 2 つの VPC をトンネルのためのコアポート インターフェイスとしてマークし、トンネルを作成して、仮想ポート チャネルをトンネルに関連付けています。

```
apic1# configure
apic1(config)# vpc domain explicit 1 leaf 101 102
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# interface vpc vpc1
apic1(config-vpc-if)# switchport mode dot1q-tunnel edgePort
apic1(config-vpc-if)# exit
apic1(config-vpc)# exit
apic1(config)# vpc domain explicit 1 leaf 103 104
apic1(config)# vpc context leaf 103 104
apic1(config-vpc)# interface vpc vpc2
apic1(config-vpc-if)# switchport mode dot1q-tunnel corePort
apic1(config-vpc-if)# exit
apic1(config-vpc)# exit
```

```

apicl(config)# tenant tenant64
apicl(config-tenant)# dot1q-tunnel vrf64_tunnel
apicl(config-tenant-tunnel)# l2protocol-tunnel cdp
apicl(config-tenant-tunnel)# l2protocol-tunnel lldp
apicl(config-tenant-tunnel)# access-encap 200
apicl(config-tenant-tunnel)# mac-learning disable
apicl(config-tenant-tunnel)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 103
apicl(config-leaf)# interface ethernet 1/6
apicl(config-leaf-if)# channel-group vpc1 vpc
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
apicl(config)# leaf 104
apicl(config-leaf)# interface ethernet 1/6
apicl(config-leaf-if)# channel-group vpc1 vpc
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
apicl(config-vpc)# interface vpc vpc1
apicl(config-vpc-if)# switchport tenant tenant64 dot1q-tunnel vrf64_tunnel
apicl(config-vpc-if)# exit

```

ダイナミック ブレークアウト ポートの設定

ダイナミック ブレークアウト ポートの設定

ブレークアウトケーブルは非常に短いリンクに適しており、コスト効率の良いラック内および隣接ラック間を接続する方法を提供します。

ブレークアウトでは、40 ギガビット (Gb) ポートを独立して 4 分割し、10Gb または 100Gb ポートを独立した状態で論理 25 Gb ポートに 4 分割できます。

ブレークアウトポートを設定する前に、次のケーブルのいずれかを使用して 40 Gb ポートを 4 つの 10 Gb ポートまたは 100 Gb ポートを 4 つの 25 Gb ポートに接続します。

- Cisco QSFP-4SFP10G
- Cisco QSFP-4SFP25G

40 Gb から 10 Gb までのダイナミック ブレークアウト機能は、次のスイッチのポートが面しているアクセスでサポートされています。

- N9K-C9332PQ
- N9K-C93180LC-EX
- N9K-C9336C-FX

100 Gb から 25 Gb までのブレークアウト機能は、次のスイッチのポートが面しているアクセスでサポートされています。

- N9K-C93180LC-EX
- N9K-C9336C-FX2

- N9K-C93180YC-FX

次の注意事項および制約事項を確認します。

- 一般に、ブレイクアウトおよびポートプロファイル（アップリンクからダウンリンクへ変更されたポート）は、同一ポートではサポートされません。

ただし Cisco APIC、リリース 3.2 から、ダイナミック ブレイクアウト（100 Gb および 40 Gb の両方）は N9K-C93180YC-FX スイッチのプロファイリングされた QSFP ポートでサポートされます。

- ファストリンク フェールオーバー ポリシーは、ダイナミック ブレイクアウト機能と同一ポートではサポートされていません。
- ブレイクアウトのサポートは、ポリシー モデルが使用されているその他のポート タイプと同じ方法で使用できます。
- ポートがダイナミック ブレイクアウトに対して有効になっている場合、親ポートのその他のポート（モニタリング ポリシー以外）は無効になります。
- ポートがダイナミック ブレイクアウトに対して有効になっている場合、親ポートのその他の EPG 展開が無効になります。
- ブレイクアウト サブポートは、ブレイクアウト ポリシー グループを使用してもこれ以上分割することはできません。

NX-OS スタイルの CLI を使用したダイナミック ブレイクアウト ポートの設定

ブレイクアウトポートを設定、設定を確認および NX-OS スタイル CLI を使用してサブポートで、EPG を設定するには、次の手順を使用します。

始める前に

- ACI ファブリックが設置され、APIC コントローラがオンラインになっており、APIC クラスタが形成されて正常に動作していること。
- 必要なファブリック インフラストラクチャ設定を作成できる APIC ファブリック管理者アカウントが使用可能であること。
- ターゲット リーフ スイッチが ACI ファブリックに登録され、使用可能であること。
- 40GE または 100GE リーフ スイッチ ポートは、ダウンリンク ポートに Cisco ブレイクアウト ケーブルを接続します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	コンフィギュレーションモードに入ります。
ステップ 2	leaf ID 例 : apicl(config)# leaf 101	ブレイクアウトポートが配置され、リーフ configuration mode(設定モード、コンフィギュレーションモード)を開始リーフ スイッチを選択します。
ステップ 3	interface ethernetlot/port 例 : apicl(config-leaf)# interface ethernet 1/16	40 ギガビット イーサネット (GE) ブレイクアウトポートとして有効にするインターフェイスを識別します。
ステップ 4	breakout10g-4x 25g-4x 例 : apicl(config-leaf-if)# breakout 10g-4x	ブレイク アウトを選択したインターフェイスを有効にします。 (注) ダイナミック ブレイク アウトポート機能は、スイッチのサポートを参照してください。 ダイナミック ブレイク アウト ポートの設定 (127 ページ) 。
ステップ 5	show run 例 : apicl(config-leaf-if)# show run # Command: show running-config leaf 101 # Time: Fri Dec 2 18:13:39 2016 leaf 101 interface ethernet 1/16 breakout 10g-4x apicl(config-leaf-if)# exit apicl(config-leaf)# exit	インターフェイスの実行コンフィギュレーションを表示することによって、設定を確認し、グローバル コンフィギュレーション モードに戻ります。
ステップ 6	tenant tenant-name 例 : apicl(config)# tenant tenant64	選択またはブレイクアウトポートで消費され、テナント configuration mode(設定モード、コンフィギュレーションモード)を開始するテナントを作成します。
ステップ 7	vrf context vrf-name 例 :	作成またはテナントに関連付けられている Virtual Routing and Forwarding (VRF) インスタンスを識別し、

	コマンドまたはアクション	目的
	<pre>apic1(config-tenant)# vrf context vrf64 apic1(config-tenant-vrf)# exit</pre>	configuration mode(設定モード、コンフィギュレーション モード) を終了します。
ステップ 8	bridge-domain <i>bridge-domain-name</i> 例 : <pre>apic1(config-tenant)# bridge-domain bd64</pre>	作成またはテナントに関連付けられているブリッジ ドメインを識別し、BD configuration mode(設定モード、コンフィギュレーション モード) を開始します。
ステップ 9	vrf member <i>vrf-name</i> 例 : <pre>apic1(config-tenant-bd)# vrf member vrf64 apic1(config-tenant-bd)# exit</pre>	ブリッジドメイン、VRF の関連付け、configuration mode(設定モード、コンフィギュレーション モード) を終了します。
ステップ 10	application <i>application-profile-name</i> 例 : <pre>apic1(config-tenant)# application app64</pre>	作成またはテナントと EPG に関連付けられているアプリケーションプロファイルを識別します。
ステップ 11	epg <i>epg-name</i> 例 : <pre>apic1(config-tenant)# epg epg64</pre>	作成または EPG を識別し、EPG configuration mode(設定モード、コンフィギュレーション モード) に入力します。
ステップ 12	bridge-domain member <i>bridge-domain-name</i> 例 : <pre>apic1(config-tenant-app-epg)# bridge-domain member bd64 apic1(config-tenant-app-epg)# exit apic1(config-tenant-app)# exit apic1(config-tenant)# exit</pre>	EPG をブリッジドメインに関連付け、グローバル設定モードをにに戻ります。 たとえば、必要に応じて、サブポートを設定コマンドを使用して、速度リーフ インターフェイス モードでサブポートを設定します。
ステップ 13	speed <i>interface-speed</i> 例 : <pre>apic1(config)# leaf 101 apic1(config-leaf)# interface ethernet 1/16/1 apic1(config-leaf-if)# speed 10G apic1(config-leaf-if)# exit</pre>	リーフ インターフェイス モードを開始し、[インターフェイスの速度を設定 configuration mode(設定モード、コンフィギュレーション モード) を終了します。
ステップ 14	show run 例 : <pre>apic1(config-leaf)# show run</pre>	サブポートを設定した後にリーフ configuration mode(設定モード、コンフィギュレーション モード) で次のコ

	コマンドまたはアクション	目的
		マンドを入力して、サブポートの詳細が表示されます。

サブポート 1/16/1、2/1/16、1/16/3 および 4/1/16 ブレイクアウトを有効になっているリーフ インターフェイス 1/16 で 101 上のポートを確認します。

例

この例では、ブレイクアウト ポートで設定します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/16
apic1(config-leaf-if)# breakout 10g-4x
```

この例では、サブインターフェイス ポートの EPG で設定します。

```
apic1(config)# tenant tenant64
apic1(config-tenant)# vrf context vrf64
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain bd64
apic1(config-tenant-bd)# vrf member vrf64
apic1(config-tenant-bd)# exit
apic1(config-tenant)# application app64
apic1(config-tenant-app)# epg epg64
apic1(config-tenant-app-epg)# bridge-domain member bd64
apic1(config-tenant-app-epg)# end
```

この例では、10 G に、ブレイクアウトの速度サブポートを設定します。

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/16/1
apic1(config-leaf-if)# speed 10G
apic1(config-leaf-if)# exit

apic1(config-leaf)# interface ethernet 1/16/2
apic1(config-leaf-if)# speed 10G
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/16/3
apic1(config-leaf-if)# speed 10G
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/16/4
apic1(config-leaf-if)# speed 10G
apic1(config-leaf-if)# exit
```

この例では、リーフ 101、インターフェイス 1/16 に接続されている、4 つのアシスタント的なポートを示します。

```
apic1#(config-leaf)# show run
# Command: show running-config leaf 101
# Time: Fri Dec 2 00:51:08 2016
leaf 101
  interface ethernet 1/16/1
    speed 10G
    negotiate auto
    link debounce time 100
  exit
  interface ethernet 1/16/2
    speed 10G
```

```

negotiate auto
link debounce time 100
exit
interface ethernet 1/16/3
speed 10G
negotiate auto
link debounce time 100
exit
interface ethernet 1/16/4
speed 10G
negotiate auto
link debounce time 100
exit
interface ethernet 1/16
breakout 10g-4x
exit
interface vfc 1/16

```

ポート プロファイルの設定

ポート プロファイルの設定

Cisco APIC リリース 3.1 (1) 以前、アップリンク ポートからダウンリンク ポート、あるいはダウンリンク ポートからアップリンク ポート（ポート プロファイル内）への変換は、Cisco ACI リーフ スイッチではサポートされていませんでした。Cisco APIC リリース 3.1 (1) から、アップリンクおよびダウンリンクの変換は、EX または FX およびそれ以降（たとえば、N9K-C9348GC-FXP）で終わる名前の Cisco Nexus 9000 シリーズ スイッチでサポートされています。変換後のダウンリンクに接続されている FEX もサポートされています。

この機能は次の Cisco スイッチでサポートされています。

- N9K-C9348GC-FXP
- N9K-C93180LC-EX および N9K-C93180YC-FX
- N9K-93180YC-EX、N9K-C93180YC-EX、N9K-C93180YC-EXU
- N9K-C93108TC-EX および N9K-C93108TC-FX
- N9K C9336C FX2（ダウンリンクからアップリンクへの変換のみサポート）

制約事項

FAST リンク フェールオーバー ポリシーとポート プロファイルは、同じポートではサポートされていません。ポート プロファイルが有効になっている場合、FAST リンク フェールオーバーを有効にすることはできません。その逆も同様です。

サポートされている TOR スイッチの最後の 2 つのアップリンク ポートは、ダウンリンク ポート（これらはアップリンク接続用に予約済み）に変換することはできません。

Cisco APIC リリース 3.2 までは、ポート プロファイルとブレイクアウト ポートは同じポートでサポートされていません。

Cisco APIC リリース 3.2 以降では、ダイナミック ブレイクアウト (100 Gb および 40 Gb の両方) が N9K-C93180YC-FX スイッチのプロファイリングされた QSFP ポートでサポートされます。ブレイクアウトおよびポートプロファイルでは、ポート 49-52 でアップリンクからダウンリンクへの変換と一緒にサポートされています。ブレイクアウト (**10 g 4 x** または **25 g 4 x** オプション) は、ダウンリンクのプロファイリングされたポートでサポートされています。

ガイドライン

アップリンクをダウンリンクに変換したり、ダウンリンクをアップリンクに変換したりする際は、次のガイドラインにご注意ください。

サブジェクト	ガイドライン
ポートプロファイルを使用したノードのデコミッション	デコミッションされたノードがポートプロファイル機能を展開している場合、ポート変換はノードのデコミッション後も削除されません。ポートをデフォルト状態に戻すには、デコミッション後に手動で設定を削除する必要があります。これを行うには、スイッチにログオンし、 setup-clean-config.sh スクリプトを実行して、実行されるまで待ちます。それから、 リロード コマンドを入力します。
FIPS	Cisco ACI ファブリックで Federal Information Processing Standards (FIPS) を有効または無効にしている場合、変更を有効にするためファブリック内のスイッチをそれぞれリロードする必要があります。FIPS 設定を変更した後に最初のリロードを発行すると、設定されているスケールプロファイル設定が失われます。スイッチは動作を継続しますが、デフォルトのスケールプロファイルを使用します。FIPS 設定が変更されていない場合は、この問題は後続のリロードでは発生しません。 FIPS は Cisco NX-OS リリース 13.1(1) またはそれ以降でサポートされています。 FIPS をサポートしているリリースから FIPS をサポートしていないリリースにファームウェアをダウングレードする必要がある場合、最初に Cisco ACI ファブリックで FIPS を無効にして、FIPS 設定の変更のためファブリック内のすべてのスイッチをリロードする必要があります。

サブジェクト	ガイドライン
最大アップリンク ポートの制限	最大アップリンク ポートの制限に達し、ポート 25 および 27 がアップリンクからダウンリンクへ返還されるとき、Cisco 93180LC EX スイッチのアップリンクに戻ります。 Cisco 93180LC EX スイッチでは、ポート 25 および 27 はネイティブのアップリンク ポートです。ポート プロファイルを使用してポート 25 および 27 をダウンリンク ポートに変換する場合は、ポート 29、30、31、32 は、4 つのネイティブ アップリンク ポートとしても使用できます。変換可能なポート数のしきい値のため（最大 12 ポート）、8 個以上のダウンリンク ポートをアップリンク ポートに変換できます。たとえば、ポート 1、3、5、7、9、13、15、17 はアップリンク ポートに変換され、ポート 29、30、31、32 ポートは 4 つのネイティブ アップリンク ポートとなります（Cisco 93180LC EX スイッチで最大のアップリンク ポートの制限）。 スイッチがこの状態でポート プロファイル設定がポート 25 および 27 で削除される場合、ポート 25 および 27 はアップリンク ポートへ再度変換されますが、前述したようにスイッチにはすでに 12 個のアップリンク ポートがあります。ポート 25 および 27 をアップリンク ポートとして適用するため、ポート範囲 1、3、5、7、9、13、15、17 からランダムで 2 個のポートがアップリンクへの変換を拒否されます。この状況はユーザにより制御することはできません。 そのため、リーフ ノードをリロードする前にすべての障害を消去し、ポートタイプに関する予期しない問題を回避することが必須です。ポート プロファイルの障害を消去せずにノードをリロードすると、特に制限超過に関する障害の場合、ポートは予想される動作状態になることに注意する必要があります。

ブレイクアウト制限

スイッチ	リリース	制限事項
N9K-C9332PQ	Cisco APIC 2.2 (1n) 以降	4X10Gb ポートへの 40Gb ダイナミック ブレイク アウトがサポートされています。 - ポート 13 および 14 はブレイクアウトをサポートしていません。 - ポート プロファイルおよびブレイクアウトは、同じポートでサポートされていません。

スイッチ	リリース	制限事項
N9K-C93180LC-EX	Cisco APIC 3.1(1i) 以降	• 40 Gb と 100 Gb のダイナミック ブレークアウトは、ポート 1 ~ 24 の奇数ポート上でサポートされます。 • 上位ポート（奇数ポート）ブレークアウトされると、下部ポート（偶数ポート）はエラーが無効になります。 • ポート プロファイルおよびブレークアウトは、同じポートでサポートされていません。
N9K-C9336C-FX2	Cisco APIC 3.1(2m) 以降	• ポート 1 ~ 30 では、40 Gb と 100 Gb のダイナミック ブレークがサポートされています。 • ポート プロファイルおよびブレークアウトは、同じポートでサポートされていません。

スイッチ	リリース	制限事項
N9K-C93180YC-FX	Cisco APIC リリース 3.2(1x) 以降	• 40 Gb と 100 Gb のダイナミック ブレークは、52、上にあるときにプロファイリング QSFP ポートがポート 49 でサポートされます。ダイナミック ブレークアウトを使用するには、次の手順を実行します。 • ポート 49~52 を前面パネルポート（ダウンリンク）に変換します。 • 次の方法のいずれかを使用して、ポート プロファイルのリロードを実行します。 • APIC GUI で、[ファブリック]>[インベントリ]>[ポッド]>[リーフ]に移動し、[シャーシ]をクリックしてから [リロード] を選択します。 • NX-OS スタイル CLI で、setup-clean-config.sh -k スクリプトを入力し、実行を待機し、reload コマンドを入力します。 • プロファイルされたポート 49 - 52 のブレーク アウトを適用します。 • ポート 53 および 54 では、ポート プロファイルまたはブレークアウトをサポートしていません。

ポート プロファイルの設定のまとめ

次の表では、アップリンクからダウンリンク、ダウンリンクからアップリンクへのポートプロファイルの変換をサポートしているスイッチで、サポートされているアップリンクおよびダウンリンクをまとめています。

スイッチ モデル	デフォルト リンク	最大アップリンク (ファブリック ポート)	最大ダウンリンク (サーバのポート)	サポートされているリリース
N9K-C9348GC-FXP	48 x 100 M/1 G BASE-T ダウンリンク 4 x 10/25 Gbps SFP28 ダウンリンク 2 x 40/100 Gbps QSFP28 アップリンク	48 x 100 M/1 G BASE-T ダウンリンク 4 x 10/25 Gbps SFP28 アップリンク 2 x 40/100 Gbps QSFP28 アップリンク	デフォルトと同じ	3.1(i)
N9K-C93180LC-EX	24 x 40 Gbps QSFP 28 ダウンリンク 6 x 40/100-Gbps QSFP 28 アップリンク または 12 x 100 Gbps QSFP 28 ダウンリンク 6 x 40/100-Gbps QSFP 28 アップリンク	12 x 40 Gbps QSFP 28 ダウンリンク 12 x 40/100-Gbps QSFP 28 アップリンク または 6 x 100 Gbps QSFP 28 ダウンリンク 12 x 40/100-Gbps QSFP 28 アップリンク	4 x 40 Gbps QSFP 28 ダウンリンク 2 x 40/100-Gbps QSFP 28 ダウンリンク 4 x 40/100-Gbps アップリンク (100 Gbps) または 12 x 100 Gbps QSFP 28 ダウンリンク 2 x 40/100-Gbps QSFP 28 ダウンリンク 4 x 40/100-Gbps アップリンク (100 Gbps)	3.1(i)
N9K-C93180YC-EX N9K-C93180YC-FX	48 x 10/25 Gbps ファ イバ ダウンリンク 6 x 40/100 Gbps QSFP28 アップリンク	48 x 10/25 Gbps ファ イバ ダウンリンク 6 x 40/100 Gbps QSFP28 アップリンク	48 x 10/25 Gbps ファイバ ダウンリ ンク 4 x 40/100 Gbps QSFP28 ダウンリ ンク 2 x 40/100 Gbps QSFP28 アップリン ク	3.1(i)

スイッチ モデル	デフォルト リンク	最大アップリンク (ファブリック ポート)	最大ダウンリンク (サーバのポート)	サポートされているリリース
N9K-C93108TC-EX N9K-C93108TC-FX	48 x 10GBASE T ダウンリンク 6 x 40/100 Gbps QSFP28 アップリンク	デフォルトと同じ	48 x 10/25 Gbps ファイバ ダウンリンク 4 x 40/100 Gbps QSFP28 ダウンリンク 2 x 40/100 Gbps QSFP28 アップリンク	3.1(1i)
N9K-C9336C-FX2	30 x 40/100 Gbps QSFP28 ダウンリンク 6 x 40/100 Gbps QSFP28 アップリンク	18 x 40/100 Gbps QSFP28 ダウンリンク 18 x 40/100 Gbps QSFP28 アップリンク	デフォルトと同じ	3.1(2m)

NX-OS スタイル CLI を使用したポート プロファイルの設定

NX-OS スタイルの CLI を使用したポート プロファイルの設定をするには、次の手順を実行します。

始める前に

- ACI ファブリックが設置され、APIC コントローラがオンラインになっており、APIC クラスタが形成されて正常に動作していること。
- 必要なファブリック インフラストラクチャ設定を作成または変更できる APIC ファブリック管理者アカウントが使用可能であること。
- ターゲット リーフ スイッチが ACI ファブリックに登録され、使用可能であること。

手順

ステップ 1 configure

グローバル コンフィギュレーション モードを開始します。

例：

```
apicl# configure
```

ステップ 2 leaf node-id

設定するリーフまたはリーフ スイッチを指定します。

例：

```
apicl(config)# leaf 102
```

ステップ 3 interface type

設定するインターフェイスを指定します。インターフェイス タイプと ID を指定できます。イーサネット ポートの場合は、`ethernet slot / port` を使用します。

例：

```
apicl(config-leaf)# interface ethernet 1/2
```

ステップ 4 port-direction {uplink | downlink}

ポートの方向を決定するか変更します。この例ではダウンリンクにポートを設定します。

(注) N9K-C9336C-FX スイッチでは、アップリンクからダウンリンクへの変更はサポートされていません。

例：

```
apicl(config-leaf-if)# port-direction downlink
```

ステップ 5 ポートがあるリーフ スイッチにログインし、`setup-clean-config.sh -k` コマンドを入力してから `reload` コマンドを入力します。

NX-OS スタイル CLI を使用したポート プロファイルの設定と変換の確認

`show interface brief` CLI コマンドを使用して、ポートの設定と変換を確認することができます。



(注) ポート プロファイルは、Cisco N9K-C93180LC EX スイッチのトップ ポートにのみ展開されます。たとえば、1、3、5、7、9、11、13、15、17、19、21、および23となります。ポート プロファイルを使用してトップ ポートを変換すると、ボトム ポートはハードウェア的に無効になります。たとえば、ポート プロファイルを使用して Eth 1/1 を変換すると、Eth 1/2 はハードウェア的に無効になります。

手順

ステップ 1 この例では、アップリンク ポートをダウンリンク ポートに変換する場合の出力を示しています。アップリンク ポートをダウンリンク ポートに変換変換する前に、この例での出力が表示

されます。**routed** というキーワードは、ポートがアップリンク ポートであることを示しています。

例：

```
switch# show interface brief
<snip>
Eth1/49          --      eth  routed  down   sfp-missing          100G (D)  --
Eth1/50          --      eth  routed  down   sfp-missing          100G (D)  --
<snip>
```

ステップ2 ポートプロファイルを設定して、スイッチのリロード、後に、例では、出力が表示されます。キーワード **トランク** ダウンリンク ポートとしてポートを示します。

例：

```
switch# show interface brief
<snip>
Eth1/49          0        eth  trunk   down   sfp-missing          100G (D)  --
Eth1/50          0        eth  trunk   down   sfp-missing          100G (D)  --
<snip>
```

仮想スイッチ上のマイクロセグメンテーション

仮想スイッチでのマイクロセグメンテーションの設定

Cisco Application Centric Infrastructure (ACI) マイクロセグメンテーションは、さまざまなネットワークベースまたは仮想マシン (VM) ベース属性に基づき、エンドポイント グループ (EPG) と呼ばれるロジカル セキュリティ ゾーンにエンドポイントを自動的に割り当てることができます。このセクションでは、仮想スイッチのマイクロセグメント (uSeg) EPGを設定する方法について説明します。

Cisco ACI マイクロセグメンテーションでは、次に接続されている仮想エンドポイントのサポートを提供します。

- VMware vSphere Distributed Switch (vDS)
- Cisco Application Virtual Switch (AVS)
- Microsoft vSwitch

マイクロセグメンテーションと Cisco ACI の動作、前提条件、ガイドライン、およびシナリオについての詳細は「[Cisco ACI 仮想化ガイド](#)」を参照してください。

NX-OS スタイル CLI を使用した Cisco ACI でのマイクロセグメンテーションの設定

ここでは、アプリケーション EPG 内で VM ベースの属性を使用して Cisco ACI for Cisco ACI Virtual Edge、Cisco AVS、VMware VDS、または Microsoft vSwitch でマイクロセグメンテーションを設定する方法について説明します。

手順

ステップ 1 CLI で、コンフィギュレーション モードに入ります。

例：

```
apic1# configure
apic1(config)#
```

ステップ 2 USeg EPG を作成します。

例：

この例は、アプリケーション EPG のためのものです。

(注) 次の例のマイクロセグメンテーションを許可するコマンドが VMware VDS にのみ必要です。

```
apic1(config)# tenant cli-ten1
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-baseEPG1
apic1(config-tenant-app-epg)# bridge-domain member cli-bd1
apic1(config-tenant-app-epg)# vmware-domain member cli-vmml allow-micro-segmentation
```

例：

(オプション) この例の設定は、uSeg EPG の EPG の優先順位と一致します。：

```
apic1(config)# tenant Coke
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented
apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1
apic1(config-tenant-app-uepg)# match-precedence 10
```

例：

この例では、属性 VM 名に基づいてフィルタを使用します。

```
apic1(config)# tenant cli-ten1
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented
apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1
apic1(config-tenant-app-uepg)# attribute-logical-expression 'vm-name contains <cos1>'
```

例：

この例では、IP アドレスに基づいてフィルタを使用します。

```
apic1(config)# tenant cli-ten1
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented
apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1
apic1(config-tenant-app-uepg)# attribute-logical-expression 'ip equals <FF:FF:FF:FF:FF:FF>'
```

例：

この例では、MAC アドレスに基づいてフィルタを使用します。

```
apic1(config)# tenant cli-ten1
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented
apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1
apic1(config-tenant-app-uepg)# attribute-logical-expression 'mac equals
<FF-FF-FF-FF-FF-FF>'
```

例：

この例では、演算子 AND を使用してすべての属性を一致させるか、演算子 OR を使用してすべての属性を一致させます。

```
apic1(config)# tenant cli-ten1
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented
apic1(config-tenant-app-uepg)# attribute-logical-expression 'hv equals host-123 OR
(guest-os equals "Ubuntu Linux (64-bit)" AND domain contains fex)'
```

ステップ 3 (Cisco ACI Virtual Edge のみ) : uSeg EPG を Cisco ACI Virtual Edge VMM ドメインに接続し、スイッチングおよびカプセル化モードを指定します。

例：

```
vmware-domain member AVE-CISCO
switching-mode AVE
encap-mode vxlan
exit
```

ステップ 4 USeg EPG の作成を確認します。

例：

次の例は、VM 名属性フィルタを持つ uSeg EPG のためのものです。

```
apic1(config-tenant-app-uepg)# show running-config
# Command: show running-config tenant cli-ten1 application cli-a1 epg cli-uepg1 type
micro-segmented # Time: Thu Oct 8 11:54:32 2015
tenant cli-ten1
application cli-a1
epg cli-uepg1 type micro-segmented
bridge-domain cli-bd1
attribute-logical-expression 'vm-name contains cos1 force'
{vmware-domain | microsoft-domain} member cli-vmm1
exit
exit
exit
```

ベア メタル上のマイクロセグメンテーションの設定

ベア メタルでのネットワークベースの属性によるマイクロセグメンテーションの使用

Cisco APIC を使用して Cisco ACI でのマイクロセグメンテーションを設定し、ネットワークベースの属性、MAC アドレス、または 1 つ以上の IP アドレスを使用した新しい属性ベースの EPG を作成できます。ネットワークベースの属性を使用して Cisco ACI でのマイクロセグメンテーションを設定し、単一のベース EPG または複数の EPG 内で VM または物理エンドポイントを分離できます。

IP ベースの属性の使用

IP ベースのフィルタを使用して、単一のマイクロセグメントで単一 IP アドレス、サブネット、または多様な非連続 IP アドレスを分離できます。ファイアウォールの使用と同様に、セキュリティ ゾーンを作成するための迅速かつ簡単な方法として、IP アドレスに基づいて物理エンドポイントを分離できます。

MAC ベースの属性の使用

MAC ベースのフィルタを使用して、単一 MAC アドレスまたは複数の MAC アドレスを分離できます。不適切なトラフィックをネットワークに送信するサーバがある場合はこの方法を推奨します。MAC ベースのフィルタを使用してマイクロセグメントを作成することで、このサーバを分離できます。

NX-OSスタイルのCLIを使用したベアメタル環境でのネットワークベースのマイクロセグメント EPG の設定

ここでは、ベアメタル環境のベース EPG 内で、ネットワークベースの属性（IP アドレスまたは MAC アドレス）を使用して Cisco ACI でマイクロセグメンテーションを設定する方法について説明します。

手順

	コマンドまたはアクション	目的
ステップ 1	CLI で、コンフィギュレーション モードに入ります。 例： apic1# configure apic1(config)#	
ステップ 2	マイクロセグメントを作成します。	

	コマンドまたはアクション	目的
	例 : この例では、IP アドレスに基づいてフィルタを使用します。 <pre> apicl(config)# tenant cli-ten1 apicl(config-tenant)# application cli-a1 apicl(config-tenant-app)# epg cli-uepg1 type micro-segmented apicl(config-tenant-app-uepg)# bridge-domain member cli-bd1 apicl(config-tenant-app-uepg)# attribute cli-upg-att match ip <X.X.X.X> #Schemes to express the ip A.B.C.D IP Address A.B.C.D/LEN IP Address and mask </pre> 例 : この例では、MAC アドレスに基づいてフィルタを使用します。 <pre> apicl(config)# tenant cli-ten1 apicl(config-tenant)# application cli-a1 apicl(config-tenant-app)# epg cli-uepg1 type micro-segmented apicl(config-tenant-app-uepg)# bridge-domain member cli-bd1 apicl(config-tenant-app-uepg)# attribute cli-upg-att match mac <FF-FF-FF-FF-FF-FF> #Schemes to express the mac E.E.E MAC address (Option 1) EE-EE-EE-EE-EE-EE MAC address (Option 2) EE:EE:EE:EE:EE:EE MAC address (Option 3) EEEE.EEEE.EEEE MAC address (Option 4) </pre> 例 : この例では、MAC アドレスに基づいてフィルタを使用し、この uSeg EPG のすべてのメンバー間に EPG 間分離を適用します。 <pre> apicl(config)# tenant cli-ten1 apicl(config-tenant)# application cli-a1 apicl(config-tenant-app)# epg cli-uepg1 type micro-segmented apicl(config-tenant-app-uepg)# isolation enforced apicl(config-tenant-app-uepg)# bridge-domain member cli-bd1 apicl(config-tenant-app-uepg)# attribute cli-upg-att match mac </pre>	

	コマンドまたはアクション	目的
	<pre><FF-FF-FF-FF-FF-FF> #Schemes to express the mac E.E.E MAC address (Option 1) EE-EE-EE-EE-EE-EE MAC address (Option 2) EE:EE:EE:EE:EE:EE MAC address (Option 3) EEEE.EEEE.EEEE MAC address (Option 4)</pre>	
ステップ 3	EPG を導入します。 例 : この例では、EPG を導入してリーフを指定します。 <pre>apic1(config)# leaf 101 apic1(config-leaf)# deploy-epg tenant cli-ten1 application cli-a1 epg cli-uepg1 type micro-segmented</pre>	
ステップ 4	マイクロセグメントの作成を確認します。 例 : <pre>apic1(config-tenant-app-uepg)# show running-config # Command: show running-config tenant cli-ten1 application cli-app1 epg cli-uepg1 type micro-segmented # Time: Thu Oct 8 11:54:32 2015 tenant cli-ten1 application cli-app1 epg cli-esx1bu type micro-segmented bridge-domain cli-bd1 attribute cli-uepg-att match mac 00:11:22:33:44:55 exit exit exit</pre>	

レイヤ 2 IGMP スヌープ マルチキャストの設定

Cisco APIC と IGMP スヌーピングについて

IGMP スヌーピングは、Internet Group Management Protocol (IGMP) ネットワーク トラフィックをリスニングするプロセスです。機能は、ホストとルータおよびフィルタ マルチキャスト リンク、する必要はありませんがどのポートが特定のマルチキャストトラフィックを受信を制御するための間での IGMP 対話リスンするようにネットワーク スイッチを使用できます。

Cisco APIC は、N9000 スタンドアロンなどの従来のスイッチに含まれるフル IGMP スヌーピング機能のサポートを提供します。

- ポリシー ベース IGMP スヌーピングごとの設定ブリッジ ドメイン

APICを使用することを有効にする、無効にすると、またはブリッジドメイン単位でIGMP スヌーピングのプロパティをカスタマイズポリシーを設定することができます。1つまたは複数のブリッジ ドメインへのポリシーを適用することができます。

- 静的ポート グループの実装

IGMP スタティック ポートのグループ化を使用すると、事前プロビジョニング、すでに静的に - に割り当てられたポートを受信し、IGMP を処理するスイッチ ポートとしてアプリケーション EPG、マルチキャスト トラフィック。この事前のプロビジョニングには、通常は IGMP スヌーピング スタック ポートを動的に学習するときに参加遅延ができなくなります。

静的ポートでのみスタティック グループ メンバーシップを事前にプロビジョニングされたいことができます(とも呼ばれる、 スタティック バインディング ポート)アプリケーション EPG に割り当てられます。

- アプリケーション Epg のアクセス グループの設定

「アクセス-グループ」ができるストリームを制御するために使用任意ポート背後に参加します。

実際に所属するするポートの設定を適用できることを確認するには、アプリケーション EPG に静的に割り当てられているインターフェイスでアクセス グループ設定を適用できる EPG。

ルート マップ ベースのアクセス グループのみが許可されます。



(注) 使用することができます **vzAny**、VRF 内のすべての Epg の IGMP スヌーピングなどのプロトコルをイネーブルにします。詳細については **vzAny** を参照してください [VRF で通信ルールを自動的にすべての Epg に適用する vzAny を使用して](#)。

使用する **vzAny** に移動 テナント > テナント名 > ネットワー キング > Vrf > *vrf*名 > VRF の EPG コレクション。

静的ポート グループの IGMP スヌーピングを有効にする

IGMP 静的ポートのグループ化により以前アプリケーション EPG に静的に割り当てられた事前プロビジョニングを有効にして、スイッチ ポートが IGMP マルチキャスト トラフィックを受信および処理できます。この事前プロビジョニングは、通常 IGMP スヌーピング スタックがポートを動的に学習するときに発生する参加遅延を防止します。

静的グループ メンバーシップは、アプリケーション EPG に割り当てられている静的ポートでのみ事前プロビジョニングできます。

APIC GUI、CLI、および REST API インターフェイスを通じて、静的グループ メンバーシップを設定できます。

NX-OS スタイル CLI を使用した IGMP スヌーピング ポリシーの設定とブリッジ ドメインへの割り当て

始める前に

- IGMP スヌーピングのポリシーを消費するテナントを作成します。
- IGMP スヌーピング ポリシーを接続するテナントのブリッジ ドメインを作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	デフォルト値に基づいてスヌーピング ポリシーを作成します。 例 : <pre> apic1(config-tenant)# template ip igmp snooping policy cookieCut1 apic1(config-tenant-template-ip-igmp-snooping)# show run all </pre> <pre> # Command: show running -config all tenant foo template ip igmp snooping policy cookieCut1 # Time: Thu Oct 13 18:26:03 2016 tenant t_10 template ip igmp snooping policy cookieCut1 ip igmp snooping no ip igmp snooping fast-leave ip igmp snooping last-member-query-interval 1 no ip igmp snooping querier ip igmp snooping query-interval 125 ip igmp snooping query-max-response-time 10 ip igmp snooping stgrtup-query-count 2 ip igmp snooping startup-query-interval 31 no description exit exit apic1(config-tenant-template-ip-igmp-snooping)# </pre>	例の NX-OS スタイル CLI シーケンス : • デフォルト値を持つ cookieCut1 という名前の IGMP スヌーピング ポリシーを作成します。 • ポリシー cookieCut1 のデフォルト IGMP スヌーピングの値が表示されます。
ステップ 2	必要に応じてスヌーピング ポリシーを変更します。 例 : <pre> apic1(config-tenant-template-ip-igmp-snooping)# </pre>	例の NX-OS スタイル CLI シーケンス : • cookieCut1 という名前の IGMP スヌーピング ポリシーのクエリ間隔値のカスタム値を指定します。

	コマンドまたはアクション	目的
	<pre> ip igmp snooping query-interval 300 apic1(config-tenant-template-ip-igmp-snooping)# show run all # Command: show running -config all tenant foo template ip igmp snooping policy cookieCut1 #Time: Thu Oct 13 18:26:03 2016 tenant foo template ip igmp snooping policy cookieCut1 ip igmp snooping no ip igmp snooping fast-leave ip igmp snooping last-member-query-interval 1 no ip igmp snooping querier ip igmp snooping query-interval 300 ip igmp snooping query-max-response-time 10 ip igmp snooping stgrtup-query-count 2 ip igmp snooping startup-query-interval 31 no description exit exit apic1(config-tenant-template-ip-igmp-snooping)# exit apic1(config--tenant)# </pre>	ポリシー cookieCut1 の変更された IGMP スヌーピング値を確認します。
ステップ 3	ブリッジ ドメインにポリシーを割り当てます。 例 : <pre> apic1(config-tenant)# int bridge-domain bd3 apic1(config-tenant-interface)# ip igmp snooping policy cookieCut1 </pre>	例の NX-OS スタイル CLI シーケンス : - ブリッジ ドメインの BD3 に移動します。IGMP スヌーピングポリシーのクエリ間隔値は cookieCut1 という名前です。 - ポリシー cookieCut1 の変更された IGMP スヌーピングの値を持つ IGMP スヌーピングのポリシーを割り当てます。

次のタスク

複数のブリッジ ドメインに IGMP スヌーピングのポリシーを割り当てることができます。

NX-OS スタイル CLI によりスタティック ポートで IGMP スヌーピングおよびマルチキャストの有効化

EPG に静的に割り当てられたポートで IGMP スヌーピングおよびマルチキャストをイネーブルにできます。それらのポートで有効な IGMP スヌーピングおよびマルチキャストトラフィック

へのアクセスを許可または拒否するアクセスユーザーのグループを作成および割り当てることができます。

このタスクで説明されている手順には、次のエンティティの事前設定を前提とします。

- テナント : tenant_A
- アプリケーション : application_A
- EPG : epg_A
- ブリッジ ドメイン : bridge_domain_A
- vrf : vrf_A -- a member of bridge_domain_A
- VLAN ドメイン : vd_A (300 ~ 310 の範囲で設定される)
- リーフ スイッチ : 101 およびインターフェイス 1/10

スイッチ 101 のターゲット インターフェイス 1/10 が VLAN 305 に関連付けられており、tenant_A、application_A、epg_A に静的にリンクされています。

- リーフ スイッチ : 101 およびインターフェイス 1/11

スイッチ 101 のターゲット インターフェイス 1/11 が VLAN 309 に関連付けられており、tenant_A、application_A、epg_A に静的にリンクされています。

始める前に

EPG に IGMP スヌーピングおよびマルチキャストを有効にする前に、次のタスクを実行します。

- この機能を有効にして静的に EPG に割り当てるインターフェイスを特定する



(注) スタティック ポートの割り当てに関する詳細は、「Cisco APIC レイヤ 3 設定ガイド」の「NX-OS スタイル CLI を使用した APIC で特定のポートの EPG を展開する」を参照してください。

- IGMP スヌーピング マルチキャスト トラフィックの受信者の IP アドレスを特定します。

手順

	コマンドまたはアクション	目的
ステップ 1	ターゲット インターフェイスで IGMP スヌーピングおよびレイヤ 2 マルチキャストリングを有効にします 例 :	例のシーケンスでは次を有効にします。 • 静的にリンクされているターゲット インターフェイス 1/10 の IGMP スヌーピング、そしてマルチキャスト

	コマンドまたはアクション	目的
	<pre> apicl# conf t apicl(config)# tenant tenant_A apicl(config-tenant)# application application_A apicl(config-tenant-app)# epg epg_A apicl(config-tenant-app-epg)# ip igmp snooping static-group 225.1.1.1 leaf 101 interface ethernet 1/10 vlan 305 apicl(config-tenant-app-epg)# end apicl# conf t apicl(config)# tenant tenant_A; application application_A; epg epg_A apicl(config-tenant-app-epg)# ip igmp snooping static-group 227.1.1.1 leaf 101 interface ethernet 1/11 vlan 309 apicl(config-tenant-app-epg)# exit apicl(config-tenant-app)# exit </pre>	IP アドレス、225.1.1.1 に関連付けます • 静的にリンクされているターゲット インターフェイス 1/11 の IGMP スヌーピング、そしてマルチキャスト IP アドレス、227.1.1.1 に関連付けます

IGMP スヌープ アクセス グループの有効化

「アクセス-グループ」ができるストリームを制御するために使用任意ポート背後に参加します。

実際に所属するするポートの設定を適用できることを確認するには、アプリケーション EPG に静的に割り当てられているインターフェイスでアクセス グループ設定を適用できる EPG。

ルート マップ ベースのアクセス グループのみが許可されます。

APIC GUI、CLI、および REST API インターフェイスを通じて、IGMP スヌープ アクセス グループを設定できます。

NX-OS スタイル CLI を使用した IGMP スヌーピングおよびマルチキャスト グループへのアクセスの有効化

EPG に静的に割り当てられたポートで IGMP スヌーピングおよびマルチキャストを有効にした後、それらのポートで有効な IGMP スヌーピングおよびマルチキャストトラフィックへのアクセスを許可または拒否するユーザーのアクセス グループを作成および割り当てできます。

このタスクで説明されている手順には、次のエンティティの事前設定を前提とします。

- テナント : tenant_A
- アプリケーション : application_A
- EPG : epg_A
- ブリッジ ドメイン : bridge_domain_A
- vrf : vrf_A -- a member of bridge_domain_A
- VLAN ドメイン : vd_A (300 ~ 310 の範囲で設定される)

- リーフ スイッチ : 101 およびインターフェイス 1/10

スイッチ 101 のターゲット インターフェイス 1/10 が VLAN 305 に関連付けられており、enant_A、application_A、epg_A に静的にリンクされています。

- リーフ スイッチ : 101 およびインターフェイス 1/11

スイッチ 101 のターゲット インターフェイス 1/11 が VLAN 309 に関連付けられており、enant_A、application_A、epg_A に静的にリンクされています。



(注) スタティック ポートの割り当てに関する詳細は、「Cisco APIC レイヤ 2 設定ガイド」の「NX-OS スタイル CLI を使用した APIC で特定のポートの EPG を展開する」を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	route-map 「アクセス グループ」 を定義します。 例 : <pre> apic1# conf t apic1(config)# tenant tenant_A; application application_A; epg epg_A apic1(config-tenant)# route-map fooBroker permit apic1(config-tenant-rtmap)# match ip multicast group 225.1.1.1/24 apic1(config-tenant-rtmap)# exit apic1(config-tenant)# route-map fooBroker deny apic1(config-tenant-rtmap)# match ip multicast group 227.1.1.1/24 apic1(config-tenant-rtmap)# exit </pre>	例のシーケンスを設定します。 - マルチキャスト グループ 225.1.1.1/24 にリンクされる Route-map-access グループ 「foobroker」 のアクセスが許可されています。 - マルチキャスト グループ 225.1.1.1/24 にリンクされる Route-map-access グループ 「foobroker」 のアクセスが拒否されています。
ステップ 2	ルート マップ設定を確認します。 例 : <pre> apic1(config-tenant)# show running-config tenant test route-map fooBroker # Command: show running-config tenant test route-map fooBroker # Time: Mon Aug 29 14:34:30 2016 tenant test route-map fooBroker permit 10 match ip multicast group 225.1.1.1/24 exit route-map fooBroker deny 20 match ip multicast group 227.1.1.1/24 exit exit </pre>	

	コマンドまたはアクション	目的
ステップ 3	アクセス グループ接続パスを指定します。 例 : <pre> apicl(config-tenant)# application application_A apicl(config-tenant-app)# epg epg_A apicl(config-tenant-app-epg)# ip igmp snooping access-group route-map fooBroker leaf 101 interface ethernet 1/10 vlan 305 apicl(config-tenant-app-epg)# ip igmp snooping access-group route-map newBroker leaf 101 interface ethernet 1/10 vlan 305 </pre>	例のシーケンスを設定します。 - リーフ スイッチ 101、インターフェイス 1/10、VLAN 305 で接続されている Route-map-access グループ「foobroker」。 - リーフ スイッチ 101、インターフェイス 1/10、VLAN 305 で接続されている Route-map-access グループ「newbroker」。
ステップ 4	アクセスグループ接続を確認します。 例 : <pre> apicl(config-tenant-app-epg)# show run # Command: show running-config tenant tenant_A application application_A epg epg_A # Time: Mon Aug 29 14:43:02 2016 tenant tenant_A application application_A epg epg_A bridge-domain member bridge_domain_A ip igmp snooping access-group route-map fooBroker leaf 101 interface ethernet 1/10 vlan 305 ip igmp snooping access-group route-map fooBroker leaf 101 interface ethernet 1/11 vlan 309 ip igmp snooping access-group route-map newBroker leaf 101 interface ethernet 1/10 vlan 305 ip igmp snooping static-group 225.1.1.1 leaf 101 interface ethernet 1/10 vlan 305 ip igmp snooping static-group 225.1.1.1 leaf 101 interface ethernet 1/11 vlan 309 exit exit exit </pre>	

NX-OS スタイルの CLI を使用した APIC の特定のポートへの EPG の導入

手順

ステップ 1 VLAN ドメインを設定します。

例 :

```
apic1(config)# vlan-domain dom1
apic1(config-vlan)# vlan 10-100
```

ステップ 2 テナントを作成します。

例 :

```
apic1# configure
apic1(config)# tenant t1
```

ステップ 3 プライベート ネットワーク/VRF を作成します。

例 :

```
apic1(config-tenant)# vrf context ctx1
apic1(config-tenant-vrf)# exit
```

ステップ 4 ブリッジ ドメインを作成します。

例 :

```
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member ctx1
apic1(config-tenant-bd)# exit
```

ステップ 5 アプリケーション プロファイルおよびアプリケーション EPG を作成します。

例 :

```
apic1(config-tenant)# application AP1
apic1(config-tenant-app)# epq EPG1
apic1(config-tenant-app-epg)# bridge-domain member bd1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
```

ステップ 6 EPG を特定のポートに関連付けます。

例 :

```
apic1(config)# leaf 1017
apic1(config-leaf)# interface ethernet 1/13
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 20 tenant t1 application AP1 epq
EPG1
```

(注) 上の例に示した `vlan-domain` コマンドと `vlan-domain member` コマンドは、ポートに EPG を導入するための前提条件です。

ポート セキュリティの設定

ポート セキュリティと ACI について

ポート セキュリティ機能は、ポートごとに取得される MAC アドレスの数を制限することによって、不明な MAC アドレスでフラッドिंगしないように ACI ファブリックを保護します。ポート セキュリティ機能のサポートは、物理ポート、ポート チャネル、および仮想ポート チャネルで使用できます。

ポート セキュリティに関するガイドラインと制約事項

次のようなガイドラインと制約事項があります。

- ポート セキュリティは、ポートごとに使用できます。
- ポートセキュリティは、物理ポート、ポートチャネル、および仮想ポートチャネル（vPC）でサポートされています。
- スタティック MAC アドレスとダイナミック MAC アドレスがサポートされています。
- セキュアなポートからセキュアでないポートへと、セキュアでないポートからセキュアなポートへの MAC アドレスの移動がサポートされています。
- MAC アドレスの制限は、MAC アドレスにのみ適用され、MAC と IP によるアドレスには実行されません。
- ポート セキュリティは、ファブリック エクステンダ（FEX）ではサポートされていません。

ポート レベルでのポート セキュリティ

APIC では、ユーザがスイッチポートのポートセキュリティを設定できます。ポート上で MAC が制限の最大設定値を超過すると、超過した MAC アドレスからすべてのトラフィックが転送されます。次の属性がサポートされます。

- **ポート セキュリティのタイムアウト**：現在サポートされているタイムアウト値は、60 ～ 3600 秒の範囲でサポートされています。
- **違反行為**：違反行為は保護モードで使用できます。保護モードでは、MAC の取得が無効になるため、MAC アドレスは CAM テーブルに追加されません。Mac ラーニングが設定されているタイムアウト値の後に再度有効になります。
- **最大エンドポイント**：現在のサポートされている最大のエンドポイント設定値は、0 ～ 12000 の範囲でサポートされています。最大エンドポイント値が 0 の場合、そのポートではポート セキュリティ ポリシーが無効になります。

ポート セキュリティ ポリシー グループ テンプレートの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーション モードに入ります。
ステップ 2	[no] template policy-group <i>policy-group-name</i> 例 : apic1(config)# template policy-group PortSecGrp1	ポリシー グループ テンプレートを作成 (または削除) します。
ステップ 3	[no] switchport access vlan <i>vlan-id</i> tenant <i>tenant-name</i> application <i>application-name</i> epg <i>epg-name</i> 例 : apic1(config-pol-grp-if)# switchport access vlan 4 tenant ExampleCorp application Web epg webEpg	
ステップ 4	[no] switchport port-security maximum <i>number-of-addresses</i> 例 : apic1(config-pol-grp-if)# switchport port-security maximum 1	ポートのセキュア MAC アドレスの最大数を設定します。範囲は 0 ～ 12000 アドレスです。デフォルトは 1 アドレスです。
ステップ 5	[no] switchport port-security violation protect 例 : apic1(config-pol-grp-if)# switchport port-security violation protect	セキュリティ違反が検出された場合に実行するアクションを設定します。 protect アクションは、十分な数のセキュア MAC アドレスを削除して最大値を下回るまで、不明な送信元アドレスのパケットをドロップします。
ステップ 6	exit 例 : apic1(config-pol-grp-if)# exit	グローバル コンフィギュレーション モードに戻ります。

例

次に、ポートセキュリティ ポリシー グループ テンプレートを作成する例を示します。

```

apic1# configure
apic1(config)# template policy-group PortSecGrp1
apic1(config-pol-grp-if)# switchport port-security maximum 20
apic1(config-pol-grp-if)# switchport port-security violation protect
apic1(config-pol-grp-if)# exit

```

次のタスク

インターフェイスにポートセキュリティ テンプレートを適用します。

テンプレートを使用したインターフェイスでのポートセキュリティの設定

始める前に

ポートセキュリティ ポリシー グループ テンプレートを作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーション モードに入ります。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	interface type-or-range 例 : apic1(config-leaf)# interface eth 1/2-4	設定するポートまたはポートの範囲を指定します。
ステップ 4	[no] policy-group policy-group-name 例 : apic1(config-leaf-if)# policy-group PortSecGrp1	ポートまたはポートの範囲にポリシー グループ テンプレートを適用します。

例

次に、イーサネット ポートの範囲にポートセキュリティ ポリシー グループ テンプレートを適用する例を示します。

```

apic1# configure

```

```

apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/2-4
apic1(config-leaf-if)# policy-group PortSecGrp1

```

次に、テンプレートを使用してポートチャネルでポートセキュリティを設定する例を示します。

```

apic1# configure
apic1(config)# template port-channel pol
apic1(config-if)# switchport port-security maximum 10
apic1(config-if)# switchport port-security violation protect
apic1(config-if)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/3-4
apic1(config-leaf-if)# channel-group pol
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit

```

オーバーライドを使用したインターフェイスでのポートセキュリティの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーション モードに入ります。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	interface type-or-range 例 : apic1(config-leaf)# interface eth 1/2-4	設定するインターフェイスまたはインターフェイスの範囲を指定します。
ステップ 4	[no] switchport port-security maximum number-of-addresses 例 : apic1(config-leaf-if)# switchport port-security maximum 1	インターフェイスのセキュア MAC アドレスの最大数を設定します。範囲は 0 ～ 12000 アドレスです。デフォルトは 1 アドレスです。
ステップ 5	[no] switchport port-security violation protect 例 :	セキュリティ違反が検出された場合に実行するアクションを設定します。 protect アクションは、十分な数のセキュア MAC アドレスを削除して最大値を下回

	コマンドまたはアクション	目的
	<code>apic1(config-leaf-if)# switchport port-security violation protect</code>	るまで、不明な送信元アドレスのパケットをドロップします。

例

次に、イーサネットインターフェイスでポートセキュリティを設定する方法を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/2
apic1(config-leaf-if)# switchport port-security maximum 10
apic1(config-leaf-if)# switchport port-security violation protect
```

次に、ポートチャネルでポートセキュリティを設定する例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po2
apic1(config-leaf-if)# switchport port-security maximum 10
apic1(config-leaf-if)# switchport port-security violation protect
```

次に、仮想ポートチャネル（VPC）でポートセキュリティを設定する例を示します。

```
apic1# configure
apic1(config)# vpc domain explicit 1 leaf 101 102
apic1(config-vpc)# exit
apic1(config)# template port-channel po4
apic1(config-if)# exit
apic1(config)# leaf 101-102
apic1(config-leaf)# interface eth 1/11-12
apic1(config-leaf-if)# channel-group po4 vpc
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# interface vpc po4
apic1(config-vpc-if)# switchport port-security maximum 10
apic1(config-vpc-if)# switchport port-security violation protect
```

802.1x ポートおよびノードの認証

802.1x ポートおよびノードの認証

IEEE 802.1x はポートベースの認証メカニズムで、不正なデバイスがネットワークにアクセスするのを防止します。NX-OS スタイル CLI を使用して、802.1x ポートおよびノードの認証を設定することができます。

ポート認証 ポリシーの設定

手順

-
- ステップ 1** CLI で、コンフィギュレーション モードに入ります。
- 例 :
- ```
apicl# configure
apicl(config)#
```
- ステップ 2** ポリシー グループを作成します:
- 例 :
- ```
apicl(config)# template policy-group mypol
```
- ステップ 3** ポリシー グループでポート レベル認証ポリシーを設定します。
- 例 :
- ```
apicl(config-pol-grp-if)# switchport port-authentication mydot1x
```
- ステップ 4** ホストモードを設定します (2 個のモードがサポートされます: マルチホストおよび単一ホスト: 単一ホストがデフォルトでは設定されています)。
- 例 :
- ```
apicl(config-port-authentication)# host-mode multi-host
```
- ステップ 5** このポリシーを有効にします (ポリシーはデフォルトでは無効です)。
- 例 :
- ```
apicl(config-port-authentication)# no shutdown
apicl(config-port-authentication)# exit
apicl(config-pol-grp-if)# exit
apicl(config)#
```
- ステップ 6** リーフ インターフェイス ポリシーを設定します。
- 例 :
- ```
apicl(config)#leaf-interface-profile myprofile
```
- ステップ 7** リーフ スイッチ インターフェイス プロファイルのポリシー グループを設定します。
- 例 :
- ```
apicl(config-leaf-if-profile)#leaf-interface-group mygroup
```
- ステップ 8** インターフェイス グループのポートおよびインターフェイスを指定します。
- 例 :
- ```
apicl(config-leaf-if-group)# interface ethernet 1/10-12
```
- ステップ 9** インターフェイス グループにポリシーを適用します。
- 例 :

```
apic1(config-leaf-if-group)# policy-group mypol
apic1(config-leaf-if-group)# exit
apic1(config-leaf-if-profile)# exit
```

ステップ 10 リーフ プロファイルを設定します。

例：

```
apic1(config)#
apic1(config)# leaf-profile myleafprofile
```

ステップ 11 リーフ ポリシー グループを設定し、グループのリーフ スイッチ ノードを指定します。

例：

```
apic1(config-leaf-profile)# leaf-group myleafgrp
apic1(config-leaf-group)# leaf 101
apic1(config-leaf-group)# exit
```

ステップ 12 リーフ スイッチ プロファイルにインターフェイス ポリシーを適用します。

例：

```
apic1(config-leaf-profile)# leaf-interface-profile myprofile
apic1(config-leaf-group)# exit
apic1(config)#
```

ノード認証ポリシーの設定

手順

ステップ 1 CLI で、コンフィギュレーション モードに入ります。

例：

```
apic1# configure
apic1(config)#
```

ステップ 2 Radius 認証グループを設定します。

例：

```
apic1(config)# aaa group server radius myradiusgrp
apic1(config-radius)#server 192.168.0.100 priority 1
apic1(config-radius)#exit
```

ステップ 3 ノード レベル ポート認証ポリシーを設定します。

例：

```
apic1(config)# policy-map type port-authentication mydot1x
apic1(config-pmap-port-authentication)#radius-provider-group myradiusgrp
```

ステップ 4 [オプション]認証が失敗した場合は、デフォルトの VLAN ID をオーバーライドします。:

例：

```
apic1(config-pmap-port-authentication)#fail-auth-vlan 2001
```


ステップ 5 [オプション]認証が失敗した場合は、デフォルト EPG をオーバーライドします。

例 :

```
apic1(config-pmap-port-authentication)#fail-auth-epg tenant tn1 application ap1 epg
epg256
apic1(config)# exit
```

ステップ 6 ポリシー グループを設定し、グループ内でポート認証ポリシーを指定します。

例 :

```
apic1(config)#template leaf-policy-group lpg2
apic1(config-leaf-policy-group)# port-authentication mydot1x
apic1(config-leaf-policy-group)#exit
```

ステップ 7 リーフ スイッチ プロファイルを設定します。

例 :

```
apic1(config)# leaf-profile mylp2
```

ステップ 8 リーフ スイッチのプロファイルのグループを設定し、ポリシー グループを指定します。

例 :

```
apic1(config-leaf-profile)#leaf-group mylg2
apic1(config-leaf-group)# leaf-policy-group lpg2
apic1(config-leaf-group)#exit
```

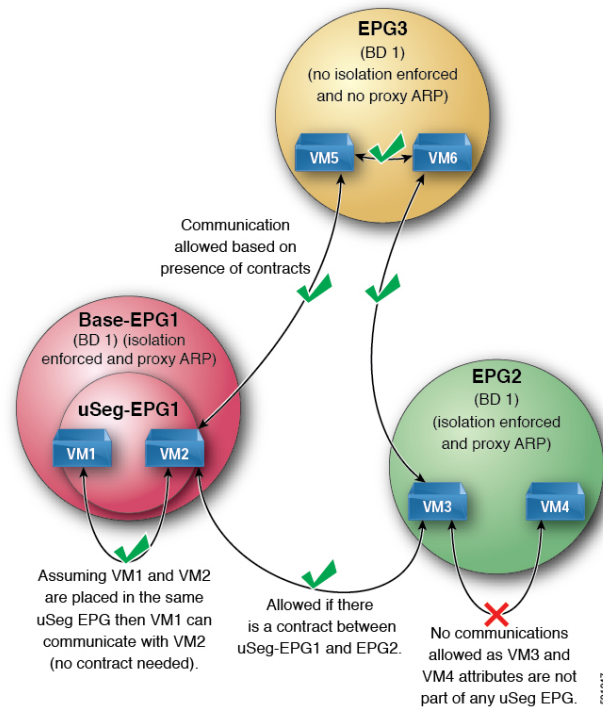
プロキシ ARP の設定

プロキシ ARP について

Cisco ACI のプロキシ ARP は、ネットワークまたはサブネット内のエンドポイントが、別のエンドポイントの MAC アドレスを知らなくても、そのエンドポイントと通信できるようにします。プロキシ ARP はトラフィックの宛先場所を知っており、代わりに、最終的な宛先として自身の MAC アドレスを提供します。

プロキシ ARP を有効にするには、EPG 内エンドポイント分離を EPG で有効にする必要があります。詳細については、次の図を参照してください。EPG 内エンドポイント分離と Cisco ACI の詳細については、「Cisco ACI 仮想化ガイド」を参照してください。

図 7: プロキシ ARP および Cisco APIC



Cisco ACI ファブリック内のプロキシ ARP は従来のプロキシ ARP とは異なります。通信プロセスの例として、プロキシ ARP が EPG で有効になっているとき、エンドポイント A が ARP 要求をエンドポイント B に送信し、エンドポイント B がファブリック内で学習される場合、エンドポイント A はブリッジドメイン (BD) MAC からプロキシ ARP 応答を受信します。エンドポイント A が B、エンドポイントの ARP 要求を送信し、エンドポイント B はすでに ACI ファブリック内で学習しない場合は、ファブリックはプロキシ ARP の BD 内で要求を送信します。エンドポイント B は、ファブリックに戻る要求、このプロキシ ARP に応答します。この時点では、ファブリックはプロキシ ARP エンドポイント A への応答を送信しませんが、エンドポイント B は、ファブリック内で学習します。エンドポイント A は、エンドポイント B に別の ARP 要求を送信する場合、ファブリックはプロキシ ARP 応答から送信 BD mac です。

次の例ではプロキシ ARP 解像度がクライアント VM1 と VM2 間の通信の手順します。

1. VM2 通信を VM1 が必要です。

図 8: VM2 通信を VM1 が必要です。

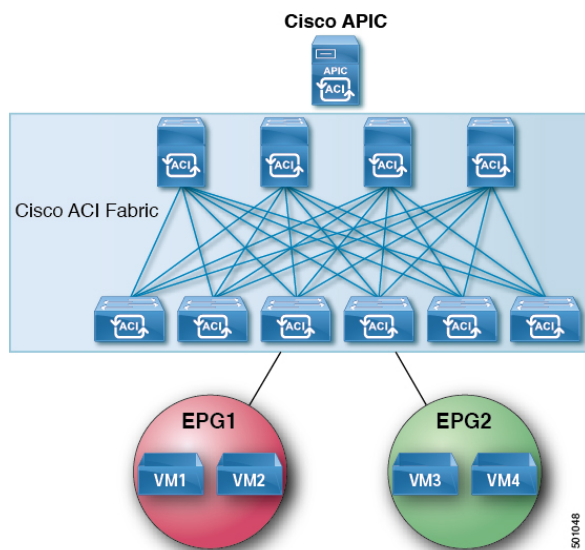


表 12: ARP 表の説明

デバイス	状態
VM1	IP = * MAC = *
ACI ファブリック	IP = * MAC = *
VM2	IP = * MAC = *

2. VM1 は、ブロードキャスト MAC アドレスとともに ARP 要求を VM2 に送信します。

図 9: VM1 はブロードキャスト MAC アドレスとともに ARP 要求を VM2 に送信します

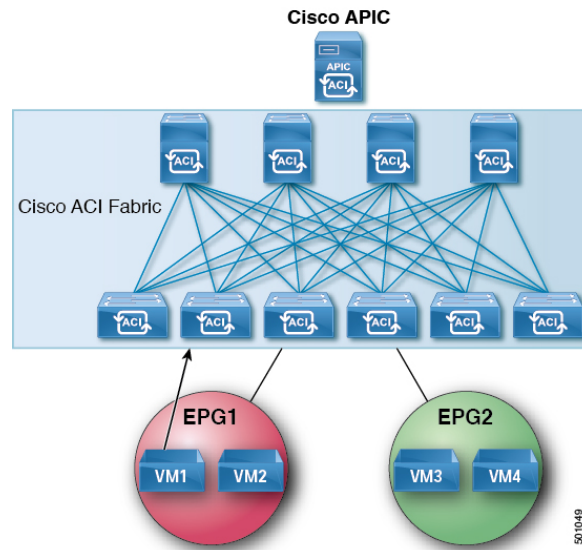


表 13: ARP 表の説明

デバイス	状態
VM1	IP = VM2 IP; MAC = ?
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC
VM2	IP = * MAC = *

3. ACI ファブリックは、ブリッジ ドメイン (BD) 内のプロキシ ARP 要求をフラッディングします。

図 10: ACI ファブリックは BD 内のプロキシ ARP 要求をフラッディングします

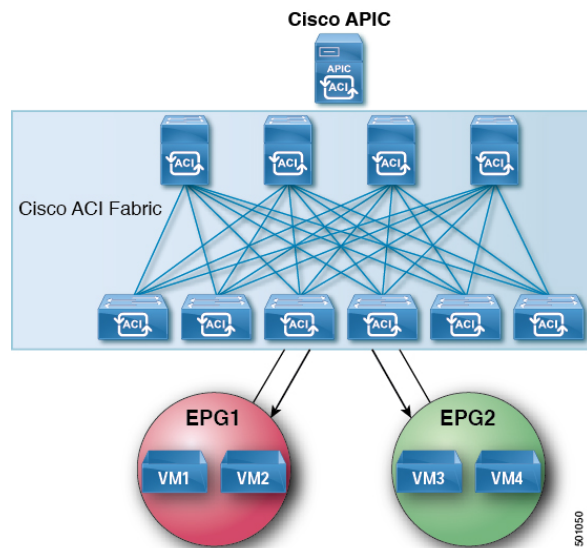


表 14: ARP 表の説明

デバイス	状態
VM1	IP = VM2 IP; MAC = ?
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC
VM2	IP = VM1 IP; MAC = BD MAC

4. VM2 は、ARP 応答を ACI ファブリックに送信します。

図 11: VM2 は ARP 応答を ACI ファブリックに送信します

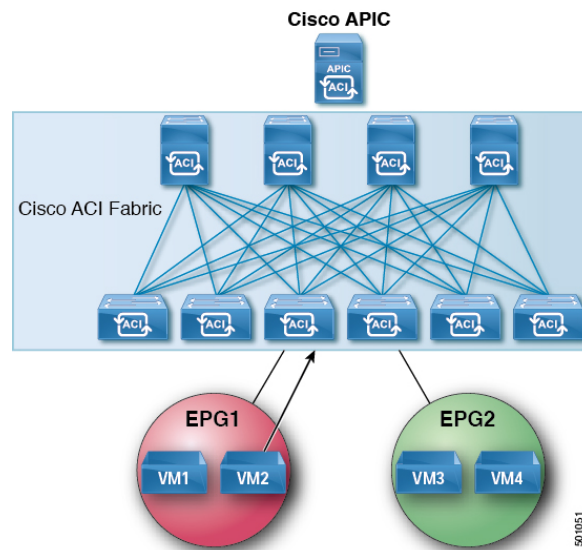


表 15: ARP 表の説明

デバイス	状態
VM1	IP = VM2 IP; MAC = ?
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC
VM2	IP = VM1 IP; MAC = BD MAC

5. VM2 が学習されます。

図 12: VM2 が学習されます

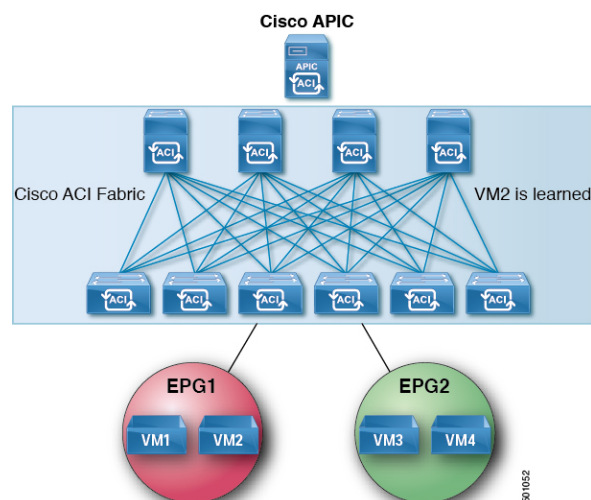


表 16: ARP 表の説明

デバイス	状態
VM1	IP = VM2 IP; MAC = ?
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC IP = VM2 IP; MAC = VM2 MAC
VM2	IP = VM1 IP; MAC = BD MAC

6. VM1 は、ブロードキャスト MAC アドレスとともに ARP 要求を VM2 に送信します。

図 13: VM1 はブロードキャスト MAC アドレスとともに ARP 要求を VM2 に送信します

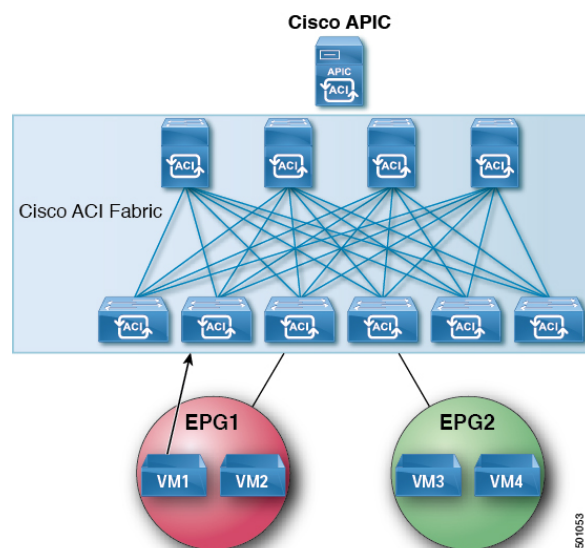


表 17: ARP 表の説明

デバイス	状態
VM1	IP = VM2 IP; MAC = ?
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC IP = VM2 IP; MAC = VM2 MAC
VM2	IP = VM1 IP; MAC = BD MAC

7. ACI ファブリックは、プロキシ ARP VM1 への応答を送信します。

図 14: ACI ファブリック VM1 にプロキシ ARP 応答を送信します。

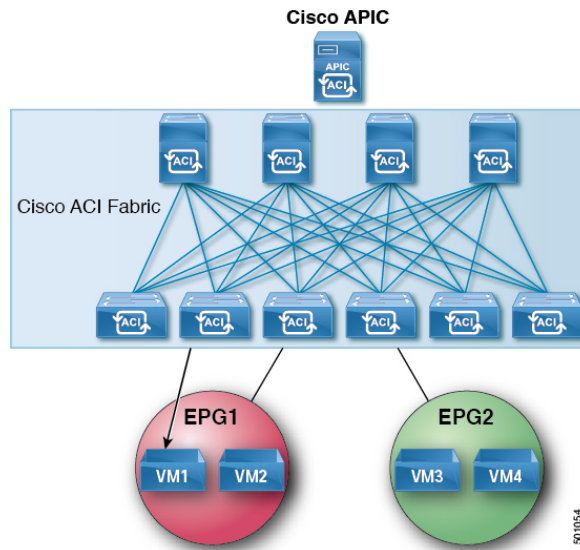


表 18: ARP 表の説明

デバイス	状態
VM1	IP = VM2 IP; MAC = BD MAC
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC IP = VM2 IP; MAC = VM2 MAC
VM2	IP = VM1 IP; MAC = BD MAC

注意事項と制約事項

プロキシ ARP を使用すると、次のガイドラインと制限事項を考慮してください。

- プロキシ ARP は、隔離 Epg でのみサポートされます。EPG が隔離ではない場合、障害が発生します。プロキシ ARP が有効になっていると隔離 Epg 内で発生する通信では、uSeg Epg を設定する必要があります。たとえば、隔離の EPG 内で別の IP アドレスを持つ複数の Vm がある可能性があり、これらの Vm の IP address range(IP アドレス範囲、IP アドレスの範囲) に一致する IP の属性を持つ uSeg EPG を設定することができます。
- 隔離されたエンドポイントを通常のエンドポイントと、定期的なエンドポイントを隔離のエンドポイントからの ARP 要求には、プロキシ ARP は使用しないでください。このような場合は、エンドポイントは、接続先の Vm の実際の MAC アドレスを使用して通信します。

プロキシ ARP は、Cisco NX-OS スタイル CLI を使用しての設定

始める前に

- 適切なテナント、VRF、ブリッジ ドメイン、アプリケーションプロファイルおよび EPG を作成する必要があります。
- プロキシ ARP が有効にするのが EPG で内通 EPG の分離を有効にする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーション モードに入ります。
ステップ 2	tenant tenant-name 例 : <code>apic1(config)# tenant Tenant1</code>	テナント コンフィギュレーション モードを開始します。
ステップ 3	application application-profile-name 例 : <code>apic1(config-tenant)# application Tenant1-App</code>	アプリケーション プロファイルを作成し、アプリケーション モードを開始します。
ステップ 4	epg application-profile-EPG-name 例 : <code>apic1(config-tenant-app)# epg Tenant1-epg1</code>	EPG を作成し、EPG モードに入ります。
ステップ 5	proxy-arp enable 例 : <code>apic1(config-tenant-app-epg)# proxy-arp enable</code>	プロキシ ARP を有効にします。 (注) プロキシ arp をディセーブルにできます、 no プロキシ arp コマンド。
ステップ 6	exit 例 : <code>apic1(config-tenant-app-epg)# exit</code>	ポート アプリケーション モードに戻ります。
ステップ 7	exit 例 : <code>apic1(config-tenant-app)# exit</code>	テナント コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 8	exit 例 : apicl(config-tenant)# exit	グローバル コンフィギュレーション モードに戻ります。

例

次に、プロキシ ARP を設定する例を示します。

```

apicl# conf t
apicl(config)# tenant Tenant1
apicl(config-tenant)# application Tenant1-App
apicl(config-tenant-app)# epg Tenant1-epg1
apicl(config-tenant-app-epg)# proxy-arp enable
apicl(config-tenant-app-epg)#
apicl(config-tenant)#

```

カプセル化のフラッディングの設定

レイヤ 2 外部接続の設定はスタティック アプリケーション EPG と似ていて、ノード、ポートの VLAN を EPG にマッピングし、EPG をブリッジドメインにマッピングして、コントラクトを提供または使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	コンフィギュレーション モードに入ります。
ステップ 2	tenant <i>tenant-name</i> 例 : apicl(config)# tenant Tenant1	テナント コンフィギュレーション モードを開始します。
ステップ 3	application <i>application-profile-name</i> 例 : apicl(config)# application Tenant1-App	アプリケーション プロファイルを作成し、アプリケーション モードを開始します。
ステップ 4	epg <i>application-profile-EPG-name</i> 例 : apicl(config)# epg Tenant1-epg1	EPG を作成し、EPG モードを開始します。
ステップ 5	flood-on-encapsulation enable 例 :	Flood-on-encapsulation を有効にします。

	コマンドまたはアクション	目的
	<code>apic1(config-tenant-app-epg)# flood-on-encapsulation enable</code>	
ステップ 6	<code>exit</code> 例 : <code>apic1(config-tenant-app-epg)# exit</code>	アプリケーション プロファイル モード に戻ります。

MACsec の設定

MACsec について

MACsec は、IEEE 802.1AE 規格ベースのレイヤ 2 ホップバイホップ暗号化であり、これにより、メディア アクセス非依存プロトコルに対してデータの機密性と完全性を確保できます。

MACsec は、暗号化キーにアウトオブバンド方式を使用して、有線ネットワーク上で MAC レイヤの暗号化を提供します。MACsec Key Agreement (MKA) プロトコルでは、必要なセッション キーを提供し、必要な暗号化キーを管理します。

802.1 ae MKA と暗号化はリンク、つまり、リンク (ネットワーク アクセス デバイスと、PC か IP 電話機などのエンドポイント デバイス間のリンク) が直面しているホストのすべてのタイプでサポートされます。リンクが接続されている他のスイッチまたはルータ。

MACsec は、イーサネット パケットの送信元および宛先 MAC アドレスを除くすべてのデータを暗号化します。ユーザは、送信元と宛先の MAC アドレスの後に最大 50 バイトの暗号化をスキップするオプションもあります。

WAN またはメトロ イーサネット上に MACsec サービスを提供するために、サービス プロバイダーは、Ethernet over Multiprotocol Label Switching (EoMPLS) および L2TPv3 などのさまざまなトランスポート レイヤ プロトコルを使用して、E-Line や E-LAN などのレイヤ 2 透過サービスを提供しています。

EAP-over-LAN (EAPOL) プロトコルデータユニット (PDU) のパケット本体は、MACsec Key Agreement PDU (MKPDU) と呼ばれます。3 回のハートビート後 (各ハートビートは 2 秒) に参加者から MKPDU を受信しなかった場合、ピアはライブ ピア リストから削除されます。たとえば、クライアントが接続を解除した場合、スイッチ上の参加者はクライアントから最後の MKPDU を受信した後、3 回のハートビートが経過するまで MKA の動作を継続します。

APIC ファブリック MACsec

APIC はまたは責任を負う MACsec キーチェーン ディストリビューションのポッド内のすべてのノードに特定のポートのノードになります。サポートされている MACsec キーチェーンし、apic 内でサポートされている MACsec ポリシー ディストリビューションのとおりです。

- 単一ユーザ提供キーチェーンと 1 ポッドあたりポリシー

- ユーザが提供されるキーチェーンとファブリックインターフェイスごとのユーザが提供されるポリシー
- 自動生成されたキーチェーンおよび 1 ポッドあたりのユーザが提供されるポリシー

ノードは、複数のポリシーは、複数のファブリックリンクの導入を持つことができます。これが発生すると、ファブリックインターフェイスごとキーチェーンおよびポリシーが優先して指定の影響を受けるインターフェイス。自動生成されたキーチェーンと関連付けられている MACsec ポリシーでは、最も優先度から提供されます。

APIC MACsec では、2つのセキュリティモードをサポートしています。MACsec **セキュリティで保護する必要があります** 中に、リンクの暗号化されたトラフィックのみを許可する **セキュリティで保護する必要があります** により、両方のクリアし、リンク上のトラフィックを暗号化します。MACsec を展開する前に **セキュリティで保護する必要があります** モードでのキーチェーンは影響を受けるリンクで展開する必要がありますまたはリンクがダウンします。たとえば、ポートをオンにできますで MACsec **セキュリティで保護する必要があります** モードがピアがしているリンクでのキーチェーンを受信する前にします。MACsec を導入することが推奨されて、この問題に対処する **セキュリティで保護する必要があります** モードとリンクの 1 回すべてにセキュリティモードを変更 **セキュリティで保護する必要があります** 。



(注) MACsec インターフェイスの設定変更は、パケットのドロップになります。

MACsec ポリシー定義のキーチェーンの定義に固有の設定と機能の機能に関連する設定で構成されています。キーチェーン定義と機能の機能の定義は、別のポリシーに配置されます。MACsec 1 ポッドあたりまたはインターフェイスごとの有効化には、キーチェーン ポリシーおよび MACsec 機能のポリシーを組み合わせることが含まれます。



(注) 内部を使用して生成キーチェーンは、ユーザのキーチェーンを指定する必要はありません。

APIC アクセス MACsec

MACsec はリーフ スイッチ L3out インターフェイスと外部のデバイス間のリンクを保護するために使用します。APIC GUI および CLI のユーザを許可するで、MACsec キーとファブリック L3Out インターフェイスの設定を MacSec をプログラムを提供する物理/pc/vpc インターフェイスごと。ピアの外部デバイスが正しい MacSec 情報を使用してプログラムすることを確認するには、ユーザの責任です。

MACSec の注意事項と制約事項

次の注意事項と制約事項に従って MACSec を設定します。

- Fex ポートは MACsec ではサポートされていません。
- ポッド レベルでは、必須セキュア モードはサポートされていません。

- 「デフォルト」名の MACsec ポリシーはサポートされていません。
- 自動キー生成はファブリック ポートのポッド レベルでのみサポートされます。
- MACSEC はリモート リーフではサポートされていません。
- そのノードのファブリック ポートが **[必須セキュア]** モードの MACsec で実行されている場合、ノードの再起動をクリアしないでください。
- MACsec を実行しているポッド内のノードのポッドまたはステートレス再起動に新しいノードを追加することで、ノードがポッドに参加するために **[必須セキュア]** モードを **[should-secure]** モードに変更する必要があります。
- ファブリック のリンクが **[should-secure]** モードの場合、アップグレイド/ダウングレードのみを開始する必要があります。アップグレード/ダウングレードが完了すると、次にモードを **[必須セキュア]** に変更できます。**[必須セキュア]** モードでのアップグレード/ダウングレードは、ノードからファブリック への接続が失われます。接続の遮断から回復するには、APIC に表示されるノードのファブリック リンクを **[should-secure]** モードに設定する必要があります。ファブリックが MACsec をサポートしていないバージョンにダウングレードされた場合、ファブリック外のノードがクリーンリブートされる必要があります。
- PC/VPC インターフェイスでは、PC/VPC インターフェイスごとのポリシー グループによって MACsec を展開できます。ポート セレクタは、特定のポートのセットにポリシーを展開するために使用されます。したがって、L3Out インターフェイスに対応する正しいポート セレクタを作成することはユーザの責任です。
- 設定がエクスポートされる前に、MACsec ポリシーを **[should-secure]** モードに設定することをお勧めします。
- スパイン上のすべてのリンクは、ファブリックリンクと見なされます。ただし、スパインリンクの IPN 接続を使用するとアクセス リンクとして次のリンクが処理されます。これは、MACsec アクセス ポリシーを次のリンクの MACsec を導入するために使用する必要があります。
- MACSEC セッションは、フォームに分をかかるとは空のキーチェーンに新しいキーが追加されるか、アクティブ キーがキーチェーンから削除切断する可能性があります。

must-secure モードの展開

不正な導入手順に設定されているポリシーの **必須 secure** モードが接続の消失で発生することができます。そのような問題を避けるため次の手順に従う必要があります。

- MACsec **[必須セキュア]** モードを有効にする前に、各リンク ペアがそれぞれのキーチェーンを持っていることを確認する必要があります。これを確認するために推奨されることは、**[should-secure]** モードでポリシーを展開し、MACsec セッションが予想されるリンク上でアクティブになったら、モードを **[必須セキュア]** に変更することです。
- **[必須セキュア]** に設定されている MACsec ポリシーでキーチェーンの交換を試行すると、リンクがダウンする原因となる可能性があります。この場合、以下の推奨手順に従う必要があります。

- 新しいキーチェーンを使用している MACsec ポリシーを **[should-secure]** モードに変更します。
 - 影響を受けるインターフェイスが **[should-secure]** モードを使用しているか確認します。
 - 新しいキーチェーンを使用するように MACsec ポリシーを更新します。
 - アクティブな MACsec セッションと関連するインターフェイスが新しいキーチェーンを使用していることを確認します。
 - MACsec ポリシーを **[必須セキュア]** モードに変更します。
- **[必須セキュア]** モードに展開されている MACsec ポリシーを無効/削除するには、次の手順に従う必要があります。
 - MACsec ポリシーを **[should-secure]** に変更します。
 - 影響を受けるインターフェイスが **[should-secure]** モードを使用しているか確認します。
 - MACsec ポリシーを無効/削除します。

キーチェーンの定義

- 開始時刻が **[現在]** のキーチェーンに 1 個のキーが存在します。 **must-secure** がすぐにアクティブになるキーを持たないキーチェーンで展開される場合、キーが現在時刻になり MACsec セッションが開始されるまでトラフィックはリンク上でブロックされます。 **should-secure** モードが使用されている場合、キーが現在になり、MACsec セッションが開始されるまでトラフィックが暗号化されます。
- 終了時刻が **infinite** のキーチェーンに 1 個のキーが存在する必要があります。キーチェーンの期限が切れると、 **must-secure** モードに設定されている影響を受けるインターフェイスでトラフィックがブロックされます。設定されたインターフェイスは **セキュア** モード暗号化されていないトラフィック送信します。
- 終了時刻のオーバーラップし、キーの間に移行すると、MACsec セッションを順番に使用されるキーの開始時刻が残っています。

NX-OS スタイルの CLI を使用した MACsec の設定

手順

ステップ 1 アクセス インターフェイスの MACsec セキュリティ ポリシーの設定

例 :

```
apic1# configure
apic1(config)# template macsec access security-policy accmacsecpoll
apic1(config-macsec-param)# cipher-suite gcm-aes-128
```

```

apicl(config-macsec-param)#      conf-offset offset-30
apicl(config-macsec-param)#      description 'description for mac sec parameters'
apicl(config-macsec-param)#      key-server-priority 1
apicl(config-macsec-param)#      sak-expiry-time 110
apicl(config-macsec-param)#      security-mode must-secure
aapicl(config-macsec-param)#      window-size 1
apicl(config-macsec-param)#      exit
apicl(config)#

```

ステップ 2 アクセス インターフェイスの MACsec キー チェーンを設定します。

PSK は、2 通りの方法で設定できます:

- (注)
- 下のキー 12ab に示すように、**psk-string** コマンドを使用してインラインで設定します。PSK は、ログに記録され、公開されるため、安全ではありません。
 - キー ab12 で示すように、新しいコマンド **Enter PSK string** を **psk-string** コマンドの後で使用し、個別に入力して設定します。ローカルにエコーされるだけで、ログには記録されないため、PSK は安全です。

例 :

```

apicl# configure
apicl(config)#      template macsec access keychain acckeychainpoll
apicl(config-macsec-keychain)#      description 'macsec key chain kc1'
apicl(config-macsec-keychain)#      key 12ab
apicl(config-macsec-keychain-key)#      life-time start 2017-09-19T12:03:15 end
2017-12-19T12:03:15
apicl(config-macsec-keychain-key)#      psk-string 123456789a223456789a323456789abc
apicl(config-macsec-keychain-key)#      exit
apicl(config-macsec-keychain)#      key ab12
apicl(config-macsec-keychain-key)#      life-time start now end infinite
apicl(config-macsec-keychain-key)#      life-time start now end infinite
apicl(config-macsec-keychain-key)#      psk-string
Enter PSK string: 123456789a223456789a323456789abc
apicl(config-macsec-keychain-key)#      exit
apicl(config-macsec-keychain)#      exit
apicl(config)#

```

ステップ 3 アクセス インターフェイスの MACsec インターフェイス ポリシーを設定します:

例 :

```

apicl# configure
apicl(config)#      template macsec access interface-policy accmacsecifpoll
apicl(config-macsec-if-policy)#      inherit macsec security-policy accmacsecpoll keychain
acckeychainpoll
apicl(config-macsec-if-policy)#      exit
apicl(config)#

```

ステップ 4 MACsec インターフェイス ポリシーをリーフ (またはスパイン) 上のアクセス インターフェイスに関連付けます:

例 :

```

apicl# configure
apicl(config)#      template macsec access interface-policy accmacsecifpoll
apicl(config-macsec-if-policy)#      inherit macsec security-policy accmacsecpoll keychain
acckeychainpoll
apicl(config-macsec-if-policy)#      exit
apicl(config)#

```

ステップ 5 ファブリック インターフェイス用に MACsec セキュリティ ポリシーを設定します:

例:

```

apic1# configure
apic1(config)# template macsec fabric security-policy fabmacsecpoll
apic1(config-macsec-param)# cipher-suite gcm-aes-xpn-128
apic1(config-macsec-param)# description 'description for mac sec parameters'
apic1(config-macsec-param)# window-size 1
apic1(config-macsec-param)# sak-expiry-time 100
apic1(config-macsec-param)# security-mode must-secure
apic1(config-macsec-param)# exit
apic1(config)#

```

ステップ 6 ファブリック インターフェイス用に MACsec キー チェーンを設定します:

PSK は、2 通りの方法で設定できます:

- (注)
- 下のキー 12ab に示すように、**psk-string** コマンドを使用してインラインで設定します。PSK は、ログに記録され、公開されるため、安全ではありません。
 - キー ab12 で示すように、新しいコマンド **Enter PSK string** を **psk-string** コマンドの後で使用し、個別に入力して設定します。ローカルにエコーされるだけで、ログには記録されないため、PSK は安全です。

例:

```

apic1# configure
apic1(config)# template macsec fabric security-policy fabmacsecpoll
apic1(config-macsec-param)# cipher-suite gcm-aes-xpn-128
apic1(config-macsec-param)# description 'description for mac sec parameters'
apic1(config-macsec-param)# window-size 1
apic1(config-macsec-param)# sak-expiry-time 100
apic1(config-macsec-param)# security-mode must-secure
apic1(config-macsec-param)# exit
apic1(config)# template macsec fabric keychain fabkeychainpoll
apic1(config-macsec-keychain)# description 'macsec key chain kc1'
apic1(config-macsec-keychain)# key 12ab
apic1(config-macsec-keychain-key)# psk-string 123456789a223456789a323456789abc
apic1(config-macsec-keychain-key)# life-time start 2016-09-19T12:03:15 end
2017-09-19T12:03:15
apic1(config-macsec-keychain-key)# exit
apic1(config-macsec-keychain)# key cd78
apic1(config-macsec-keychain-key)# psk-string
Enter PSK string: 123456789a223456789a323456789abc
apic1(config-macsec-keychain-key)# life-time start now end infinite
apic1(config-macsec-keychain-key)# exit
apic1(config-macsec-keychain)# exit
apic1(config)#

```

ステップ 7 MACsec インターフェイス ポリシーをリーフ (またはスパイン) 上のファブリック インターフェイスに関連付けます:

例:

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# fabric-interface ethernet 1/52-53
apic1(config-leaf-if)# inherit macsec interface-policy fabmacsecifpol2

```



```
apicl(config-leaf-if)#      exit
apicl(config-leaf)#
```



第 7 章

レイヤ 3 外部接続の設定

- [レイヤ 3 外部接続の設定のモードについて \(179 ページ\)](#)
- [レイヤ 3 外部接続の設定 \(181 ページ\)](#)
- [外部ネットワークへのルーテッド接続 \(182 ページ\)](#)
- [レイヤ 3 ルーティングとサブインターフェイス ポート チャネル \(194 ページ\)](#)
- [レイヤ 3 アウトからレイヤ 3 アウト内部 VRF への漏洩 \(201 ページ\)](#)
- [SVI 外部カプセル化の範囲について \(206 ページ\)](#)
- [SVI 自動状態について \(209 ページ\)](#)
- [インターフェイスとスタティック ルーティングの設定 \(211 ページ\)](#)
- [OSPF の設定 \(214 ページ\)](#)
- [BGP の設定 \(222 ページ\)](#)
- [EIGRP の設定 \(240 ページ\)](#)
- [ルート マップの設定 \(248 ページ\)](#)
- [双方向ルート転送 \(BFD\) の設定 \(259 ページ\)](#)
- [レイヤ 3 マルチキャストの設定 \(275 ページ\)](#)
- [外部 L3 EPG の設定 \(291 ページ\)](#)
- [名前付きモードを使用したレイヤ 3 外部接続の設定 \(294 ページ\)](#)
- [IPv6 ネイバー探索 \(309 ページ\)](#)
- [『Configuring HSRP』 \(314 ページ\)](#)
- [Cisco ACI GOLF \(317 ページ\)](#)
- [Multipod_Fabric \(337 ページ\)](#)
- [リモート リーフ スイッチ \(344 ページ\)](#)
- [トランジット ルーティング \(352 ページ\)](#)

レイヤ 3 外部接続の設定のモードについて

APIC は設定のための複数のユーザ インターフェイス (UI) をサポートしているので、1 つの UI を使用して設定を作成し、その後、別の UI を使用して設定を変更する場合は、予期しないインタラクションが潜んでいます。ここでは、さらに他の APIC のユーザ インターフェイスを使

用した可能性がある場合、APIC NX-OS スタイルの CLI を使用してレイヤ 3 外部接続を設定するための考慮事項を説明します。

APIC NX-OS スタイルの CLI を使用してレイヤ 3 外部接続を設定する場合、次の 2 つのモードを選択することができます。

- よりシンプルな暗黙 モードは、APIC GUI または REST API と互換性がありません。
- 名前付き（または明示）モードは、APIC GUI および REST API と互換性があります。

いずれの場合も、設定は互換性がない UI では読み取り専用であると考えてください。

モードの違いについて

どちらのモードでも、構成設定は API の **I3extOut** クラスのインスタンスである内部コンテナオブジェクト「L3 Outside」（または「L3Out」）内で定義されます。2 つのモード間の主な違いは、このコンテナ オブジェクト インスタンスの命名にあります。

- 暗黙モード：コンテナのネーミングは潜在的であり、CLI コマンドには表示されません。CLI は、これらのオブジェクトを内部的に作成し保持します。
- 名前付きモード：名前はユーザーが決定します。名前付きモードの CLI コマンドには、追加の **I3Out** フィールドがあります。名前付き L3Out がを正常に設定され障害を回避するためには、ユーザーが外部レイヤ 3 用の API オブジェクト モデルを理解する必要があります。



(注) 「名前付きモードセクションを使用したレイヤ 3 外部接続の設定」セクションの手順を除き、このガイドでは、暗黙モードの手順を説明します。

注意事項および制約事項

- 同じ APIC インターフェイスでは、両方のモードを、次の制限でレイヤ 3 外部接続を設定するために一緒に使用することができます。テナント VRF、およびリーフの特定の組み合わせのレイヤ 3 外部接続設定は、1 つのモードを介してのみ実行できます。
- 特定のテナント VRF の場合、外部 L3 EPG を配置できるポリシー ドメインは、名前付きモードまたは暗黙モードのいずれかになります。推奨する設定方式は、特定のテナント VRF が、レイヤ 3 外部接続用に展開されたすべてのノード全体で、特定のテナント VRF の組み合わせに対して 1 つのモードだけを使用することです。モードは、異なるテナントまたは異なる VRF 全体で変えることができ、制限は適用されません。
- 外部レイヤ 3 機能は、次の例外を除いて、両方の設定モードでサポートされます
 - L4 ~ L7 サービス アプライアンスを使用したルーティング ピアリングとルート ヘルスインジェクション(RHI)は、名前付きモードでのみをサポートされます。名前付きモードは、ルーティング ピアリングが含まれるテナント VRF のすべての境界リーフスイッチ全体で使用する必要があります。

- 暗黙モード CLI 手順を使用して作成されたレイヤ 3 外部ネットワーク オブジェクト (l3extOut) は、「_ui_」で始まる名前で識別され、GUI で読み取り専用としてマークされます。CLI は、インターフェイス、プロトコル、ルートマップ、EPG などの機能で、これらの外部 L3 ネットワークを分割します。REST API を介して実行される設定変更は、この構造を破棄することができ、CLI を介してさらなる変更を防ぐことができます。

このようなオブジェクトを削除する手順については、『*APIC Troubleshooting Guide*』の「*Troubleshooting Unwanted _ui_ Objects*」を参照してください。

レイヤ 3 外部接続の設定

外部ネットワークへのレイヤ 3 (L3) ルーティング接続の設定は、次のコンポーネントで構成されています。

- インターフェイス：レイヤ 3 ポートの設定インターフェイス、サブインターフェイス、外部ルータに接続するために使用される外部 SVI。
- ルーティング プロトコルの設定：CLI は静的ルート、BGP、OSPF、EIGRP プロトコル設定をサポートします。
- ルート マップ制御：ルート マップは、prefixes/BD パブリック サブネットと一致させ、ルーティング制御ポリシーを適用するために使用されます。ルート マップが作成されると、「in」（BGP または OSPF）、「out」（BGP、OSPF、EIGRP）などの、方向のルーティング プロトコルと関連付けることができます。

インターフェイス、ルート プロトコル、およびルートマップに関する設定は、構成リーフの設定モードでリーフ スイッチ単位で保持されます。

- L3 外部 EPG：コントラクトと QoS ポリシーを適用するためのエンドポイントグループに分類されるテナント VRF の外部サブネットのリスト。L3 外部 EPG（プレフィックス EPG ともいいます）には、他の外部 L3 EPG およびアプリケーション EPG とコントラクトを持つことができます。L3 外部 EPG 設定はテナント設定で保持されます。L3 外部 EPG は、VRF が設定されているノードのサブセットに展開できます。

レイヤ 3 外部接続を設定するための手順は、次のように要約できます。

1. テナントの下に VRF を作成します。
2. 境界リーフ スイッチ上の VRF を設定して展開します。
3. 境界リーフ インターフェイス上のレイヤ 3 インターフェイスを設定します。
4. リーフ スイッチ上のルート マップを設定します。
5. リーフおよびリーフ インターフェイスの下にルーティング プロトコル（BGP、OSPF、EIGRP）を設定します。
6. テナントの下に L3 外部 EPG を作成して設定します。
7. 境界リーフ スイッチの外部 L3 EPG を展開します。

外部ネットワークへのルーテッド接続

外部ネットワークへルートされた接続について

ネットワーク構成 (L3Out) 外部レイヤ 3 では、ファブリック以外のトラフィックを転送する方法を定義します。レイヤ 3 はし、他のノードのアドレスを見つける、ルートを選択して、サービスの品質を選択して、入力して、終了、およびファブリックを移動する際は、トラフィックを転送に使用されます。



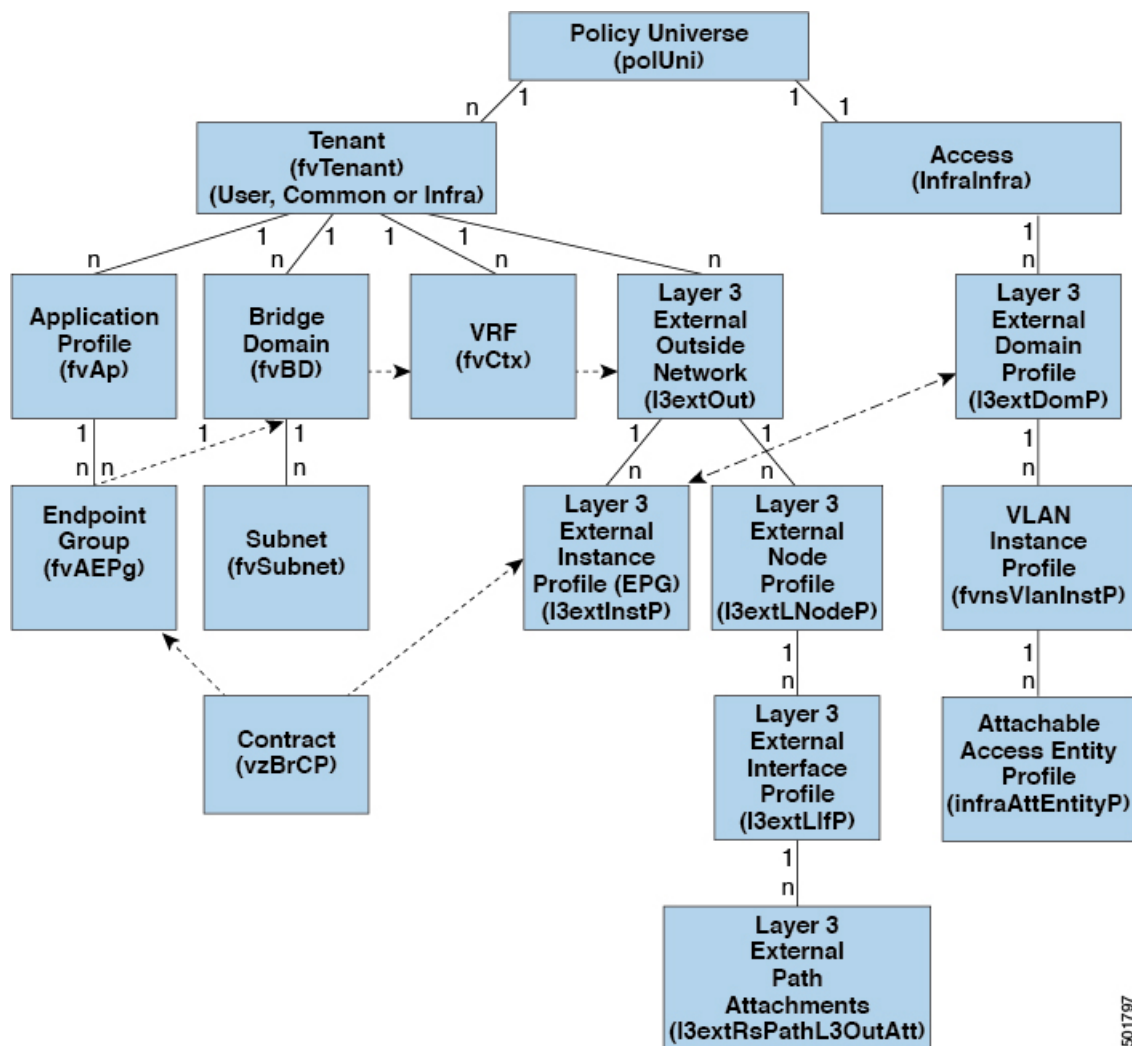
(注) ガイドラインとの設定と接続の外部レイヤ 3 を維持するための注意事項は、次を参照してください。 [外部ネットワークへの接続をルーティングするための注意事項 \(184 ページ\)](#)。

L3Outs の種類についての詳細は、[外部レイヤ 3 Outside 接続タイプ \(186 ページ\)](#) を参照してください。

外部ネットワークへのルーテッド接続のためのレイヤ 3 Out

外部ネットワークへのルーテッド接続は、次の図の階層で示すようにファブリック アクセス (infraInfra) 外部ルーテッド ドメイン (l3extDomP) をレイヤ 3 外部外側ネットワーク (l3extOut) のテナント レイヤ 3 外部インスタンス プロファイル (l3extInstP または外部 EPG) に関連付けることによって有効になります。

図 15: レイヤ 3 外部接続のポリシー モデル



501797

レイヤ 3 外部アウトサイドネットワーク (l3extOut オブジェクト) には、ルーティング プロトコルのオプション (BGP、OSPF、または EIGRP またはサポートされている組み合わせ) およびスイッチとインターフェイス固有の設定が含まれています。l3extOut にルーティング プロトコル (たとえば、関連する仮想ルーティングおよび転送 (VRF) およびエリア ID を含む OSPF) が含まれる一方で、レイヤ 3 外部インターフェイスのプロファイルには必要な OSPF インターフェイスの詳細が含まれます。いずれも OSPF のイネーブル化に必要です。

l3extInstP EPG は、コントラクトを通してテナント EPG に外部ネットワークを公開します。たとえば、Web サーバのグループを含むテナント EPG は、l3extOut に含まれるネットワーク設定に応じてコントラクトを介して l3extInstP EPG と通信できます。外部ネットワーク設定は、ノードを L3 外部ノードプロファイルに関連付けることで複数のノードに容易に再利用できます。同じプロファイルを使用する複数のノードをフェールオーバーやロードバランシングのために設定できます。ノードを複数の l3extOuts に追加することで、l3extOuts に関連付けられている VRF がノードでも展開されます。拡張性に関する情報については、現行の「*Verified Scalability Guide for Cisco ACI*」を参照してください。

外部ネットワークへの接続をルーティングするための注意事項

レイヤ 3 外部接続を作成し、維持する際には、次のガイドラインを使用してください。

トピック	注意またはガイドライン
入力ベース ポリシーの適用	Cisco APIC リリース 1.2(1) 以降、入力ベース ポリシーの適用により、出入力両方向でレイヤ 3 アウトサイド (L3Out) トラフィックにポリシー適用を定義できます。デフォルトでは入力になっています。リリース 1.2(1) 以降にアップグレード中、既存の L3Out 設定が出力に設定され、動作が既存の設定と一致します。特別なアップグレードのシーケンスは必要ありません。アップグレード後、グローバルプロパティ値を入力に変更します。変更されると、システムがルールとプレフィックスエントリを再プログラミングします。規則は出力リーフから削除され、入力リーフ上に既存の規則がない場合は、入力リーフ上にインストールされます。既存の設定がない場合、Actrl プレフィックス エントリが入力リーフ上にインストールされます。ダイレクト サーバリターン (DSR) および属性 EPG には入力ベースのポリシー適用が必要です。vzAny と禁止コントラクトは、入力ベースのポリシー適用を契約無視します。入力には中継規則が適用されます。
L3Outs によるブリッジ ドメイン	テナントのブリッジドメインには、共通テナントでプロビジョニングされている l3extOut によってアドバタイズされたパブリック サブネットを含めることができます。
BGP 最大プレフィックス制限	Cisco APIC リリース 1.2 (1x) 以降、BGP l3extOut 接続のテナントポリシーは、最大プレフィックス制限を使用して設定できます。これにより、ピアから受信されるルートプレフィックスの数をモニタし、制限することができます。最大プレフィックス制限を超えると、ログエントリが記録され、さらにプレフィックスが拒否されます。カウントが一定の間隔でしきい値を下回る場合、接続を再起動することができますが、そうしない場合接続がシャットダウンします。一度に 1 つのオプションだけを使用できます。デフォルト設定では 20,000 プレフィックスに制限され、その後は新しいプレフィックスは拒否されます。拒否オプションが導入されると、APIC でエラーが発生する前に BGP は設定されている制限よりも 1 つ多くプレフィックスを受け入れます。

トピック	注意またはガイドライン
MTU	Cisco ACI は、IP フラグメンテーションをサポートしていません。したがって、外部ルータへのレイヤ 3 Outside (L3Out) 接続、または Inter-Pod Network (IPN) を介した multipod 接続を設定する場合は、MTU が両側で適切に設定されていることが重要です。ACI、Cisco NX-OS、Cisco IOS などの一部のプラットフォームでは、設定された MTU 値は IP ヘッダーを考慮に入れています（結果として、最大パケットサイズは、ACI で 9216 バイト、NX-OS および IOS で 9000 バイトに設定されます）。ただし、IOS XR などの他のプラットフォームは、パケットヘッダーのを除く MTU 値を設定します（結果として最大パケットサイズは 8986 バイトになります）。 各プラットフォームの適切な MTU 値については、それぞれの設定ガイドを参照してください。 CLI ベースのコマンドを使用して MTU をテストすることを強く推奨します。たとえば、Cisco NX-OS CLI で <code>ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1</code> などのコマンドを使用します。
レイヤ 4 から 7	マルチノードサービス グラフを使用する際、異なる VRF で 2 つの EPG が必須です。これらの機能では、システムはレイヤ 3 検索を実行する必要があるため、EPG が異なる VRF を分離する必要があります。この制限は、レイヤ 2 およびレイヤ 3 の検索に基づいてレガシサービスの挿入に続きます。
L3Outs の QoS	L3Out 用の QoS ポリシーを設定し、L3Out が存在する BL スイッチで適用されるポリシーを有効にするには、次の注意事項に従ってください。 • VRF ポリシー制御の適用方向を 出力 に設定する必要があります。 • VRF ポリシー制御適用の優先度設定を 有効 に設定する必要があります。 • L3Out を使用して EPG 間の通信を制御するコントラクトを設定する際に、コントラクトまたはコントラクトの件名に QoS クラスまたはターゲット DSCP を含めません。

外部レイヤ 3 Outside 接続タイプ

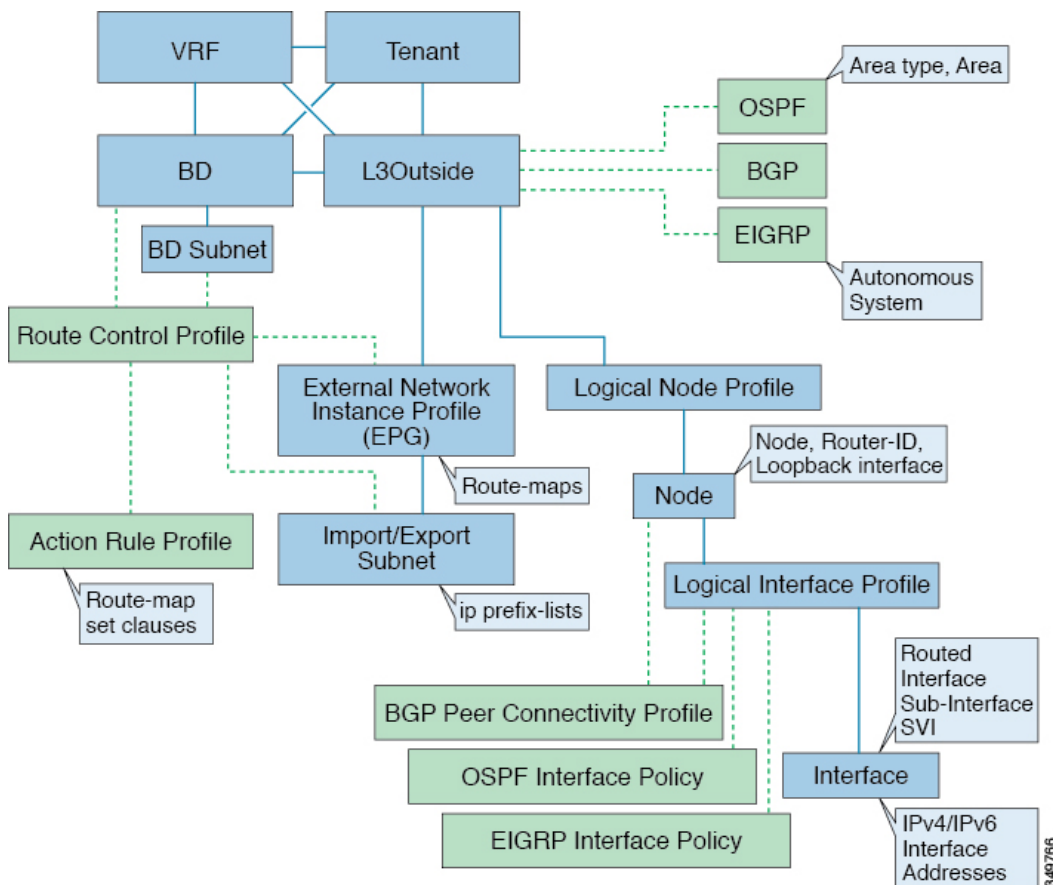
ACI は、以下の外部レイヤ 3 Outside 接続オプションをサポートします。

- スタティック ルーティング (IPv4 および IPv6 でサポート)
- 標準および NSSA エリアの OSPFv2 (IPv4)
- 標準および NSSA エリアの OSPFv3 (IPv6)
- iBGP (IPv4 および IPv6)
- eBGP (IPv4 および IPv6)
- BGP (IPv4 および IPv6)

外部レイヤ 3 Outside 接続は、以下のインターフェイスでサポートされます。

- レイヤ 3 ルーテッド インターフェイス
- 802.1Q タギング対応のサブインターフェイス：サブインターフェイスを使用すると、複数のプライベート ネットワークに対するレイヤ 2 外部接続を提供できます。
- スイッチ仮想インターフェイス (SVI)：SVI インターフェイスを使用すると、レイヤ 2 とレイヤ 3 をサポートする同じ物理インターフェイスをレイヤ 2 外部接続とレイヤ 3 外部接続に使用できます。

図 16: ACI レイヤ 3 管理対象オブジェクト



L3Outside 接続に使用される管理対象オブジェクトは、次のとおりです。

- 外部レイヤ 3 Outside (L3ext) : ルーティングプロトコルオプション (OSPF エリア タイプ、エリア、EIGRP AS、BGP)、プライベートネットワーク、外部物理ドメイン。
- 論理ノードプロファイル : 外部レイヤ 3 Outside 接続に対して 1 つ以上のノードが定義されたプロファイル。ルータ ID とループバック インターフェイス設定はプロファイルで定義されます。



(注) 複数の外部レイヤ 3 Outside 接続間の同じノードには同じルータ ID を使用してください。

- 論理インターフェイスプロファイル : IPv4 および IPv6 インターフェイスの IP インターフェイス設定。これは、ルートインターフェイス、ルーテッドサブインターフェイス、および SVI でサポートされます。SVI は、物理ポート、ポートチャネルまたは VPC で設定できます。
- OSPF インターフェイスポリシー : OSPF のネットワークタイプ、優先順位などの詳細が含まれています。

- EIGRP インターフェイス ポリシー：タイマー、スプリット ホライズン タイマーなどの詳細が含まれています。
- BGP ピア接続プロファイル：ほとんどの BGP ピア設定、リモート AS、ローカル AS、および BGP ピア接続オプションが設定されるプロファイル。BGP ピア接続プロファイルは、ノードプロファイルの下で論理インターフェイス プロファイルまたはループバック インターフェイスに関連付けることができます。これは、BGP ピアリングセッションの update-source 設定を決定します。
- 外部ネットワーク インスタンス プロファイル (EPG) (l3extInstP)：外部 EPG はプレフィックススペースの EPG または InstP とも呼ばれます。インポートおよびエクスポートのルート制御ポリシー、セキュリティ インポート ポリシー、およびコントラクトの関連付けは、このプロファイルで定義されます。単一 L3Out に複数の外部 EPG を設定できます。単一外部レイヤ 3 Outside 接続で別のルートまたはセキュリティ ポリシーが定義されている場合、複数の外部 EPG を使用できます。1 つの外部 EPG または複数の外部 EPG がルートマップにまとめられます。外部 EPG で定義されるインポート/エクスポートサブネットは、ルートマップの IP プレフィックス リストの match 句と関連しています。外部 EPG は、インポートセキュリティサブネットとコントラクトが関連付けられる場所でもあります。これは、この L3out のトラフィックの許可またはドロップに使用されます。
- アクションルールプロファイル：アクションルールプロファイルは、L3Out のルートマップの set 句を定義するために使用されます。サポートされる set 句は、BGP communities (standard および extended)、Tags、Preference、Metric、および Metric type です。
- ルート制御プロファイル：ルート制御プロファイルは、アクションルールプロファイルを参照するために使用されます。これは、アクションルールプロファイルの順序付きプロファイルにすることができます。ルート制御プロファイルは、テナント BD、BD サブネット、外部 EPG、または外部 EPG サブネットで参照できます。

BGP、OSPF、および EIGRP L3Out 用の追加のプロトコル設定が存在します。これらの設定は、GUI の [ACI Protocol Policies] セクションでテナントごとに設定されます。



- (注) 外部 EPG (中継ルーティング ケース) の間でポリシーの適用を設定する際には、エクスポートルート制御、集約エクスポート、および外部のセキュリティのために、デフォルトプレフィックスである 0/0 で 2 番目の外部 EPG (InstP) を設定する必要があります。さらに、優先グループを除外し、中継 InstPs 間で任意の契約（または希望の契約）を使用する必要があります。

NX-OS スタイルの CLI を使用したテナント ネットワークのレイヤ 3 Outside の設定

次の手順では、テナント ネットワークのネットワークの外部レイヤ 3 を設定する方法について説明します。次に、NX-OS CLI を使用してテナント VRF 外部 L3 接続にノードと L3 ポートを配備する例を示します。

この例は、わかりやすくするための手順に分割されます。マージされた例については、[NX-OS スタイル CLI の例：L3Out（192 ページ）](#) を参照してください。

始める前に

- ノード、ポート、機能プロファイル、AEP、レイヤ 3 ドメインを設定します。
- 使用して VLAN ドメイン設定、`vlan ドメイン ドメイン` および `vlan vlan 範囲 コマンド`。
- BGP ルート リフレクタ ポリシーを設定し、ファブリック内でルーテッドを伝達します。

コマンドを使用して、これらの前提条件の例を参照してください。[NX-OS スタイル CLI の例: L3Out の前提条件（192 ページ）](#)。

手順

ステップ 1 テナントおよび VRF を設定します。

この例では VRF `v1` でテナント `t1` を設定します。これらはまだ展開されていません。

例：

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)#
```

ステップ 2 L3Out のノードとインターフェイスを設定します。

この例では設定 VRF `v1` ノード 103 (border リーフ スイッチ) と呼ばれるで `nodep1`、ルータ ID を `11.11.11.103`。インターフェイスの設定も `eth1/3` ルーテッドインターフェイス (レイヤ 3 のポート)、IP アドレスとして `12.12.12.3/24` とレイヤ 3 ドメイン `dom1`。

例：

```
apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 11.11.11.103
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 12.12.12.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

ステップ 3 ルーティング プロトコルを設定します。

この例では、BGP ピアのアドレスを使用して、プライマリのルーティング プロトコルとして BGP を設定 `15.15.15.2` ASN 100 ドルとします。

例：

```

apic1(config)# leaf 103
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

```

ステップ 4 オプション。接続ルーティング プロトコルを設定します。

この例では、定期的なエリア ID と、通信プロトコルとして OSPF を設定 0.0.0.0 、ループバック アドレスと 30.30.30.0 。

例：

```

apic1(config)# leaf 103
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 30.30.30.0
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit

```

ステップ 5 ノード 103 上に外部 EPG を設定します。

この例では、ネットワークで 20.20.20.0/24 外部ネットワークとして設定されている extnw1 。

例：

```

apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epɡ extnw1
apic1(config-tenant-l3ext-epɡ)# vrf member v1
apic1(config-tenant-l3ext-epɡ)# match ip 20.20.20.0/24
apic1(config-tenant-l3ext-epɡ)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epɡ extnw1
apic1(config-leaf-vrf)# exit

```

ステップ 6 オプション。ルート マップを設定します。

この例では、ルート マップ設定 rp1 アウトバウンド方向に BGP ピアの。ルート マップがルートの宛先に一致するのに適用される 200.3.2.0/24 。

また、正常な一致で (ルートには、この範囲が一致する) ルート AS パスアトリビュートが更新され、 200 および 100 。

例：

```

apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 200.3.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config)# leaf 103
apic1(config-leaf)# router bgp 100

```

```
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)#exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# exit
```

ステップ 7 ブリッジ ドメインを追加します。

例 :

```
apicl(config)# tenant t1
apicl(config-tenant)# bridge-domain bd1
apicl(config-tenant-bd)# vrf member v1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# interface bridge-domain bd1
apicl(config-tenant-interface)# ip address 44.44.44.1/24 scope public
apicl(config-tenant-interface)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# route-map rp1
apicl(config-leaf-vrf-route-map)# match bridge-domain bd1 tenant t1
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# exit
```

ステップ 8 ノード 101 で EPG アプリケーションを作成します。

例 :

```
apicl(config)# tenant t1
apicl(config-tenant)# application appl
apicl(config-tenant-app)# epg epg1
apicl(config-tenant-app-epg)# bridge-domain member bd1
apicl(config-tenant-app-epg)# exit
apicl(config-tenant-app)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/3
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# switchport trunk allowed vlan 2011 tenant t1 application appl epg
epg1
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
apicl(config)#
```

ステップ 9 フィルタ (アクセス リスト) と契約を作成します。

例 :

```
apicl(config)# tenant t1
apicl(config-tenant)# access-list http-filter
apicl(config-tenant-acl)# match ip
apicl(config-tenant-acl)# match tcp dest 80
apicl(config-tenant-acl)# exit
apicl(config-tenant)# contract httpCtct
apicl(config-tenant-contract)# scope vrf
apicl(config-tenant-contract)# subject subj1
apicl(config-tenant-contract-subj)# access-group http-filter both
apicl(config-tenant-contract-subj)# exit
apicl(config-tenant-contract)# exit
```

ステップ 10 契約を設定し、Epg に関連付けます。

例 :

```
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# contract provider httpCtrct
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# application appl
apic1(config-tenant-app)# epg epg1
apic1(config-tenant-app-epg)# contract consumer httpCtrct
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
apic1(config)#
```

NX-OS スタイル CLI の例: L3Out の前提条件

L3Out を設定する前に、次の手順を実行します。

1. VLAN ドメインを設定します。

```
apic1# configure
apic1(config)# vlan-domain dom1
apic1(config-vlan)# vlan 1024-2048
apic1(config-vlan)# exit
```

2. BGP ルート リフレクタを設定します:

```
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 100
apic1(config-bgp-fabric)# route-reflector spine 104,105
```

NX-OS スタイル CLI の例 : L3Out

次の例は、L3Out を設定する手順のマージバージョン NX-OS スタイル CLI を使用します。
L3Out を設定する前に、次の前提条件を設定します。

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 11.11.11.103
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 12.12.12.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
```



```
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant t1 vrf v1
apicl(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 30.30.30.0
apicl(config-leaf-ospf-vrf)# exit
apicl(config-leaf-ospf)# exit
apicl(config-leaf)# exit
apicl(config)# tenant t1
apicl(config-tenant)# external-l3 epg extnw1
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# match ip 20.20.20.0/24
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 103
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# external-l3 epg extnw1
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# template route group match-rule1 tenant t1
apicl(config-route-group)# ip prefix permit 200.3.2.0/24
apicl(config-route-group)# exit
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# route-map rp1
apicl(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# router bgp 100
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# exit
apicl(config)# tenant t1
apicl(config-tenant)# bridge-domain bd1
apicl(config-tenant-bd)# vrf member v1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# interface bridge-domain bd1
apicl(config-tenant-interface)# ip address 44.44.44.1/24 scope public
apicl(config-tenant-interface)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# route-map map1
apicl(config-leaf-vrf-route-map)# match bridge-domain bd1 tenant t1
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# exit
apicl(config)# tenant t1
apicl(config-tenant)# application app1
apicl(config-tenant-app)# epg epg1
apicl(config-tenant-app-epg)# bridge-domain member bd1
apicl(config-tenant-app-epg)# exit
apicl(config-tenant-app)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/3
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# switchport trunk allowed vlan 2011 tenant t1 application app1 epg
epg1
```

```

apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# tenant t1
apic1(config-tenant)# access-list http-filter
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# exit
apic1(config-tenant)# contract httpCtrct
apic1(config-tenant-contract)# scope vrf
apic1(config-tenant-contract)# subject subj1
apic1(config-tenant-contract-subj)# access-group http-filter both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# contract provider httpCtrct
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# application appl
apic1(config-tenant-app)# epg epg1
apic1(config-tenant-app-epg)# contract consumer httpCtrct
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
apic1(config)#

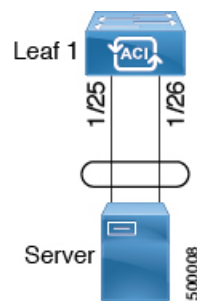
```

レイヤ3 ルーティングとサブインターフェイス ポート チャンネル

レイヤ3 ポート チャンネルについて

以前、Cisco APIC ではレイヤ2 ポート チャンネルのみサポートしていました。リリース 3.2(1) より、Cisco APIC ではレイヤ3 ポート チャンネルもサポートしています。

図 17: スイッチ ポート チャンネル設定



ポート チャンネルの **NX-OS** は、**CLI** を使用してをルーテッド レイヤ3 の設定

この手順では、レイヤ3 ルーテッドポート チャンネルを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apicl# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id 例 : <code>apicl(config)# leaf 101</code>	リーフ スイッチまたはリーフ スイッチ の設定を指定します。 <i>Node-id</i> は形式 <i>node-id1-node-id2</i> の単一ノード ID または ID の範囲となる可能性があり、設定 が適用されます。
ステップ 3	interface port-channel channel-name 例 : <code>apicl(config-leaf)# interface port-channel po1</code>	指定したポート チャネルのインターフェイス コンフィギュレーション モードを開始します。
ステップ 4	no switchport 例 : <code>apicl(config-leaf-if)# no switchport</code>	レイヤ 3 インターフェイスを可能になります。
ステップ 5	vrf member vrf-name tenant tenant-name 例 : <code>apicl(config-leaf-if)# vrf member v1 tenant t1</code>	この仮想ルーティングおよび転送(VRF) インスタンスと L3 ポリシー、外部には、このポートチャネルを関連付けます場所。 • <i>Vrf-name</i> は VRF 名です。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • テナント名 は、テナント名です。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
ステップ 6	vlan-domain member vlan-domain-name 例 : <code>apicl(config-leaf-if)# vlan-domain member dom1</code>	以前に設定された VLAN ドメインには、ポートチャネルのテンプレートを関連付けます。
ステップ 7	ip address ip-address / subnet-mask 例 : <code>apicl(config-leaf-if)# ip address 10.1.1.1/24</code>	指定した インターフェイスの IP アドレスとサブネット マスクを設定します。

	コマンドまたはアクション	目的
ステップ 8	ipv6 address sub-bits/prefix-length preferred 例 : <pre>apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred</pre>	IPv6 の一般的なプレフィックスに基づいて IPv6 アドレスを設定し、インターフェイスにおける IPv6 処理をイネーブルにします。場所 : • <i>sub-bits</i> 引数は、<i>prefix-name</i> 引数で指定された一般的なプレフィックスによって提供されるプレフィックスに連結する、アドレスのサブプレフィックスビットおよびホスト ビットです。sub-bits 引数は、RFC 2373 に記載された形式で指定する必要があります。この形式では、アドレスは、16 進数値を 16 ビット単位でコロンで区切って指定します。 • <i>Prefix-length</i> は IPv6 プレフィックスの長さです。プレフィックス (アドレスのネットワーク部分) を構成するアドレスの上位連続ビット数を示す 10 進値です。10 進数値の前にスラッシュ記号が必要です。
ステップ 9	ipv6 link-local ipv6-link-local-address 例 : <pre>apic1(config-leaf-if)# ipv6 link-local fe80::1</pre>	インターフェイスに IPv6 リンクローカルアドレスを設定します。
ステップ 10	mac-address mac-address 例 : <pre>apic1(config-leaf-if)# mac-address 00:44:55:66:55::01</pre>	インターフェイス MAC アドレスを手動で設定します。
ステップ 11	mtu mtu-value 例 : <pre>apic1(config-leaf-if)# mtu 1500</pre>	このサービス クラスの MTU を設定します

例

この例では、基本レイヤ 3 ポートチャネルを設定する方法を示します。

```
apic1# configure
```

```

apicl(config)# leaf 101
apicl(config-leaf)# interface port-channel po1
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vrf member v1 tenant t1
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# ip address 10.1.1.1/24
apicl(config-leaf-if)# ipv6 address 2001::1/64 preferred
apicl(config-leaf-if)# ipv6 link-local fe80::1
apicl(config-leaf-if)# mac-address 00:44:55:66:55::01
apicl(config-leaf-if)# mtu 1500

```

NX-OS CLI を使用したレイヤ 3 サブインターフェイス ポート チャンネルの設定

この手順では、レイヤ 3 サブインターフェイス ポート チャンネルを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id 例 : apicl(config)# leaf 101	リーフ スイッチまたはリーフ スイッチの設定を指定します。 <i>Node-id</i> は形式 <i>node-id1-node-id2</i> の単一ノード ID または ID の範囲となる可能性があり、設定が適用されます。
ステップ 3	vrf member vrf-name tenant tenant-name 例 : apicl(config-leaf-if)# vrf member v1 tenant t1	この仮想ルーティングおよび転送(VRF) インスタンスと L3 アウトサイド ポリシーにポート チャンネルを関連付けます。場所 : • <i>Vrf-name</i> は VRF 名です。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • テナント名 は、テナント名です。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

	コマンドまたはアクション	目的
ステップ 4	vlan-domain member <i>vlan-domain-name</i> 例 : <pre>apic1(config-leaf-if)# vlan-domain member dom1</pre>	以前に設定された VLAN ドメインには、ポートチャネルのテンプレートを関連付けます。
ステップ 5	ip address <i>ip-address / subnet-mask</i> 例 : <pre>apic1(config-leaf-if)# ip address 10.1.1.1/24</pre>	指定した インターフェイスの IP アドレスとサブネット マスクを設定します。
ステップ 6	ipv6 address <i>sub-bits/prefix-length preferred</i> 例 : <pre>apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred</pre>	IPv6 の一般的なプレフィックスに基づいて IPv6 アドレスを設定し、インターフェイスにおける IPv6 処理をイネーブルにします。場所 : • <i>sub-bits</i> 引数は、<i>prefix-name</i> 引数で指定された一般的なプレフィックスによって提供されるプレフィックスに連結する、アドレスのサブプレフィックスビットおよびホスト ビットです。<i>sub-bits</i> 引数は、RFC 2373 に記載された形式で指定する必要があります。この形式では、アドレスは、16 進数値を 16 ビット単位でコロンで区切って指定します。 • <i>Prefix-length</i> は IPv6 プレフィックスの長さです。プレフィックス (アドレスのネットワーク部分) を構成するアドレスの上位連続ビット数を示す 10 進値です。10 進数値の前にスラッシュ記号が必要です。
ステップ 7	ipv6 link-local <i>ipv6-link-local-address</i> 例 : <pre>apic1(config-leaf-if)# ipv6 link-local fe80::1</pre>	インターフェイスに IPv6 リンクローカルアドレスを設定します。
ステップ 8	mac-address <i>mac-address</i> 例 : <pre>apic1(config-leaf-if)# mac-address 00:44:55:66:55::01</pre>	インターフェイス MAC アドレスを手動で設定します。

	コマンドまたはアクション	目的
ステップ 9	mtu <i>mtu-value</i> 例 : <code>apic1(config-leaf-if) # mtu 1500</code>	このサービス クラスの MTU を設定します
ステップ 10	exit 例 : <code>apic1(config-leaf-if) # exit</code>	設定モードに戻ります。
ステップ 11	interface port-channel <i>channel-name</i> 例 : <code>apic1(config-leaf) # interface port-channel po1</code>	指定したポート チャンネルのインターフェイス コンフィギュレーション モードを開始します。
ステップ 12	vlan-domain member <i>vlan-domain-name</i> 例 : <code>apic1(config-leaf-if) # vlan-domain member dom1</code>	以前に設定された VLAN ドメインには、ポートチャンネルのテンプレートを関連付けます。
ステップ 13	exit 例 : <code>apic1(config-leaf-if) # exit</code>	設定モードに戻ります。
ステップ 14	interface port-channel <i>channel-name.number</i> 例 : <code>apic1(config-leaf) # interface port-channel po1.2001</code>	指定したサブインターフェイス ポートチャンネルのインターフェイス設定モードを開始します。
ステップ 15	vrf member <i>vrf-name</i> tenant <i>tenant-name</i> 例 : <code>apic1(config-leaf-if) # vrf member v1 tenant t1</code>	この仮想ルーティングおよび転送 (VRF) インスタンスと L3 アウトサイド ポリシーにポート チャンネルを関連付けます。場所 : • <i>Vrf-name</i> は VRF 名です。32 文字以内の英数字のストリング (大文字と小文字を区別) で指定します。 • テナント名 は、テナント名です。32 文字以内の英数字のストリング (大文字と小文字を区別) で指定します。
ステップ 16	exit 例 :	設定モードに戻ります。

NX-OS CLI を使用したレイヤ 3 ポート チャンネルにポートを追加する

	コマンドまたはアクション	目的
	apic1(config-leaf-if) # exit	

例

この例では、基本的なレイヤ 3 サブインターフェイス ポートチャンネルを設定する方法を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 2001
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member v1 tenant t1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
apic1(config-leaf-if)# ipv6 link-local fe80::1
apic1(config-leaf-if)# mac-address 00:44:55:66:55::01
apic1(config-leaf-if)# mtu 1500
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel po1.2001
apic1(config-leaf-if)# vrf member v1 tenant t1
apic1(config-leaf-if)# exit
```

NX-OS CLI を使用したレイヤ 3 ポート チャンネルにポートを追加する

この手順では、以前に設定したレイヤ 3 ポート チャンネルにポートを追加します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	リーフ スイッチまたはリーフ スイッチの設定を指定します。 <i>Node-id</i> は形式 <i>node-id1-node-id2</i> の単一ノード ID または ID の範囲となる可能性があり、設定が適用されます。

	コマンドまたはアクション	目的
ステップ 3	interface Ethernet <i>slot/port</i> 例 : apic1(config-leaf) # interface Ethernet 1/1-2	設定するインターフェイスのインターフェイス コンフィギュレーション モードを開始します。
ステップ 4	channel-group チャンネル名 例 : apic1(config-leaf-if) # channel-group p01	チャンネル グループでポートを設定します。

例

この例では、ポートをレイヤ 3 にポートチャンネルを追加する方法を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf) # interface Ethernet 1/1-2
apic1(config-leaf-if) # channel-group p01
```

レイヤ 3 アウトからレイヤ 3 アウト内部 VRF への漏洩

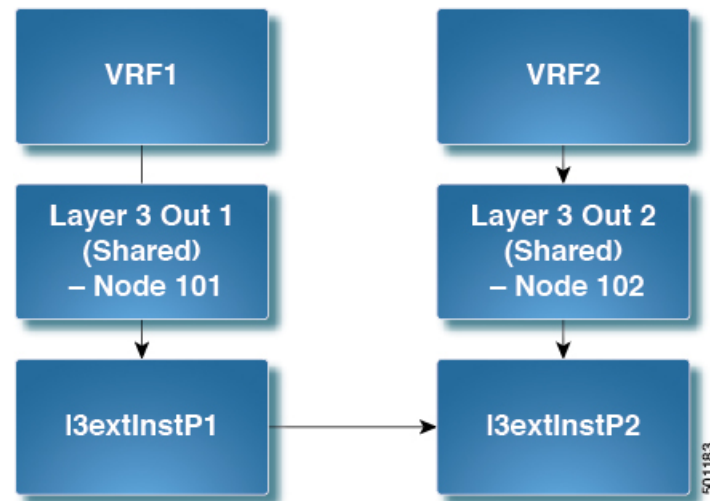
Cisco APIC リリース 2.2(2e) から、2つの異なる VRF に 2 個のレイヤ 3 アウトがある場合、VRF 内部の漏洩がサポートされています。

この機能を稼働するには、次の条件を満たす必要があります。

- 2 個のレイヤ 3 アウト間にはコントラクトが必要です。
- レイヤ 3 アウトの接続したり移行したりするサブネットのルートは、コントラクトを適用し（L3Out-L3Out および L3Out-EPG）、VRF 間の動的または静的ルートを漏洩させることなく漏洩します。
- 動的または静的ルートは、コントラクトを適用し（L3Out-L3Out および L3Out-EPG）、VRF 間で直接接続したり移行したりするルートをアドバタイズすることなく漏洩します。
- 異なる VRF の共有のレイヤ 3 アウトは相互に通信できます。
- 2 個のレイヤ 3 アウトは異なる 2 個の VRF に存在し、正常にルートを交換できます。
- この強化は、アプリケーション EPG およびレイヤ 3 アウト内部 VRF 間の通信と同じです。唯一の違いは、アプリケーション EPG ではなく別のレイヤ 3 アウトが存在します。したがってこの状況では、コントラクトは 2 個のレイヤ 3 アウト間で記録されます。

次の図では、共有サブネットによる 2 個のレイヤ 3 アウトが存在します。両方の VRF でレイヤ 3 外部インスタンス プロファイル (l3extInstP) 間のコントラクトがあります。この場合、VRF 1 の共有レイヤ 3 アウトは VRF 2 の共有レイヤ 3 と通信できます。

図 18: 2 個の VRF 間で通信する共有レイヤ 3 アウト



NX-OS スタイル CLI を使用して共有 レイヤ 3 VRF 内リークを設定する - 名前が付けられた例

手順

	コマンドまたはアクション	目的
ステップ 1	コンフィギュレーション モードを開始します。 例 : <pre>apic1# configure</pre>	
ステップ 2	プロバイダー レイヤ 3 を設定します。 例 : <pre>apic1(config)# tenant t1_provider apic1(config-tenant)# external-l3 ep l3extInstP-1 l3out T0-ol-L3OUT-1 apic1(config-tenant-l3ext-epg)# vrf member VRF1 apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24 shared apic1(config-tenant-l3ext-epg)# contract provider vzBrCP-1 apic1(config-tenant-l3ext-epg)# exit apic1(config-tenant)# exit apic1(config)# leaf 101 apic1(config-leaf)# vrf context tenant</pre>	

	コマンドまたはアクション	目的
	<pre> t1_provider vrf VRF1 l3out T0-o1-L3OUT-1 apic1(config-leaf-vrf) # route-map T0-o1-L3OUT-1_shared apic1(config-leaf-vrf-route-map) # ip prefix-list l3extInstP-1 permit 192.168.2.0/24 apic1(config-leaf-vrf-route-map) # match prefix-list l3extInstP-1 apic1(config-leaf-vrf-route-map-match) # exit apic1(config-leaf-vrf-route-map) # exit apic1(config-leaf-vrf) # exit apic1(config-leaf) # exit </pre>	
ステップ 3	レイヤ 3 Out コンシューマを設定します。 例 : <pre> apic1(config) # tenant t1_consumer apic1(config-tenant) # external-l3 epg l3extInstP-2 l3out T0-o1-L3OUT-1 apic1(config-tenant-l3ext-epg) # vrf member VRF2 apic1(config-tenant-l3ext-epg) # match ip 199.16.2.0/24 shared apic1(config-tenant-l3ext-epg) # contract consumer vzBrCP-1 imported apic1(config-tenant-l3ext-epg) # exit apic1(config-tenant) # exit apic1(config) # leaf 101 apic1(config-leaf) # vrf context tenant t1_consumer vrf VRF2 l3out T0-o1-L3OUT-1 apic1(config-leaf-vrf) # route-map T0-o1-L3OUT-1_shared apic1(config-leaf-vrf-route-map) # ip prefix-list l3extInstP-2 permit 199.16.2.0/24 apic1(config-leaf-vrf-route-map) # match prefix-list l3extInstP-2 apic1(config-leaf-vrf-route-map-match) # exit apic1(config-leaf-vrf-route-map) # exit apic1(config-leaf-vrf) # exit apic1(config-leaf) # exit apic1(config) # </pre>	

NX-OS Style CLI を使用した共有レイヤ 3 VRF 間リークの設定：名前を付けた例

手順

	コマンドまたはアクション	目的
ステップ 1	コンフィギュレーション モードを開始します。 例： <pre>apicl# configure</pre>	
ステップ 2	プロバイダ テナントおよび VRF の設定 例： <pre>apicl(config)# tenant t1_provider apicl(config-tenant)# vrf context VRF1 apicl(config-tenant-vrf)# exit apicl(config-tenant)# exit</pre>	
ステップ 3	コンシューマ テナントおよび VRF の設定 例： <pre>apicl(config)# tenant t1_consumer apicl(config-tenant)# vrf context VRF2 apicl(config-tenant-vrf)# exit apicl(config-tenant)# exit</pre>	
ステップ 4	コントラクトの設定 例： <pre>apicl(config)# tenant t1_provider apicl(config-tenant)# contract vzBrCP-1 apicl(config-tenant-contract)# type permit apicl(config-tenant-contract)# scope exportable apicl(config-tenant-contract)# export to tenant t1_consumer apicl(config-tenant-contract)# exit</pre>	
ステップ 5	プロバイダ外部レイヤ 3 EPG の設定 例： <pre>apicl(config-tenant)# external-l3 epg l3extInstP-1 apicl(config-tenant-l3ext-epg)# vrf member VRF1 apicl(config-tenant-l3ext-epg)# match ip 192.168.2.0/24 shared apicl(config-tenant-l3ext-epg)# contract provider vzBrCP-1 apicl(config-tenant-l3ext-epg)# exit apicl(config-tenant)# exit</pre>	

	コマンドまたはアクション	目的
ステップ 6	プロバイダ エクスポート マップの設定 例： <pre> apic1(config)# leaf 101 apic1(config-leaf)# vrf context tenant t1_provider vrf VRF1 apic1(config-leaf-vrf)# route-map map1 apic1(config-leaf-vrf-route-map)# ip prefix-list p1 permit 192.168.2.0/24 apic1(config-leaf-vrf-route-map)# match prefix-list p1 apic1(config-leaf-vrf-route-map-match)# exit apic1(config-leaf-vrf-route-map)# exit apic1(config-leaf-vrf)# export map map1 apic1(config-leaf-vrf)# exit apic1(config-leaf)# exit </pre>	
ステップ 7	コンシューマ外部レイヤ 3 EPG の設定 例： <pre> apic1(config)# tenant t1_consumer apic1(config-tenant)# external-l3 ep l3extInstP-2 apic1(config-tenant-l3ext-epg)# vrf member VRF2 apic1(config-tenant-l3ext-epg)# match ip 199.16.2.0/24 shared apic1(config-tenant-l3ext-epg)# contract consumer vzBrCP-1 imported apic1(config-tenant-l3ext-epg)# exit apic1(config-tenant)# exit </pre>	
ステップ 8	コンシューマ エクスポート マップの設定 例： <pre> apic1(config)# leaf 101 apic1(config-leaf)# vrf context tenant t1_consumer vrf VRF2 apic1(config-leaf-vrf)# route-map map2 apic1(config-leaf-vrf-route-map)# ip prefix-list p2 permit 199.16.2.0/24 apic1(config-leaf-vrf-route-map)# match prefix-list p2 apic1(config-leaf-vrf-route-map-match)# exit apic1(config-leaf-vrf-route-map)# exit apic1(config-leaf-vrf)# export map map2 apic1(config-leaf-vrf)# exit apic1(config-leaf)# exit apic1(config)# </pre>	

SVI 外部カプセル化の範囲について

レイヤ 3 アウト設定のコンテキストでは、スイッチ仮想インターフェイス (SVI) は ACI リーフスイッチとルータ間に接続性を提供するように設定されます。

デフォルトで単一のレイヤ 3 アウトが SVI インターフェイスで設定されている場合、VLAN のカプセル化はファブリック内の複数のノードに範囲が及びます。これは、図で示されるように SVI インターフェイスが同じ外部カプセル化 (SVI) を使用する限り、レイヤ 3 アウト SVI が展開されているファブリックで、ACI ファブリックがすべてのノード上に同じブリッジドメイン (VXLAN VN) を設定するため発生します。

ただし、異なるレイヤ 3 アウトが展開されている場合、同じ外部カプセル化 (SVI) を使用している場合でも ACI ファブリックは異なるブリッジドメインを使用します。

図 19: ローカル範囲のカプセル化と 1 個のレイヤ 3 アウト

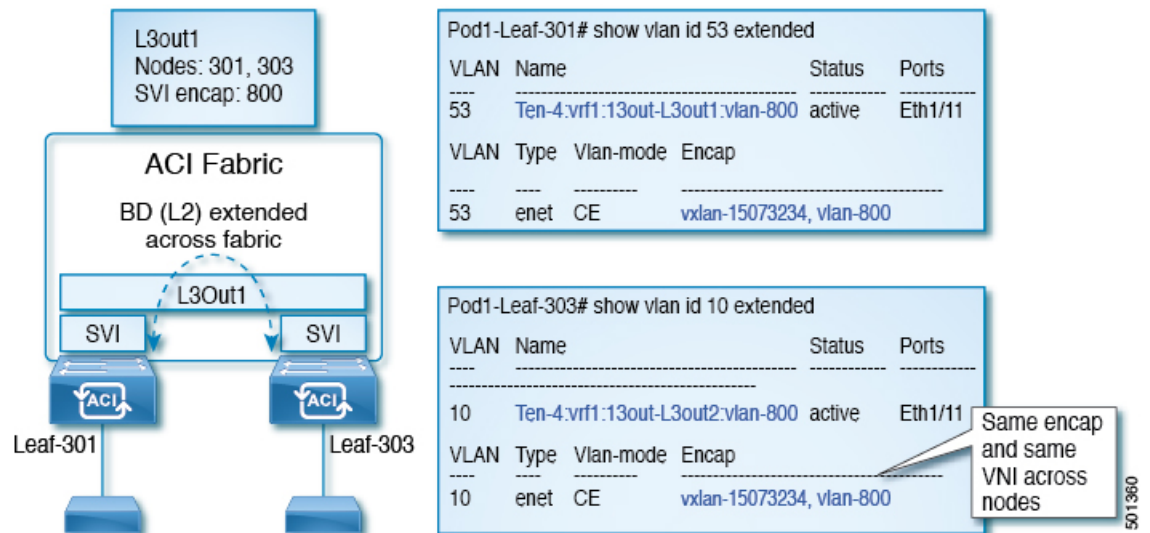
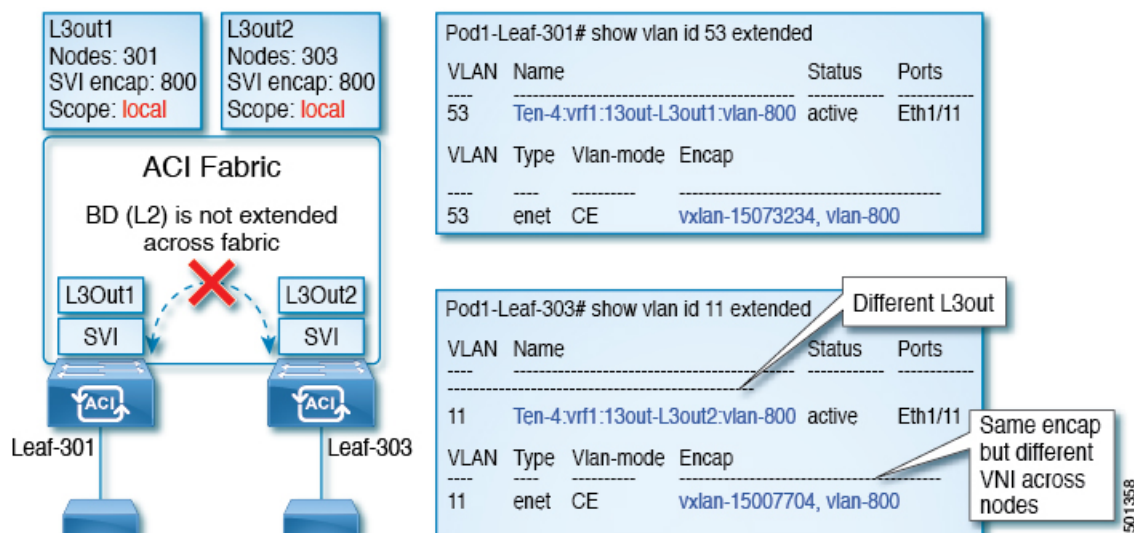


図 20: ローカル範囲のカプセル化と 2 個のレイヤ 3 アウト

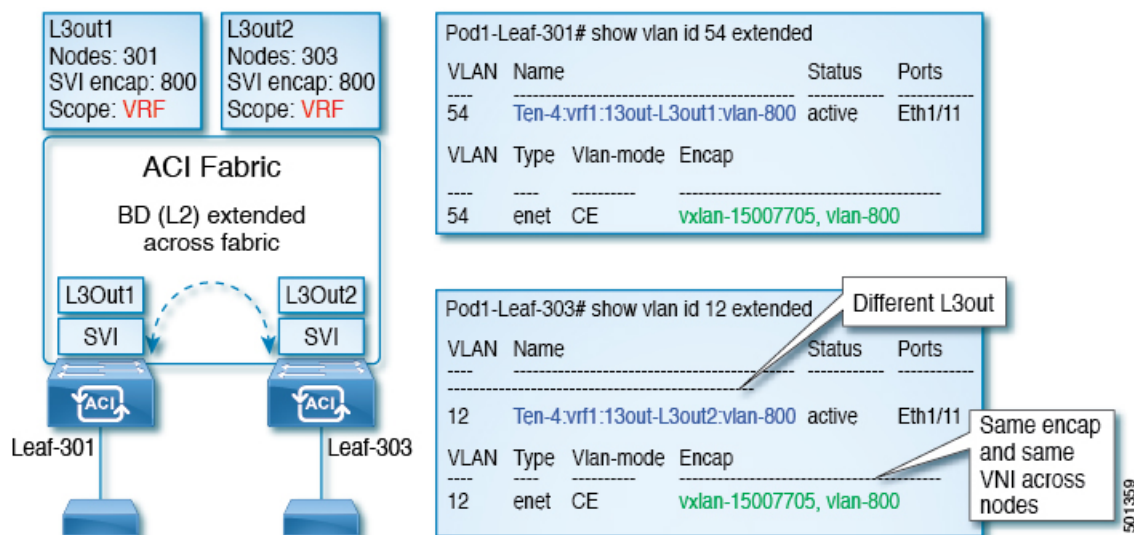


Cisco APIC リリース 2.3 以降、同じ外部カプセル化（SVI）を使用して、2 個以上のレイヤ 3 アウトを展開する場合の動作を選択できるようになりました。

カプセル化の範囲は、ローカルまたは VRF として設定できます。

- ローカル範囲（デフォルト）：例の動作が「ローカル範囲のカプセル化および 2 個のレイヤ 3 アウト」というタイトルの図に表示されます。
- VRF 範囲：ACI ファブリックが、同じ外部カプセル化（SVI）が展開されているすべてのノードとレイヤ 3 アウト上で同じブリッジドメイン（VXLAN VNI）を設定します。「VRF 範囲のカプセル化および 2 個のレイヤ 3 アウト」というタイトルの図の例を参照してください。

図 21: VRF 範囲のカプセル化および 2 個のレイヤ 3 アウト



カプセル化スコープ構文

レイヤ 3 Out プロファイルで使用されるカプセル化の範囲を設定するためのオプションは次のとおりです。

- **Ctx]**: 特定の VLAN のカプセル化の同じ VRF に、すべてのレイヤ 3 が記録されるで同じ外部 SVI。これはグローバル値です。
- **ローカル** : レイヤ 3 Out ごとの一意の外部 SVI。これはデフォルト値です。

CLI、API、および GUI 構文間のマッピングは次のとおりです。

表 19: カプセル化スコープ構文

CLI	API	GUI
l3out	local	local
vrf	ctx	VRF



(注) カプセル化の範囲を設定する CLI コマンドでは、名前付きのレイヤ 3 アウト設定、VRF が設定されている場合にのみサポートされます。

NX-OS スタイル CLI を使用して、SVI インターフェイスのカプセル化スコープの設定

SVI インターフェイスカプセル化のスコープ設定を次の例表示する手順では、名前付きのレイヤ 3 アウト設定です。

手順

	コマンドまたはアクション	目的
ステップ 1	コンフィギュレーション モードを開始します。 例 : <code>apic1# configure</code>	コンフィギュレーション モードを開始します。
ステップ 2	スイッチ モードを開始します。 例 : <code>apic1(config)# leaf 104</code>	スイッチ モードを開始します。
ステップ 3	VLAN インターフェイスを作成します。 例 :	VLAN インターフェイスを作成します。 VLAN の範囲は 1 ～ 4094 です。

	コマンドまたはアクション	目的
	<code>apic1(config-leaf) # interface vlan 2001</code>	
ステップ 4	カプセル化の範囲を指定します。 例： <code>apic1(config-leaf-if) # encaps scope vrf context</code>	カプセル化の範囲を指定します。
ステップ 5	インターフェイスモードを終了します。 例： <code>apic1(config-leaf-if) # exit</code>	インターフェイスモードを終了します。

SVI 自動状態について



- (注) この機能は、APIC リリース 2.2(3x) リリースおよび APIC リリース 3.1 (1) で使用できます。APIC リリース 3.0(x) ではサポートされていません。

スイッチ仮想インターフェイス (SVI) は、デバイスの VLAN のブリッジング機能とルーティング機能間の論理インターフェイスを表します。SVI は、物理ポート、直接ポートチャネル、仮想ポートチャネルのメンバーを有することができます。SVI 論理インターフェイスは VLAN に関連付けられ、VLAN ポート メンバーシップを有します。

SVI の状態はメンバーに依存しません。Cisco APIC の SVI のデフォルトの自動状態動作は、自動状態の値が無効になっているときに最新の状態になっていることを意味します。これは、インターフェイスが対応する VLAN で動作していない場合、SVI がアクティブであることを意味します。

SVI 自動状態の値を有効に変更する場合、関連する VLAN のポート メンバーに依存します。VLAN インターフェイスが VLAN で複数のポートを有する場合、SVI は VLAN のすべてのポートがダウンするとダウン状態になります。

表 20: SVI 自動状態

SVI 自動状態	SVI 状態の説明
ディセーブル	インターフェイスが対応する VLAN で動作していない場合、SVI がアップ状態であることを意味します。 無効がデフォルトの SVI 自動状態の値です。
イネーブル	SVI は、関連付けられている VLAN のポート メンバによって異なります。VLAN インターフェイスに複数のポートを含む場合、SVI は VLAN のすべてのポートがダウンするとダウン状態になります。

SVI 自動状態の動作のガイドラインと制限事項

次のガイドラインをお読みください。

- SVI の自動状態の動作を有効化または無効化にすると、SVI あたりの自動状態の動作を設定します。これらはグローバル コマンドではありません。

NX-OS スタイル CLI を使用した SVI 自動状態の設定

始める前に

- テナントおよび VRF が設定されています。
- レイヤ 3 アウトが設定されており、レイヤ 3 アウトの論理ノードプロファイルと論理インターフェイス プロファイルが設定されています。

手順

	コマンドまたはアクション	目的
ステップ 1	コンフィギュレーション モードを開始します。 例： apicl# configure	コンフィギュレーション モードを開始します。
ステップ 2	スイッチ モードを開始します。 例： apicl(config)# leaf 104	スイッチ モードを開始します。
ステップ 3	VLAN インターフェイスを作成します。 例： apicl(config-leaf)# interface vlan 2001	VLAN インターフェイスを作成します。 VLAN の範囲は 1 ～ 4094 です。
ステップ 4	SVI 自動状態を有効にします。 例： apicl(config-leaf-if)# autostate	SVI 自動状態を有効にします。 デフォルトで、SVI 自動状態の値は有効ではありません。
ステップ 5	インターフェイスモードを終了します。 例： apicl(config-leaf-if)# exit	インターフェイスモードを終了します。

インターフェイスとスタティック ルーティングの設定

始める前に

テナントおよび VRF を設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーションモードに入ります。
ステップ 2	leaf node-id 例 : <code>apic1(config)# leaf 101</code>	設定するリーフを指定します。
ステップ 3	[no] vrf context tenant tenant-name vrf vrf-name 例 : <code>apic1(config-leaf)# vrf context tenant exampleCorp vrf v1</code>	ノードのテナント VRF を設定します。
ステップ 4	(任意) [no] router-id ipv4-address 例 : <code>apic1(config-leaf-vrf)# router-id 1.2.3.4</code>	VRF で実行されているルーティングプロトコルに対してルータ ID を割り当てます。ルータ ID を割り当てない場合、各リーフ スイッチに固有の ID が内部で生成されます。
ステップ 5	[no] {ip ipv6} route ip-prefix/masklen next-hop-address [preferred] 例 : <code>apic1(config-leaf-vrf)# ip route 21.1.1.1/32 32.1.1.1</code> <code>apic1(config-leaf-vrf)# ipv6 route 5001::1/128 6002::1</code>	VRF の静的ルートの情報を設定します。
ステップ 6	exit 例 : <code>apic1(config-leaf-vrf)# exit</code>	リーフ コンフィギュレーションモードに戻ります。
ステップ 7	interface type 例 : <code>apic1(config-leaf)# interface eth 1/1</code>	外部インターフェイスのポートを指定します。

	コマンドまたはアクション	目的
ステップ 8	vlan-domain member <i>domain-name</i> 例 : <pre>apic1(config-leaf-if)# vlan-domain member dom1</pre>	インターフェイスに VLAN ドメインを割り当てます。The VLAN domain must have already been created using the vlan-domain command in the global configuration mode.
ステップ 9	no switchport 例 : <pre>apic1(config-leaf-if)# no switchport</pre>	インターフェイスをレイヤ 3 インターフェイスとして設定し、設定オプションでレイヤ 3 コマンドを公開します。
ステップ 10	vrf member tenant <i>tenant-name</i> vrf <i>vrf-name</i> 例 : <pre>apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1</pre>	テナント VRF にインターフェイスを接続します。
ステップ 11	[no] {ip ipv6} address <i>ip-prefix/masklen</i> [eui64] [secondary] [preferred] 例 : <pre>apic1(config-leaf-if)# ip address 10.1.1.1/24 apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred</pre>	インターフェイスに IP アドレスを設定します。指定されたアドレスは次のいずれかとして宣言できます。 • preferred—インターフェイスからトラフィックのデフォルト送信元アドレス。 • secondary—インターフェイスのセカンダリ アドレス。 With the optional eui64 keyword, the host can assign itself a 64-bit Extended Unique Identifier (EUI). In this mode, you can also configure ipv6 link-local, mac address, mtu, and other layer 3 properties on the interface.
ステップ 12	[[no]]ip dhcprelayaddress <i>tenant</i><i>tenant-name</i> <i>dhcp-address</i> {application<i>app-name</i> <i>epg</i><i>epg-name</i> <i>external1</i><i>ext1/2</i> <i>qg</i><i>qg-name</i> <i>external3</i><i>ext1/3</i> <i>qg</i><i>qg-name</i>} 例 : <pre>apic(config-leaf-if)# ip dhcp relay address 192.0.20.1 tenant exampleCorp application appl1 epg epg1</pre>	サポートされているすべてのオプションとともに、外部インターフェイスの DHCP リレー アドレスを設定または削除します。

例

次に、外部接続用にレイヤ 3 ポートを導入する例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf)# router-id 1.2.3.4
apic1(config-leaf-vrf)# ip route 21.1.1.1/32 32.1.1.1
apic1(config-leaf-vrf)# ipv6 route 5001::1/128 6002::1 preferred
apic1(config-leaf-vrf)# exit

apic1(config-leaf)# interface eth 1/1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ip address 11.1.1.1/24 secondary
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
apic1(config-leaf-if)# ipv6 link-local fe80::1
apic1(config-leaf-if)# mac-address 00:44:55:66:55::01
apic1(config-leaf-if)# mtu 4470
```

次に、外部接続用にレイヤ 3 サブインターフェイス ポートを設定する例を示します。この例では、サブインターフェイス ID (1/2.100 の「100」) は実際には ID ではなく VLAN のカプセル化です。サブインターフェイスでは、レイヤ 3 ポートでサポートされるすべてのプロパティを同様に利用できます。

```
apic1# configure
apic1(config)# leaf 101
    # SAME VRF CONTEXT CONFIGURATION AS PREVIOUS EXAMPLE

apic1(config-leaf)# interface eth 1/2.100
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1
    # SAME L3 PROPERTIES CONFIGURATION AS PREVIOUS EXAMPLE
```

次に、外部接続用にスイッチ仮想インターフェイス (SVI) を設定する方法を示します。各外部 SVI は SVIID で示されるカプセル化 VLAN によって一意に識別されます。

```
apic1# configure
apic1(config)# leaf 101
    # SAME VRF CONTEXT CONFIGURATION AS PREVIOUS EXAMPLE

apic1(config-leaf)# interface vlan 200
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1
apic1(config-leaf-if)# ip address 13.1.1.1/24

    # HOW TO ATTACH A PORT TO THE EXTERNAL SVI:
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/4
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf)# switchport trunk allowed vlan 10 tenant exampleCorp external-svi

    # HOW TO ATTACH A PORT CHANNEL TO THE EXTERNAL SVI:
apic1(config)# leaf 102
```

```

apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf)# switchport trunk allowed vlan 10 tenant exampleCorp external-svi

# HOW TO ATTACH A VIRTUAL PORT CHANNEL (vPC) TO THE EXTERNAL SVI:
apic1(config)# vpc context leaf 101 102
apic1(config-leaf)# interface vpc vpc103
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf)# switchport trunk allowed vlan 10 tenant exampleCorp external-svi

```



- (注) 外部 SVI は参加ノードごとに設定する必要があります。これにより、同じ SVI でノードごとに異なる IP アドレスを設定することができます。vPC が外部 SVI の一部である場合、参加している vPC ピアごとに SVI を個別に作成する必要があり、SVI ごとに異なる IP アドレスを指定することができます。

OSPF の設定

『Configuring OSPF』

OSPF はエリアで次のいずれかのモードで動作できます。

- OSPF がノードでテナント VRF の主要なルーティングプロトコルとして使用される場合、OSPF は OSPF エリアで設定されたルートマップに定義されているルートをインポートおよびエクスポートします。ルートマップにはエクスポート ルートが含まれます。
- OSPF が BGP の接続プロトコルとして使用される場合、OSPF は BGP セッションの送信元として使用されるループバック アドレスをアドバタイズします。ループバック ID ではなくループバック IP アドレスが使用されることに注意してください。この場合、OSPF に依存している BGP セッションは `update-source` コマンドで同じループバック IP アドレスを使用します。

OSPF と OSPFv3 に個別の設定は必要ありません。ルータ OSPF モードは OSPF で実行されているエリアでは暗黙的に OSPFv2 と OSPFv3 の両方を処理します。

OSPF セッションは、次に示す、リーフのすべてのタイプのレイヤ 3 インターフェイスでサポートされます。

- レイヤ 3 ポート
- サブインターフェイス
- 外部 SVI

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	コンフィギュレーションモードに入ります。
ステップ 2	leaf node-id 例 : apicl(config)# leaf 101	設定するリーフを指定します。
ステップ 3	router ospf default 例 : apicl(config-leaf)# router ospf default	OSPF ルーティングプロセスを作成し、OSPF ポリシー設定を開始します。
ステップ 4	vrf member tenant tenant-name vrf vrf-name 例 : apicl(config-leaf-ospf)# vrf member tenant exampleCorp vrf v100	OSPF セッションで VRF を有効にします。
ステップ 5	(任意) default-information originate [always] 例 : apicl(config-leaf-ospf-vrf)# default-information originate	スイッチでデフォルトルートが生成されます。
ステップ 6	area area-id nssa [no-redistribution] [default-information-originate] 例 : apicl(config-leaf-ospf-vrf)# area 0 nssa	Not-So-Stubby Area (NSSA) を定義します。
ステップ 7	area area-id stub 例 : apicl(config-leaf-ospf-vrf)# area 17 stub	エリアをスタブエリアとして定義します。
ステップ 8	area エリア id default-cost コスト 例 : apicl(config-leaf-ospf-vrf)# area 17 default-cost 20	OSPF のデフォルト エリア コストを 0 ～ 16777215 の値に設定します。
ステップ 9	area area-id route-map map-name out 例 :	アウトバウンドフィルタリングのルートマップを指定します。

	コマンドまたはアクション	目的
	<code>apic1(config-leaf-ospf-vrf)# area 17 route-map ospf-to-eigrp out</code>	
ステップ 10	area area-id loopback loopback-address 例 : <code>apic1(config-leaf-ospf-vrf)# area 17 loopback 192.0.20.11/32</code>	OSPF が BGP の接続プロトコルとして使用される場合、OSPF は BGP セッションの送信元として使用されるループバックアドレスをアドバタイズします。ループバック ID ではなくループバック IP アドレスが使用されることに注意してください。この場合、OSPF に依存している BGP セッションは <code>update-source</code> コマンドで同じループバック IP アドレスを使用します。
ステップ 11	inherit {ipv4 ipv6} ospf vrf-policy policy-name 例 : <code>apic1(config-leaf-ospf-vrf)# inherit ipv4 ospf vrf-policy vrfTemplate2</code>	この VRF で OSPF テンプレート ポリシーを継承します。
ステップ 12	summary-address ip-address 例 : <code>apic1(config-leaf-ospf-vrf)# summary-address 182.1.20.0/24</code>	外部ルート集約を設定します。他のプロトコルから学習した外部ルートのサマリー アドレスを入力します。
ステップ 13	area area-id range address-range cost cost 例 : <code>apic1(config-leaf-ospf-vrf)# area 17 range 192.0.20.0/24 cost 20</code>	エリア間のネットワークを集約するエリア間ルート集約を設定します。 The range is the summary route to be advertised in areas. The cost is a value between 0 and 16777215.
ステップ 14	exit 例 : <code>apic1(config-leaf-ospf-vrf)# exit</code>	OSPF コンフィギュレーション モードに戻ります。
ステップ 15	exit 例 : <code>apic1(config-leaf-ospf)# exit</code>	リーフ コンフィギュレーション モードに戻ります。
ステップ 16	interface slot/port 例 : <code>apic1(config-leaf)# interface eth 1/2</code>	OSPF インターフェイスのポートを指定します。The interface could also be specified as interface slot/port.vlan-id or interface vlanvlan-id .
ステップ 17	{ip ipv6} router ospf default area area-id 例 :	OSPF ルーティングプロセスを作成し、OSPF ポリシー設定を開始します。

	コマンドまたはアクション	目的
	<code>apicl(config-leaf-if) # ip router ospf default area 17</code>	
ステップ 18	{ip ipv6} ospf inherit interface-policy if-policy-name tenant tenant-name 例 : <code>apicl(config-leaf-if) # ip ospf inherit interface-policy ifPolicy3 tenant exampleCorp</code>	このテナントで OSPF インターフェイス テンプレート ポリシーを継承します。
ステップ 19	[no] {ip ipv6} ospf prefix-suppression {enable disable inherit} 例 : <code>apicl(config-leaf-if) # ip ospf prefix-suppression enable</code>	セカンダリ IP アドレスに関連付けられているプレフィックスを除き、特定のインターフェイスに属するすべての IP プレフィックスを OSPF がアドバタイズするのを防ぎます。
ステップ 20	[no] {ip ipv6} ospf passive-interface 例 : <code>apicl(config-leaf-if) # ip ospf passive-interface</code>	インターフェイス上でルーティングが更新されないようにします。
ステップ 21	[no] ip ospf authentication {md5 none simple} 例 : <code>apicl(config-leaf-if) # ip ospf authentication md5</code>	認証タイプを指定します。
ステップ 22	ip ospf authentication-key key 例 : <code>apicl(config-leaf-if) # ip ospf authentication-key c1\$c0123</code>	認証キーを指定します

例

```

apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant exampleCorp vrf v100
apicl(config-leaf-ospf-vrf)# area 0 nssa
apicl(config-leaf-ospf-vrf)# area 17 stub
apicl(config-leaf-ospf-vrf)# area 17 default-cost 20
apicl(config-leaf-ospf-vrf)# area 17 route-map ospf-to-eigrp out
apicl(config-leaf-ospf-vrf)# area 17 loopback 192.0.20.11/32
apicl(config-leaf-ospf-vrf)# inherit ipv4 ospf vrf-policy vrfTemplate2
apicl(config-leaf-ospf-vrf)# summary-address 182.1.20.0/24
apicl(config-leaf-ospf-vrf)# area 17 range 192.0.20.0/24 cost 20
apicl(config-leaf-ospf-vrf)# exit
apicl(config-leaf-ospf)# exit
apicl(config-leaf)# interface eth 1/3

```

```

apic1(config-leaf-if)# ip router ospf default area 17
apic1(config-leaf-if)# ip ospf inherit interface-policy ifPolicy3 tenant exampleCorp
apic1(config-leaf-if)# ip ospf prefix-suppression enable
apic1(config-leaf-if)# ip ospf passive-interface
apic1(config-leaf-if)# ip ospf authentication md5
apic1(config-leaf-if)# ip ospf authentication-key c1$c0123

```

OSPF VRF とインターフェイス テンプレートの作成

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーションモードに入ります。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	template ospf vrf-policy vrf-policy-name tenant tenant-name 例 : apic1(config-leaf)# template ospf vrf-policy vrfTemplate3 tenant exampleCorp	指定したテナント下に OSPF VRF ポリシー テンプレートを作成します。
ステップ 4	timers throttle lsa start-time hold-interval max-time 例 : apic1(config-vrf-policy)# timers throttle lsa 200 10000 45000	リンクステート アドバタイズメント (LSA) の開始間隔、ホールド間隔、および最大間隔を設定します。
ステップ 5	timers lsa-group-pacing seconds 例 : apic1(config-vrf-policy)# timers lsa-group-pacing 240	LSA のグループ化、リフレッシュ、チェックサム、またはエージングを行う間隔を設定します。
ステップ 6	timers lsa-arrival milliseconds 例 : apic1(config-vrf-policy)# timers lsa-arrival 1000	各 LSA の到着間の最小間隔を設定します。
ステップ 7	timers throttle spf spf-start spf-hold spf-max-wait 例 :	LSA の SPF 開始間隔、ホールド間隔、および最大間隔を設定します。

	コマンドまたはアクション	目的
	<code>apicl(config-vrf-policy)# timers throttle spf 5 1000 90000</code>	
ステップ 8	auto-cost reference-bandwidth <i>bandwidth</i> 例 : <code>apicl(config-vrf-policy)# auto-cost reference-bandwidth 1000</code>	OSPF ポリシー帯域幅参照を設定します (Mbps)。
ステップ 9	distance <i>distance</i> 例 : <code>apicl(config-vrf-policy)# distance 200</code>	OSPF ポリシー優先アドミニストレーティブ ディスタンスを設定します。
ステップ 10	maximum-paths <i>max-paths</i> 例 : <code>apicl(config-vrf-policy)# maximum-paths 8</code>	OSPF がルーティング テーブル内にインストールできるパラレルルート of の最大数を設定します。範囲は 1 ～ 16 ルートです。
ステップ 11	graceful-restart helper-disable 例 : <code>apicl(config-vrf-policy)# graceful-restart helper-disable</code>	グレースフルリスタートヘルパー モードをディセーブルにします。
ステップ 12	prefix-suppression 例 : <code>apicl(config-vrf-policy)# prefix-suppression</code>	ループバック、セカンダリ IP アドレス およびパッシブ インターフェイスに関連付けられているプレフィックスを除き、すべての IP プレフィックスを OSPF がアドバタイズするのを防ぎます。
ステップ 13	name-lookup 例 : <code>apicl(config-vrf-policy)# name-lookup</code>	DNS 名をルックアップするように OSPF を設定します。
ステップ 14	exit 例 : <code>apicl(config-vrf-policy)# exit</code>	リーフ コンフィギュレーション モードに戻ります。
ステップ 15	template ospf interface-policy <i>if-policy-name</i> <i>tenant</i> <i>tenant-name</i> 例 : <code>apicl(config-leaf)# template ospf interface-policy ifTemplate5 tenant exampleCorp</code>	指定したテナント下に OSPF インターフェイス ポリシー テンプレートを作成します。

	コマンドまたはアクション	目的
ステップ 16	[no] advertise-subnet 例 : apic1(config-interface-policy) # advertise-subnet	/32 の代わりにプライマリ IP アドレスサブネットマスクをアドバタイズします。
ステップ 17	[no] cost if-cost 例 : apic1(config-interface-policy) # cost 300	インターフェイスの OSPF コストを設定します。指定できる範囲は 0 ～ 65535 です。
ステップ 18	[no] dead-interval seconds 例 : apic1(config-interface-policy) # dead-interval 60	ルータがダウンしたとネイバーが宣言するまでの間、hello パケットを非表示にしなければならない間隔を秒単位で設定します。指定できる範囲は 1 ～ 65535 秒です。
ステップ 19	[no] hello-interval seconds 例 : apic1(config-interface-policy) # hello-interval 10	hello パケット間の間隔を秒単位で指定します。指定できる範囲は 1 ～ 65535 秒です。
ステップ 20	[no] mtu-ignore 例 : apic1(config-interface-policy) # mtu-ignore	インターフェイスで MTU 不一致の検出を無効にします。
ステップ 21	[no] network {bcast p2p unspecified} 例 : apic1(config-interface-policy) # network p2p	OSPF インターフェイスポリシーのネットワークタイプをブロードキャストまたはポイントツーポイントのいずれかに設定します。
ステップ 22	[no] passive-interface 例 : apic1(config-interface-policy) # passive-interface	インターフェイス上で OSPF ルーティングが更新されないようにします。
ステップ 23	[no] priority priority 例 : apic1(config-interface-policy) # priority 4	特定のネットワークの代表ルータ (DR) の決定に使用される OSPF インターフェイスプライオリティを設定します。指定できる範囲は 0 ～ 255 です。
ステップ 24	[no] retransmit-interval seconds 例 : apic1(config-interface-policy) # retransmit-interval 5	インターフェイスに属する隣接に対するリンクステートアドバタイズメント (LSA) の再送信間隔を指定します。指定できる範囲は 1 ～ 65535 秒です。

	コマンドまたはアクション	目的
ステップ 25	[no] transmit-delay seconds 例 : apic1(config-interface-policy) # transmit-delay 2	インターフェイス上でリンクステート更新パケットを送信するために必要とされる時間を設定します。範囲は 1 ～ 450 秒です。

例

次に、VRF テンプレートおよびインターフェイステンプレートを設定する例を示します。

```

apic1# configure
apic1(config)# leaf 101

      # CONFIGURING THE VRF TEMPLATE:
apic1(config-leaf)# template ospf vrf-policy vrfTemplate3 tenant exampleCorp
apic1(config-vrf-policy)# timers throttle lsa 200 10000 45000
apic1(config-vrf-policy)# timers lsa-group-pacing 240
apic1(config-vrf-policy)# timers lsa-arrival 1000
apic1(config-vrf-policy)# timers throttle spf 5 1000 90000
apic1(config-vrf-policy)# auto-cost reference-bandwidth 1000
apic1(config-vrf-policy)# distance 200
apic1(config-vrf-policy)# maximum-paths 8
apic1(config-vrf-policy)# graceful-restart helper-disable
apic1(config-vrf-policy)# prefix-suppression
apic1(config-vrf-policy)# name-lookup
apic1(config-vrf-policy)# exit

      # CONFIGURING THE INTERFACE TEMPLATE:
apic1(config-leaf)# template ospf interface-policy ifTemplate5 tenant exampleCorp
apic1(config-ospf-if-policy)# advertise-subnet
apic1(config-ospf-if-policy)# cost 300
apic1(config-ospf-if-policy)# dead-interval 60
apic1(config-ospf-if-policy)# hello-interval 10
apic1(config-ospf-if-policy)# mtu-ignore
apic1(config-ospf-if-policy)# network p2p
apic1(config-ospf-if-policy)# passive-interface
apic1(config-ospf-if-policy)# priority 4
apic1(config-ospf-if-policy)# retransmit-interval 5
apic1(config-ospf-if-policy)# transmit-delay 2

```

BGP の設定

『Configuring BGP』

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	bgp-fabric 例 : apic1(config)# bgp-fabric	ファブリックの BGP 設定モードを開始します。
ステップ 3	asn <i>asn-number</i> 例 : apic1(config-bgp-fabric)# asn 100	BGP 自律システム番号 (ASN) を指定します。
ステップ 4	route-reflector spine <i>spine-id</i> 例 : apic1(config-bgp-fabric)# route-reflector spine 105	指定したスパイン スイッチを BGP ルート リフレクタに設定します。

例

```
apic1# configure
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 100
apic1(config-bgp-fabric)# route-reflector spine 105
```

次のタスク

BGP アドレス ファミリとカウンタを設定します。

BGP アドレス ファミリとタイマー テンプレートの作成

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	template bgp timers timer-policy-name tenant tenant-name 例 : apic1(config-leaf)# template bgp timers bgpTimers tenant exampleCorp This template will be available on all leaves where tenant exampleCorp has a VRF deployment	指定したテナント下に BGP タイマー ポリシーテンプレートを作成します。
ステップ 4	graceful-restart-helper	BGP ポリシーのグレースフルリスタート ヘルパーを設定します。 apic1(config-bgp-timers)# graceful-restart-helper
ステップ 5	graceful-restart stalepath-time seconds	BGP が再起動中の BGP ピアからの古いルートを維持する最大時間を設定します。指定できる範囲は 1 ～ 3600 秒です。 apic1(config-bgp-timers)# graceful-restart stalepath-time 3600
ステップ 6	timers bgp keep-alive-seconds hold-seconds	キープアライブタイマーおよびホールドタイマーの値を設定します。両方の値の範囲は 1 ～ 3600 秒です。 apic1(config-bgp-timers)# timers bgp 10 20
ステップ 7	exit	apic1(config-bgp-timers)# exit
ステップ 8	template bgp address-family family-name tenant tenant-name 例 : apic1(config-leaf)# template bgp address-family bgpAfl tenant	指定したテナント下に BGP アドレス ファミリ テンプレートを作成します。

	コマンドまたはアクション	目的
	exampleCorp This template will be available on all leaves where tenant exampleCorp has a VRF deployment	
ステップ 9	distance <i>ebgp-distance ibgp-distance local-distance</i>	eBGP ルート、iBGP ルート、およびローカル ルートのアドミニストレーティブディスタンスを設定します。指定できる範囲は 1 ～ 255 です。 apic1(config-bgp-af) # distance 250 240 230
ステップ 10	exit	リーフ コンフィギュレーションモードに戻ります。 apic1(config-bgp-af) # exit

例

次に、BGP タイマー テンプレートとアドレス ファミリ テンプレートを作成する例を示します。

```
apic1# configure
apic1(config)# leaf 101

# CREATE A TIMER TEMPLATE
apic1(config-leaf)# template bgp timers bgpTimers tenant exampleCorp
This template will be available on all leaves where tenant exampleCorp has a VRF deployment
apic1(config-bgp-timers)# timers bgp 10 20
apic1(config-bgp-timers)# graceful-restart stalepath-time 3600
apic1(config-bgp-timers)# exit

# CREATE AN ADDRESS FAMILY TEMPLATE
apic1(config-leaf)# template bgp address-family bgpAf1 tenant bgp_t1
This template will be available on all leaves where tenant exampleCorp has a VRF deployment
apic1(config-bgp-af)# distance 250 240 230
apic1(config-bgp-af)# exit
apic1(config-leaf)# exit
```

BGP アドレス ファミリとタイマーの設定

始める前に

BGP アドレス ファミリのテンプレートとタイマーのテンプレートを作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーション モードに入ります。
ステップ 2	leaf node-id 例 : <code>apic1(config)# leaf 101</code>	設定するリーフを指定します。
ステップ 3	router bgp asn-number 例 : <code>apic1(config-leaf)# router bgp 100</code>	BGP ポリシー設定を入力します。
ステップ 4	vrf member tenant tenant-name vrf vrf-name 例 : <code>apic1(config-bgp)# vrf member tenant exampleCorp vrf v100</code>	後続のアドレス ファミリ コンフィギュレーション モード コマンドと関連付ける VRF インスタンスを指定します。
ステップ 5	inherit bgp timer timer-name 例 : <code>apic1(config-leaf-bgp-vrf)# inherit bgp timer bgpTimers</code> This template will be inherited on all leaves where VRF v100 has been deployed	既存のタイマー設定が適用されます。
ステップ 6	address-family {ipv4 ipv6} unicast 例 : <code>apic1(config-leaf-bgp-vrf)# address-family ipv4 unicast</code>	正常な IPv4 ユニキャスト ルートを交換するネイバーを宣言します。
ステップ 7	inherit bgp address-family family-name 例 : <code>apic1(config-leaf-bgp-vrf-af)# inherit bgp address-family ipv4-af-pol</code> This template will be inherited on all leaves where VRF v100 has been deployed	このアドレス ファミリに指定したアドレス ファミリを追加します。
ステップ 8	exit 例 : <code>apic1(config-leaf-bgp-vrf-af)# exit</code>	

例

この例では、BGP タイマー設定、IPv4 および IPv6 アドレス ファミリを継承する方法を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant exampleCorp vrf v100
apic1(config-leaf-bgp-vrf)# inherit bgp timer bgpTimers
This template will be inherited on all leaves where VRF v100 has been deployed
apic1(config-leaf-bgp-vrf)# address-family ipv4 unicast
apic1(config-leaf-bgp-vrf-af)# inherit bgp address-family ipv4-af-pol
This template will be inherited on all leaves where VRF v100 has been deployed
apic1(config-leaf-bgp-vrf-af)# exit
apic1(config-leaf-bgp-vrf)# address-family ipv6 unicast
apic1(config-leaf-bgp-vrf-af)# inherit bgp address-family ipv6-af-pol
This template will be inherited on all leaves where VRF v100 has been deployed
apic1(config-leaf-bgp-vrf-af)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf)# exit
```

BGP ネイバーの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーションモードに入ります。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	router bgp asn-number 例 : apic1(config-leaf)# router bgp 100	BGP ポリシー設定を入力します。
ステップ 4	vrf member tenant tenant-name vrf vrf-name 例 : apic1(config-bgp)# vrf member tenant exampleCorp vrf v100	後続のポリシー設定モードコマンドと関連付ける VRF インスタンスを指定します。

	コマンドまたはアクション	目的
ステップ 5	(任意) aggregate-address <i>ip-address/masklength</i> [as-set] 例 : <pre>apicl(config-leaf-bgp-vrf)# aggregate-address 192.0.10.0/24 as-set</pre>	アドレスの範囲のサマリーアドレスを設定し、BGP データベースに集約エントリを作成します。アドレスは IPv4 または IPv6 を使用できます。The as-set option generates autonomous system set path information.
ステップ 6	neighbor <i>ip-address</i>[/<i>masklength</i>] 例 : <pre>apicl(config-leaf-bgp-vrf)# neighbor 192.0.2.229/32</pre>	ネイバーの IP アドレスを指定します。
ステップ 7	address-family {<i>ipv4</i> <i>ipv6</i>} unicast 例 : <pre>apicl(config-leaf-bgp-vrf-neighbor)# address-family ipv4 unicast</pre>	正常な IPv4 ユニキャスト ルートを交換するネイバーを宣言します。
ステップ 8	[no] maximum-prefix <i>count</i> [action {log shut restart [restart-time <i>minutes</i>]}] [threshold <i>percent</i>] 例 : <pre>apicl(config-leaf-bgp-vrf-neighbor-af)# maximum-prefix 10 threshold 10 action restart restart-time 10</pre>	このネイバーからのプレフィックスの最大数を設定します。範囲は 1 ～ 300000 プレフィックスです。その他のオプションの設定は次のとおりです。 • action : 最大プレフィックスの制限に達したときに実行するアクションです。If the action is restart, you can optionally specify the restart-time, which is the period of time in minutes before restarting the peer when the maximum prefix limit is reached. 指定できる範囲は 1 ～ 65535 分です。 • threshold : 警告が発行されるまでのプレフィックスの最大数のしきい値の割合。指定できる範囲は 1 ～ 100% です。
ステップ 9	exit 例 : <pre>apicl(config-leaf-bgp-vrf-neighbor-af)# exit</pre>	
ステップ 10	update-source {loopback <i>ip-address</i> ethernet <i>ip-address</i> vlan <i>vlan-id</i>} 例 : <pre>apicl(config-leaf-bgp-vrf-neighbor)# update-source loopback 192.0.2.230</pre>	ネイバー アドレスが OSPF を通じて学習される場合、OSPF で使用されるのと同じループバックアドレスを指定します。

	コマンドまたはアクション	目的
ステップ 11	weight number 例 : <pre>apic1(config-leaf-bgp-vrf-neighbor) # weight 2000</pre>	最適なパスを選択するために、重み属性を指定します。重みは、0 ～ 65,535 で指定できます。同じ宛先に対して複数のルートがある場合、より高い重み値のルートが優先されます。
ステップ 12	private-as-control {remove-exclusive remove-exclusive-all remove-exclusive-all-replace-as} 例 : <pre>apic1(config-leaf-bgp-vrf-neighbor) # private-as-control</pre>	自律システムパスからプライベート自律システム番号を削除します。次の 3 つの可能なバリエーションに従って、プライベート AS 番号をピアごとの AS パスから削除し、eBGP ピアのみで 사용할ことができます。 • remove-exclusive : AS パスがプライベート AS 番号のみ場合、削除します。 • remove-exclusive-all : AS パスがプライベートおよびパブリック AS 番号の両方の場合、削除します。 • remove-exclusive-all-replace-as : ルータのローカル AS 番号とプライベート AS 番号を置き換えます。 このコマンドは例として表示されています。この時点で、次の表に示すネイバー設定のいずれかを設定できます。

次の表に、この時点で設定できるインターフェイス設定を示します。

コマンド	目的
allow-self-as	as-path を、その中に自身の AS を含めて受け入れます
allowed-self-as-count <i>count</i>	ローカルアクセスサービスネットワークの発生件数を指定します
disable-connected-check	直接接続されたピアのチェックをディセーブルにします
disable-peer-as-check	アドバタイジング中のピア AS 番号のチェックをディセーブルにします
ebgp-multihop <i>count</i>	リモートピアにマルチホップ TTL を指定します

コマンド	目的
<code>local-as <i>asn</i></code>	BGP ピアのローカルの自律システム設定をします
<code>next-hop-self</code>	ピアリング アドレスをネクストホップとして設定します
<code>password <i>password</i></code>	ネイバーのパスワードを設定します
<code>private-as-control</code>	AS パスからプライベート ASN を削除します
<code>remote-as <i>asn</i></code>	ネイバーの自律システム番号を指定します
<code>route-map <i>name</i> {in out}</code>	ネイバーにルートマップ適用します
<code>send-community [extended]</code>	コミュニティ属性をこのネイバーに送信します
<code>update-source vlan <i>vlan-id</i></code>	送信元 VLAN インターフェイスを指定します
<code>update-source ethernet <i>slot/port</i></code>	送信元イーサネット インターフェイスを指定します
<code>update-source loopback <i>ip-address</i></code>	送信元ルックバック インターフェイスを指定します
<code>weight <i>number</i></code>	ルーティングテーブルの BGP 重みを指定します

例

この例では、IPv4 BGP ネイバーを設定する方法を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant exampleCorp vrf v100
apic1(config-leaf-bgp-vrf)# aggregate-address 192.0.10.0/24 as-set
apic1(config-leaf-bgp-vrf)# neighbor 192.0.2.229/32
apic1(config-leaf-bgp-vrf-neighbor)# address-family ipv4 unicast
apic1(config-leaf-bgp-vrf-neighbor-af)# maximum-prefix 10 threshold 10 action restart
restart-time 10
apic1(config-leaf-bgp-vrf-neighbor-af)# exit
apic1(config-leaf-bgp-vrf-neighbor)# allow-self-as
apic1(config-leaf-bgp-vrf-neighbor)# allowed-self-as-count 2
apic1(config-leaf-bgp-vrf-neighbor)# disable-connected-check
apic1(config-leaf-bgp-vrf-neighbor)# disable-peer-as-check
apic1(config-leaf-bgp-vrf-neighbor)# ebgp-multihop 4
apic1(config-leaf-bgp-vrf-neighbor)# local-as 100
apic1(config-leaf-bgp-vrf-neighbor)# next-hop-self
apic1(config-leaf-bgp-vrf-neighbor)# password abcdef
apic1(config-leaf-bgp-vrf-neighbor)# remote-as 200

```

```

apic1(config-leaf-bgp-vrf-neighbor)# send-community extended
apic1(config-leaf-bgp-vrf-neighbor)# update-source vlan 601
apic1(config-leaf-bgp-vrf-neighbor)# update-source ethernet 1/15
apic1(config-leaf-bgp-vrf-neighbor)# update-source loopback 192.0.2.230
Warning: BGP Configuration changed. Please re-configure BGP Password if it was enabled
apic1(config-leaf-bgp-vrf-neighbor)# local-as 100 no-prepend replace-as dual-as
apic1(config-leaf-bgp-vrf-neighbor)# route-map rMapT3 out
apic1(config-leaf-bgp-vrf-neighbor)# weight 2000
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf)# exit

```

この例では、IPv6 BGP ネイバーを設定する方法を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant exampleCorp vrf v100
apic1(config-leaf-bgp-vrf)# neighbor 2001:80:1:2::229
apic1(config-leaf-bgp-vrf-neighbor)# address-family ipv6 unicast
apic1(config-leaf-bgp-vrf-neighbor-af)# maximum-prefix 100
apic1(config-leaf-bgp-vrf-neighbor-af)# exit
apic1(config-leaf-bgp-vrf-neighbor)# allow-self-as
apic1(config-leaf-bgp-vrf-neighbor)# allowed-self-as-count 2
apic1(config-leaf-bgp-vrf-neighbor)# disable-connected-check
apic1(config-leaf-bgp-vrf-neighbor)# disable-peer-as-check
apic1(config-leaf-bgp-vrf-neighbor)# ebgp-multihop 4
apic1(config-leaf-bgp-vrf-neighbor)# local-as 100
apic1(config-leaf-bgp-vrf-neighbor)# next-hop-self
apic1(config-leaf-bgp-vrf-neighbor)# password abcdef
apic1(config-leaf-bgp-vrf-neighbor)# remote-as 200
apic1(config-leaf-bgp-vrf-neighbor)# send-community extended
apic1(config-leaf-bgp-vrf-neighbor)# update-source vlan 601
apic1(config-leaf-bgp-vrf-neighbor)# update-source ethernet 1/15
apic1(config-leaf-bgp-vrf-neighbor)# update-source loopback 2001:80:1:2::230/128
Warning: BGP Configuration changed. Please re-configure BGP Password if it was enabled
apic1(config-leaf-bgp-vrf-neighbor)# local-as 100 no-prepend replace-as dual-as
apic1(config-leaf-bgp-vrf-neighbor)# route-map rMapT3 out
apic1(config-leaf-bgp-vrf-neighbor)# weight 2000
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf-af)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf)# exit

```

NX-OS スタイル CLI を使用してノード BGP タイマー ポリシーあたりの VRF あたりを設定する

手順

	コマンドまたはアクション	目的
ステップ 1	タイマー ポリシーを作成する前に、BGP ASN およびルート リフレクタを設定します。 例 : <pre> apic1(config)# apic1(config)# bgp-fabric apic1(config-bgp-fabric)# route-reflector spine 102 apic1(config-bgp-fabric)# asn 42 apic1(config-bgp-fabric)# exit apic1(config)# exit apic1# </pre>	
ステップ 2	タイマー ポリシーを作成します。 例 : <pre> apic1# config apic1(config)# leaf 101 apic1(config-leaf)# template bgp timers pol7 tenant tn1 This template will be available on all nodes where tenant tn1 has a VRF deployment apic1(config-bgp-timers)# timers bgp 120 240 apic1(config-bgp-timers)# graceful-restart stalepath-time 500 apic1(config-bgp-timers)# maxas-limit 300 apic1(config-bgp-timers)# exit apic1(config-leaf)# exit apic1(config)# exit apic1# </pre>	特定の値は、例としてのみ提供されます。
ステップ 3	設定された BGP ポリシーを表示します。 例 : <pre> apic1# show run leaf 101 template bgp timers pol7 # Command: show running-config leaf 101 template bgp timers pol7 leaf 101 template bgp timers pol7 tenant tn1 timers bgp 120 240 graceful-restart stalepath-time </pre>	

	コマンドまたはアクション	目的
	<pre> 500 maxas-limit 300 exit exit </pre>	
ステップ 4	ノードで特定のポリシーを参照します。 例 : <pre> apicl# config apicl(config)# leaf 101 apicl(config-leaf)# router bgp 42 apicl(config-leaf-bgp)# vrf member tenant tn1 vrf ctx1 apicl(config-leaf-bgp-vrf)# inherit node-only bgp timer pol7 apicl(config-leaf-bgp-vrf)# exit apicl(config-leaf-bgp)# exit apicl(config-leaf)# exit apicl(config)# exit apicl# </pre>	
ステップ 5	特定の BGP のタイマー ポリシーのノードが表示されます。 例 : <pre> apicl# show run leaf 101 router bgp 42 vrf member tenant tn1 vrf ctx1 # Command: show running-config leaf 101 router bgp 42 vrf member tenant tn1 vrf ctx1 leaf 101 router bgp 42 vrf member tenant tn1 vrf ctx1 inherit node-only bgp timer pol7 exit exit exit apicl# </pre>	

BGP 最長パスの設定

始める前に

適切なテナントおよび BGP 外部ルーテッド ネットワークが作成され使用可能です。

次の機能を使用すると、ルート表にパスの最大数を追加し、コスト マルチパスのロード バランシングを有効にできます。

さらに多くのパスを設定できるようにする 2 つのプロパティは、bgpCtxAfPol オブジェクトの maxEcmp と maxEcmpIbgp です。これら 2 つのプロパティを設定した後、実装の残り部分に反映されます。

BGP にログインして、次のコマンドを使用します:

```
maximum-paths [ibgp]
no maximum-paths [ibgp]
```

設定例:

手順

例:

```
apicl(config)# leaf 101
apicl(config-leaf)# template bgp address-family newAf tenant t1
This template will be available on all nodes where tenant t1 has a VRF deployment
apicl(config-bgp-af)# maximum-paths ?
<1-16> Maximum number of equal-cost paths for load sharing. The default is 16.
ibgp Configure multipath for IBGP paths
apicl(config-bgp-af)# maximum-paths 10
apicl(config-bgp-af)# maximum-paths ibgp 8
apicl(config-bgp-af)# end
apicl#
no maximum-paths [ibgp]
```

AS パス プリペンドの設定

BGP ピアは、AS パス アトリビュートの長さを増やすことで、リモート ピアでベストパス選択の影響を与えることができます。番号として指定桁の前に付加して AS パス アトリビュートの長さを向上するために使用するメカニズムを提供する AS パス Prepend。

AS パス前に付加は、ルートマップを使用してアウトバウンド方向にのみ適用できます。パスとして前に付加が機能しない iBGP セッションで。

AS パス Prepend 機能は、次のように変更を有効に。

プリペンド	ルートマップと一致するルートの AS パスに、指定した AS 番号を付加します。 (注) • 1 個以上の AS 番号を設定できます。 • 4 バイト番号がサポートされています。 • 合計を prepend は 32 の AS 番号。AS 番号は、AS パスアトリビュートに挿入されます順序を指定する必要があります。
Prepend-最後-として	最後の前に付加 AS パス 1 から 10 までの範囲に番号として。

次の表では、AS パス Prepend の実装の選択基準について説明します。

プリペンド	1	指定された AS 番号を追加します。

Prepend-最後-として	2	最後の AS 番号を AS パスに付加します。
デフォルト	Prepend(1)	指定された AS 番号を追加します。

NX-OS スタイルの CLI を使用した AS パスのプリペンド

このセクションでは、NX-OS スタイル コマンドライン インターフェイス (CLI) を使用して、AS パスのプリペンド機能を実現する方法について説明します。

始める前に

構成済みのテナント

手順

境界ゲートウェイ プロトコル (BGP) ルートの自動システムパス (AS パス) を変更するには、`set as-path` コマンドを使用します。`set as-path` コマンドは、`apic1(config-leaf-vrf-template-route-profile)# set as-path {'prepend as-num [ ,... as-num ] | prepend-last-as num}` の形式で実行します。

例 :

```
apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# template route-profile rp1
apic1(config-leaf-vrf-template-route-profile)# set as-path ?
prepend Prepend to the AS-Path
prepend-last-as Prepend last AS to the as-path
apic1(config-leaf-vrf-template-route-profile)# set as-path prepend 100, 101, 102, 103
apic1(config-leaf-vrf-template-route-profile)# set as-path prepend-last-as 8
apic1(config-leaf-vrf-template-route-profile)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
```

次のタスク

AS パスのプリペンドを無効にするには、示されているコマンドの `no` 形式を使用します:

```
apic1(config-leaf-vrf-template-route-profile)# [no] set
as-path { prepend as-num [ ,... as-num ] | prepend-last-as num}
```

BGP にディストリビューションをルーティング

テナントの範囲によるルート プロファイルの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id 例 : <code>apic1(config)# leaf 101</code>	設定するリーフを指定します。
ステップ 3	template route-profile profile-name tenant tenant-name 例 : <code>apic1(config-leaf)# template route-profile map_eigrp tenant exampleCorp</code>	BGP ダンプニングとルート再配布のテナントの下にルート プロファイル テンプレートを作成します。
ステップ 4	必須: [no] set tag name 例 : <code>apic1(config-leaf-template-route-profile)# set tag 200</code>	タグ値を設定します。 <i>name</i> パラメータは符号なし整数です。
ステップ 5	必須: exit 例 : <code>apic1(config-leaf-template-route-profile)# exit</code>	リーフ コンフィギュレーション モードに戻ります。
ステップ 6	template route-profile profile-name tenant tenant-name 例 : <code>apic1(config-leaf)# template route-profile map_ospf tenant exampleCorp</code>	BGP ダンプニングとルート再配布のテナントの下にルート プロファイル テンプレートを作成します。
ステップ 7	必須: [no] set tag name 例 : <code>apic1(config-leaf-template-route-profile)# set tag 100</code>	タグ値を設定します。 <i>name</i> パラメータは符号なし整数です。

例

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# template route-profile map_eigrp tenant exampleCorp
apic1(config-leaf-template-route-profile)# set tag 200
apic1(config-leaf-template-route-profile)# exit
apic1(config-leaf)# template route-profile map_ospf tenant exampleCorp
apic1(config-leaf-template-route-profile)# set tag 100
apic1(config-leaf-template-route-profile)# exit

```

次のタスク

この手順で作成されたルートプロファイルのいずれかを使用して、OSPF および EIGRP の BGP の下に再配布ルート プロファイルを設定します。

再配布ルート プロファイルの設定

始める前に

ルート再配布のテナントの下にルート プロファイル テンプレートを作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	router bgp asn-number 例 : apic1(config-leaf)# router bgp 100	BGP ポリシー設定を入力します。
ステップ 4	vrf member tenant tenant-name vrf vrf-name 例 : apic1(config-bgp)# vrf member tenant exampleCorp vrf v100	後続のポリシー設定モード コマンドと関連付ける VRF インスタンスを指定します。
ステップ 5	必須: redistribute {ospf eigrp} route-map map-name	

	コマンドまたはアクション	目的
	例 : <pre>apic1(config-leaf-bgp-vrf)# redistribute ospf route-map map_ospf</pre>	

例

この例では、テナントの範囲によるルートプロファイルの作成の例で作成したルートプロファイルを使用して、OSPF および EIGRP の BGP の下に再配布ルートプロファイルを設定します。再配布ルートマップは、すべてのルートを可能（許可）し、ルート制御アクションに関するルートプロファイルを適用します。この例では、すべての EIGRP 学習ルートは、タグ 200 を使用して BGP に再配布され、OSPF ルートは、タグ 100 を使用して BGP に再配布されます。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant exampleCorp vrf v1
apic1(config-leaf-bgp-vrf)# redistribute eigrp route-map map_eigrp
apic1(config-leaf-bgp-vrf)# redistribute ospf route-map map_ospf
```

BGP ルート ダンプニングの設定

BGP ルート ダンプニングでは、境界リーフ スイッチ (BLs) に接続されている外部ルータから受信したフラッピング eBGP ルートのファブリックへの伝搬を最小限に抑えます。外部ルータから頻繁にフラッピングするルートは設定された条件に基づいて BLs で抑制され、iBGP ピア (ACI スパインスイッチ) への再配布が禁止されます。抑制されたルートは設定された時間条件の後で再使用されます。各フラップは eBGP ルートに 1000 のペナルティを科します。フラップのペナルティが定義された抑制限度のしきい値 (デフォルトは 2000) に達すると、eBGP ルートは抑制としてマーキングされます。抑制されたルートは他の BGP ピアにはアドバタイズされません。ペナルティは、半減期 (デフォルトは 15 分) ごとに半分に減少します。抑制されたルートは、ペナルティが指定された再利用の限度 (デフォルトは 750) を下回ると再利用されます。抑制されたルートは、指定された最大抑制時間 (最大 45 分間) の間最大限に抑制されます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>apic1# configure</pre>	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	leaf node-id 例 : <pre>apic1(config)# leaf 101</pre>	設定するリーフを指定します。
ステップ 3	template route-profile profile-name tenant tenant-name 例 : <pre>apic1(config-leaf)# template route-profile damp_rp tenant exampleCorp</pre>	BGP ダンプニングとルート再配布のテナントの下にルートプロファイルテンプレートを作成します。
ステップ 4	必須: [no] set dampening half-life reuse suppress max-suppress-time 例 : <pre>apic1(config-leaf-template-route-profile)# set dampening 15 750 2000 60</pre>	ルートフラップダンプニングの動作を設定します。パラメータは次のとおりです。 • half-life : 時間経過後にペナルティが減少する分単位の半減期。ルートにペナルティを割り当てると、ペナルティは半減期で半分まで小さくなります。指定できる範囲は 1 ～ 60 分で、デフォルト値は 15 分です。 • reuse : フラッピングルートのペナルティがこの値を下回るまで減少すると、ルートの抑制が中止（再利用）されます。指定できる値の範囲は 1 ～ 20000 です。デフォルトは 750 です。 • suppress : ペナルティがこの制限を超えるとルートは抑制されます。指定できる値の範囲は 1 ～ 20000 です。デフォルトは 2000 です。 • max-suppress-time : 安定したルートを抑制できる分単位の最大時間。指定できる範囲は 1 ～ 255 です。
ステップ 5	必須: exit 例 : <pre>apic1(config-leaf-template-route-profile)# exit</pre>	リーフ コンフィギュレーションモードに戻ります。
ステップ 6	router bgp asn-number 例 :	BGP ポリシー設定を入力します。

	コマンドまたはアクション	目的
	<code>apic1(config-leaf) # router bgp 100</code>	
ステップ 7	vrf member tenant <i>tenant-name</i> vrf <i>vrf-name</i> 例 : <code>apic1(config-bgp) # vrf member tenant exampleCorp vrf v100</code>	後続のポリシー設定モードコマンドと関連付ける VRF インスタンスを指定します。
ステップ 8	neighbor <i>ip-address</i>[/<i>masklength</i>] 例 : <code>apic1(config-leaf-bgp-vrf) # neighbor 192.0.2.229/32</code>	ネイバーの IP アドレスを指定します。マスク長は 32 です。
ステップ 9	address-family {<i>ipv4</i> <i>ipv6</i>} unicast 例 : <code>apic1(config-leaf-bgp-vrf-neighbor) # address-family ipv4 unicast</code>	正常な IPv4 ユニキャスト ルートを交換するネイバーを宣言します。
ステップ 10	inherit bgp dampening <i>profile-name</i> 例 : <code>apic1(config-leaf-bgp-vrf-neighbor-af) # inherit bgp dampening damp_rp</code>	
ステップ 11	exit 例 : <code>apic1(config-leaf-bgp-vrf-neighbor-af) # exit</code>	
ステップ 12	exit 例 : <code>apic1(config-leaf-bgp-vrf-neighbor) # exit</code>	
ステップ 13	address-family {<i>ipv4</i> <i>ipv6</i>} unicast 例 : <code>apic1(config-leaf-bgp-vrf) # address-family ipv4 unicast</code>	正常な IPv4 ユニキャスト ルートを交換するネイバーを宣言します。
ステップ 14	inherit bgp dampening <i>profile-name</i> 例 : <code>apic1(config-leaf-bgp-vrf-af) # inherit bgp dampening damp_rp</code>	
ステップ 15	exit 例 : <code>apic1(config-leaf-bgp-vrf-af) # exit</code>	

例

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# template route-profile damp_rp tenant exampleCorp
apic1(config-leaf-template-route-profile)# set dampening 15 750 2000 60
apic1(config-leaf-template-route-profile)# exit
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant exampleCorp vrf v100
apic1(config-leaf-bgp-vrf)# neighbor 192.0.2.229/32
apic1(config-leaf-bgp-vrf-neighbor)# address-family ipv4 unicast
apic1(config-leaf-bgp-vrf-neighbor-af)# inherit bgp dampening damp_rp
apic1(config-leaf-bgp-vrf-neighbor-af)# exit
apic1(config-leaf-bgp-vrf)# address-family ipv6 unicast
apic1(config-leaf-bgp-vrf-af)# inherit bgp dampening damp_rp
apic1(config-leaf-bgp-vrf-af)# exit

```

EIGRP の設定

EIGRP VRF とインターフェイス テンプレートの作成

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	template eigrp vrf-policy vrf-policy-name tenant tenant-name 例 : apic1(config-leaf)# template eigrp vrf-policy vrfTemplate3 tenant exampleCorp This template will be available on all leaves where tenant exampleCorp has a VRF deployment	指定したテナント下に EIGRP VRF ポリシー テンプレートを作成します。
ステップ 4	distance internal external 例 :	EIGRP 内部ルートと外部ルートの管理 ディスタンスプリファレンスを設定し

	コマンドまたはアクション	目的
	<code>apic1(config-template-eigrp-vrf-pol)# distance 2 5</code>	ます。ディスタンスは 1 ～ 255 で指定できます。
ステップ 5	maximum-paths limit 例 : <code>apic1(config-template-eigrp-vrf-pol)# maximum-paths 8</code>	VRF ポリシー テンプレートの EIGRP の最大パスの制限を設定します。制限は 1 ～ 32 で指定できます。
ステップ 6	metric version 64bit 例 : <code>apic1(config-template-eigrp-vrf-pol)# metric version 64bit</code>	広範なメトリック (64 ビット) に EIGRP メトリック形式を設定します。
ステップ 7	timers active-time minutes 例 : <code>apic1(config-template-eigrp-vrf-pol)# timers active-time 1</code>	EIGRP のアクティブ タイマーの間隔を設定します。指定できる範囲は 1 ～ 65535 分です。
ステップ 8	template eigrp interface-policy if-policy-name tenant tenant-name 例 : <code>apic1(config-leaf)# template eigrp interface-policy ifTemplate5 tenant exampleCorp</code> This template will be available on all leaves where tenant exampleCorp has a VRF deployment	指定したテナント下に EIGRP インターフェイス ポリシー テンプレートを作成します。
ステップ 9	ip hello-interval eigrp default seconds 例 : <code>apic1(config-template-eigrp-if-pol)# ip hello-interval eigrp default 10</code>	EIGRP Hello の間隔時間を設定します。指定できる範囲は 1 ～ 65535 秒です。
ステップ 10	ip hold-interval eigrp default seconds 例 : <code>apic1(config-template-eigrp-if-pol)# ip hold-interval eigrp default 10</code>	EIGRP ホールドの間隔時間を設定します。指定できる範囲は 1 ～ 65535 秒です。
ステップ 11	ip next-hop-self eigrp default 例 : <code>apic1(config-template-eigrp-if-pol)# ip next-hop-self eigrp default</code>	EIGRP の next-hop-self フラグを設定します。
ステップ 12	ip passive-interface eigrp default 例 : <code>apic1(config-template-eigrp-if-pol)# ip passive-interface eigrp default</code>	EIGRP の passive-interface フラグを設定します。

	コマンドまたはアクション	目的
ステップ 13	ip split-horizon eigrp default 例 : <pre>apic1(config-template-eigrp-if-pol)# ip split-horizon eigrp default</pre>	EIGRP の split-horizon フラグを設定します。
ステップ 14	exit 例 : <pre>apic1(config-template-eigrp-if-pol)# exit</pre>	リーフ コンフィギュレーションモードに戻ります。

例

```
apic1# configure
apic1(config)# leaf 101

# CONFIGURING THE VRF TEMPLATE:
apic1(config-leaf)# template eigrp vrf-policy vrfTemplate3 tenant exampleCorp
This template will be available on all leaves where tenant exampleCorp has a VRF deployment
apic1(config-template-eigrp-vrf-pol)# distance 2 5
apic1(config-template-eigrp-vrf-pol)# maximum-paths 8
apic1(config-template-eigrp-vrf-pol)# metric version 64bit
apic1(config-template-eigrp-vrf-pol)# timers active-time 1
apic1(config-template-eigrp-vrf-pol)# exit

# CONFIGURING THE INTERFACE TEMPLATE:
apic1(config-leaf)# template eigrp interface-policy ifTemplate5 tenant exampleCorp
This template will be available on all leaves where tenant exampleCorp has a VRF deployment
apic1(config-template-eigrp-if-pol)# ip hello-interval eigrp default 5
apic1(config-template-eigrp-if-pol)# ip hold-interval eigrp default 10
apic1(config-template-eigrp-if-pol)# ip next-hop-self eigrp default
apic1(config-template-eigrp-if-pol)# ip passive-interface eigrp default
apic1(config-template-eigrp-if-pol)# ip split-horizon eigrp default
apic1(config-template-eigrp-if-pol)# exit
apic1(config-leaf)# exit
apic1(config)# exit
```

次のタスク

EIGRP アドレス ファミリとカウンタを設定します。

EIGRP アドレス ファミリとカウンタの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id 例 : apicl(config)# leaf 101	設定するリーフを指定します。
ステップ 3	router eigrp default 例 : apicl(config-leaf)# router eigrp default	EIGRP ポリシー設定を入力します。
ステップ 4	vrf member tenant tenant-name vrf vrf-name 例 : apicl(config-eigrp)# vrf member tenant exampleCorp vrf v100	後続のアドレスファミリ コンフィギュレーションモードコマンドと関連付ける VRF インスタンスを指定します。
ステップ 5	autonomous-system asn 例 : apicl(config-eigrp-vrf)# autonomous-system 300	EIGRP の自律システム設定を入力します。
ステップ 6	address-family {ipv4 ipv6} unicast 例 : apicl(config-eigrp-vrf)# address-family ipv4 unicast	EIGRP ポリシー アドレス ファミリを設定します。
ステップ 7	distance internal external 例 : apicl(config-address-family)# distance 2 5	EIGRP 内部ルートと外部ルートの管理ディスタンスプリファレンスを設定します。ディスタンスは 1 ～ 255 で指定できます。
ステップ 8	maximum-paths limit 例 : apicl(config-address-family)# maximum-paths 8	VRF ポリシー テンプレートの EIGRP の最大パスの制限を設定します。制限は 1 ～ 32 で指定できます。

	コマンドまたはアクション	目的
ステップ 9	metric version 64bit 例 : <pre>apic1(config-address-family)# metric version 64bit</pre>	広範なメトリック (64 ビット) に EIGRP メトリック形式を設定します。
ステップ 10	timers active-time minutes 例 : <pre>apic1(config-address-family)# timers active-time 1</pre>	EIGRP のアクティブ タイマーの間隔を設定します。指定できる範囲は 1 ～ 65535 分です。
ステップ 11	inherit eigrp vrf-policy vrf-policy-name 例 : <pre>apic1(config-address-family)# inherit eigrp vrf-policy vrfTemplate3</pre>	このアドレス ファミリに EIGRP VRF ポリシーを適用します。

例

この例では、EIGRP アドレス ファミリを設定する方法および EIGRP VRF ポリシーを継承する方法を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router eigrp default
apic1(config-eigrp)# vrf member tenant exampleCorp vrf v100
apic1(config-eigrp-vrf)# autonomous-system 300
apic1(config-eigrp-vrf)# address-family ipv4 unicast
This configuration will affect all leaves where VRF v100 has been deployed
apic1(config-address-family)# distance 2 5
This configuration will affect all leaves where VRF v100 has been deployed
apic1(config-address-family)# maximum-paths 8
This configuration will affect all leaves where VRF v100 has been deployed
apic1(config-address-family)# metric version 64bit
This configuration will affect all leaves where VRF v100 has been deployed
apic1(config-address-family)# timers active-time 1
This configuration will affect all leaves where VRF v100 has been deployed
apic1(config-address-family)# inherit eigrp vrf-policy vrfTemplate3
This template will be inherited on all leaves where VRF v100 has been deployed
apic1(config-address-family)# exit
apic1(config-eigrp-vrf)# exit
apic1(config-eigrp)# exit
```

EIGRP インターフェイスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	コンフィギュレーションモードに入ります。
ステップ 2	leaf node-id 例 : apicl(config)# leaf 101	設定するリーフを指定します。
ステップ 3	interface ethernet slot/port 例 : apicl(config-leaf)# interface ethernet 1/21	設定するインターフェイスを指定します。
ステップ 4	[no] switchport slot/port 例 : apicl(config-leaf-if)# no switchport	デフォルトでは、ポートはレイヤ 2 トランク モードです。ポートがレイヤ 3 モードである場合、このコマンドを使用して、レイヤ 2 トランク モードに変換する必要があります。
ステップ 5	[no] vlan-domain member vlan-id 例 : apicl(config-leaf-if)# vlan-domain member dom1	VLAN ドメインのコンフィギュレーション モードを作成および開始します。
ステップ 6	[no] vrf member tenant exampleCorp vrf vrf-name 例 : apicl(config-leaf-if)# vrf member tenant exampleCorp vrf v100	インターフェイスを VRF に関連付けます。
ステップ 7	[no] {ip ipv6} address ip-address/mask-length 例 : apicl(config-leaf-if)# ip address 181.12.12.1/24	インターフェイスの IP アドレスを設定します。
ステップ 8	[no] {ip ipv6} router eigrp default 例 : apicl(config-leaf-if)# ip router eigrp default	EIGRP ルータのポリシーをデフォルトに設定します。

	コマンドまたはアクション	目的
ステップ 9	[no] {ip ipv6} distribute-list eigrp default route-map <i>map-name</i> out 例 : <pre>apic1(config-leaf-if)# ip distribute-list eigrp default route-map rMapT5 out</pre>	EIGRP は、distribute-list コマンドで指定されたルートマップで一致するルートをアドバタイズします。ルートマップのプレフィックスリストに記載されているルーティングプレフィックスは、BGP、OSPF、Static、Connected などの他のプロトコルソースから学習できます。再配布ルートマップは、distribute-list コマンドに基づいて自動的に作成されます。同じスイッチ上の別のインターフェイス上で実行されている EIGRP セッションから学んだプレフィックスが、配布リストによってフィルタリングされず、常に外にアドバタイズされることに注意してください。
ステップ 10	[no] {ip ipv6} hello-interval eigrp default <i>seconds</i> 例 : <pre>apic1(config-leaf-if)# ip hello-interval eigrp default 10</pre>	EIGRP Hello の間隔時間を設定します。指定できる範囲は 1 ～ 65535 秒です。
ステップ 11	[no] {ip ipv6} hold-interval eigrp default <i>seconds</i> 例 : <pre>apic1(config-leaf-if)# ip hold-interval eigrp default 10</pre>	EIGRP ホールドの間隔時間を設定します。指定できる範囲は 1 ～ 65535 秒です。
ステップ 12	[no] {ip ipv6} next-hop-self eigrp default 例 : <pre>apic1(config-leaf-if)# ip next-hop-self eigrp default</pre>	EIGRP の next-hop-self フラグを設定します。
ステップ 13	[no] {ip ipv6} passive-interface eigrp default 例 : <pre>apic1(config-leaf-if)# ip passive-interface eigrp default</pre>	EIGRP の passive-interface フラグを設定します。
ステップ 14	[no] {ip ipv6} split-horizon eigrp default 例 : <pre>apic1(config-leaf-if)# ip split-horizon eigrp default</pre>	EIGRP の split-horizon フラグを設定します。

	コマンドまたはアクション	目的
ステップ 15	[no] inherit eigrp ip interface-policy if-policy-name 例 : <pre>apicl(config-leaf-if)# inherit eigrp ip interface-policy ifTemplate5</pre>	このインターフェイスに EIGRP インターフェイスポリシーを適用します。
ステップ 16	[no] ip summary-address eigrp default ip-prefix 例 : <pre>apicl(config-leaf-if)# ip summary-address eigrp default 172.10.1.0/24</pre> <pre>apicl(config-leaf-if)# ip summary-address eigrp default 2001::/64</pre>	EIGRP のルート集約を設定します。サマリーアドレスは、EIGRP セッション上に集約されたプレフィックスをアドバタイズするように設定できます。 (注) 1 つのインターフェイスでネーブルになっているサマリーアドレスは、スイッチ上の同じ VRF 上の他の EIGRP 対応インターフェイスに適用されます。

例

この例では、EIGRP インターフェイスを設定する方法を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/21
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# vrf member tenant exampleCorp vrf v100
apicl(config-leaf-if)# ip address 181.12.12.1/24
apicl(config-leaf-if)# ip router eigrp default
apicl(config-leaf-if)# ip distribute-list eigrp default route-map rMapT5 out
distribute list will be updated on all EIGRP interfaces on node 1021 VRF exampleCorp/v100
apicl(config-leaf-if)# ip hello-interval eigrp default 5
apicl(config-leaf-if)# ip hold-interval eigrp default 10
apicl(config-leaf-if)# ip next-hop-self eigrp default
apicl(config-leaf-if)# ip passive-interface eigrp default
apicl(config-leaf-if)# ip split-horizon eigrp default
apicl(config-leaf-if)# inherit eigrp ip interface-policy ifTemplate5
apicl(config-leaf-if)# ip summary-address eigrp default 172.10.1.0/24
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
apicl(config)# exit
```

ルート マップの設定

テンプレートの設定

ルート プロファイルについて

ルートプロファイルは、インポート、エクスポートに使用されるルーティング制御設定のアクションを指定し、ルート マップを再配布します。ルート プロファイル テンプレートは、テナントまたはテナント VRF で定義できます。

テナント スコープ ルート プロファイルの設定

この手順では、BGP ダンプニングとルート再配布の設定に使用されるテナントスコープのルート プロファイルを作成します。

始める前に

- テナントおよび VRF を設定します。
- リーフで VRF を有効にします。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーションモードに入ります。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	[no] template route-profile profile-name tenant tenant-name 例 : apic1(config-leaf)# template route-profile rp1 tenant exampleCorp	テナントスコープのルートプロファイルを作成します。
ステップ 4	必須: [no] set community {regular extended} 値 {none replace additive} 例 : apic1(config-leaf-template-route-profile)# set community extended 20:22 additive	BGP コミュニティ属性を設定します。

	コマンドまたはアクション	目的
ステップ 5	必須: [no] set dampening half-life reuse suppress max-suppress-time 例 : <pre>apic1(config-leaf-template-route-profile)# set dampening 15 750 2000 60</pre>	ルートフラップ ダンプニングの動作を設定します。パラメータは次のとおりです。 • half-life : 時間経過後にペナルティが減少する分単位の半減期。ルートにペナルティを割り当てると、ペナルティは半減期で半分まで小さくなります。指定できる範囲は 1 ～ 60 分です。 • reuse : フラッピングルートのペナルティがこの値を下回るまで減少すると、ルートの抑制が中止（再利用）されます。指定できる範囲は 1 ～ 20000 です。 • suppress : ペナルティがこの制限を超えるとルートは抑制されます。指定できる範囲は 1 ～ 20000 です。 • max-suppress-time : 安定したルートを抑制できる分単位の最大時間。指定できる範囲は 1 ～ 255 です。
ステップ 6	必須: [no] set local-preference value 例 : <pre>apic1(config-leaf-template-route-profile)# set local-preference 64</pre>	BGP ローカルプリファレンス値を設定します。範囲は 0 ～ 4294967295 です。
ステップ 7	必須: [no] set metric value 例 : <pre>apic1(config-leaf-template-route-profile)# set metric 128</pre>	宛先ルーティング プロトコルのメトリックを設定します。
ステップ 8	必須: [no] set metric-type {type-1 type-2} 例 : <pre>apic1(config-leaf-template-route-profile)# set metric-type type-2</pre>	オプションは次のとおりです。 • type-1 : OSPF 外部タイプ 1 メトリック • type-2 : OSPF 外部タイプ 2 メトリック

	コマンドまたはアクション	目的
ステップ 9	必須: [no] set tag name 例 : <pre>apic1(config-leaf-template-route-profile)# set tag 1111</pre>	宛先ルーティングプロトコルのタグ値を設定します。 <i>name</i> パラメータは符号なし整数です。
ステップ 10	必須: [no] set weight weight 例 : <pre>apic1(config-leaf-template-route-profile)# set weight 20</pre>	宛先ルーティングプロトコルのタグ値を設定します。 <i>weight</i> パラメータは符号なし整数です。

例

次に、テナントスコープのルート プロファイルを設定する例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# template route-profile rp1 tenant exampleCorp
This template will be available on all leaves where tenant exampleCorp has a VRF deployment
apic1(config-leaf-template-route-profile)# set community extended 20:22 additive
apic1(config-leaf-template-route-profile)# set dampening 15 750 2000 60
apic1(config-leaf-template-route-profile)# set local-preference 64
apic1(config-leaf-template-route-profile)# set metric 128
apic1(config-leaf-template-route-profile)# set metric-type type-2
apic1(config-leaf-template-route-profile)# set tag 1111
apic1(config-leaf-template-route-profile)# set weight 20
```

VRF スコープ ルート プロファイルの設定

この手順では、「default-export」および「default-import」を含む VRF スコープのルート プロファイルを作成します。This route profile can be attached to a bridge domain (BD) while ‘matching’ a bridge-domain inside a route map through the **inherit route-profile** command.



- (注) VRF-scoped route profiles name **default-export** and **default-import** values, which are automatically applied on the match statements on the respective export/import route-maps used in the same tenant VRF.

始める前に

- テナントおよび VRF を設定します。
- リーフで VRF を有効にします。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	コンフィギュレーションモードに入ります。
ステップ 2	leaf node-id 例 : apicl(config)# leaf 101	設定するリーフを指定します。
ステップ 3	[no] vrf context tenant tenant-name vrf vrf-name 例 : apicl(config-leaf)# vrf context tenant exampleCorp vrf vrf1	リーフで VRF を有効にし、VRF コンフィギュレーションモードを開始します。
ステップ 4	[no] template route-profile profile-name 例 : apicl(config-leaf-vrf)# template route-profile default-export	VRF スコープのルートプロファイルを作成します。
ステップ 5	必須: [no] set community {regular extended} {no-advertise no-export 値 {none replace additive}} 例 : apicl(config-leaf-vrf-template-route-profile)# set community extended 20:22 additive	BGP コミュニティ属性を設定します。
ステップ 6	必須: [no] set local-preference value 例 : apicl(config-tenant-vrf-route-profile)# set local-preference 64	BGP ローカルプリファレンス値を設定します。範囲は 0 ～ 4294967295 です。
ステップ 7	必須: [no] set metric value 例 : apicl(config-tenant-vrf-route-profile)# set metric 128	宛先ルーティングプロトコルのメトリックを設定します。
ステップ 8	必須: [no] set metric-type {type-1 type-2} 例 : apicl(config-tenant-vrf-route-profile)# set metric-type type-2	オプションは次のとおりです。 • type-1 : OSPF 外部タイプ 1 メトリック • type-2 : OSPF 外部タイプ 2 メトリック

	コマンドまたはアクション	目的
ステップ 9	必須: [no] set tag name 例 : <pre>apic1(config-tenant-vrf-route-profile)# set tag 1111</pre>	宛先ルーティングプロトコルのタグ値を設定します。 <i>name</i> パラメータは符号なし整数です。
ステップ 10	必須: [no] set weight weight 例 : <pre>apic1(config-tenant-vrf-route-profile)# set weight 20</pre>	宛先ルーティングプロトコルのタグ値を設定します。 <i>weight</i> パラメータは符号なし整数です。

例

次に、VRF スコープのルート プロファイルを設定する例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf vrf1
apic1(config-leaf-vrf)# template route-profile default-export
apic1(config-leaf-vrf-template-route-profile)# set community extended 20:22 additive
apic1(config-leaf-vrf-template-route-profile)# set local-preference 64
apic1(config-leaf-vrf-template-route-profile)# set metric 128
apic1(config-leaf-vrf-template-route-profile)# set metric-type type-2
apic1(config-leaf-vrf-template-route-profile)# set tag 1111
apic1(config-leaf-vrf-template-route-profile)# set weight 20
```

ルート マップの作成

ルートマップは、外部ルータにアドバタイズされるブリッジドメインのパブリック サブネットを示すため、テナント単位のプレフィックスリストとともに作成されます。また、プレフィックスリストはすべての中継ルートが外部ルータにアドバタイズされるように作成する必要があります。中継ルートのプレフィックスリストは管理者によって設定されます。デフォルトの動作では、外部ルータへの中継ルートのアドバタイズメントはすべて拒否されます。

始める前に

テナントおよび VRF を設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>apic1# configure</pre>	コンフィギュレーションモードに入ります。

	コマンドまたはアクション	目的
ステップ 2	leaf node-id 例 : <pre>apic1(config)# leaf 101</pre>	設定するリーフを指定します。
ステップ 3	[no] vrf context tenant tenant-name vrf vrf-name 例 : <pre>apic1(config-leaf)# vrf context tenant exampleCorp vrf v1</pre>	ノードのテナント VRF を設定します。
ステップ 4	(任意) [no] router-id ipv4-address 例 : <pre>apic1(config-leaf-vrf)# router-id 1.2.3.4</pre>	VRF で実行されているルーティングプロトコルに対してルータ ID を割り当てます。ルータ ID を割り当てない場合、各リーフ スイッチに固有の ID が内部で生成されます。
ステップ 5	必須: [no] route-map name 例 : <pre>apic1(config-leaf-vrf)# route-map bgpMap</pre>	ルートマップを作成し、ルートマップコンフィギュレーションを開始します。
ステップ 6	必須: [no]ip permit le 32 リスト名プレフィックス/マスク長[{ prefix-list128}] 例 : <pre>apic1(config-leaf-vrf-route-map)# ip prefix-list list1 permit 13.13.13.0/24</pre>	ルートマップにプレフィックスリストを作成します。
ステップ 7	必須: [no] match prefix-list list-name 例 : <pre>apic1(config-leaf-vrf-route-map)# match prefix-list list1</pre>	作成済みのプレフィックスリストと照合し、一致モードを開始して、プレフィックスリストのルート制御プロファイルを設定します。
ステップ 8	必須: [no] set metric value 例 : <pre>apic1(config-leaf-vrf-route-map-match)# set metric 128</pre>	宛先ルーティングプロトコルのメトリックを設定します。
ステップ 9	必須: [no] set metric-type {type-1 type-2} 例 : <pre>apic1(config-leaf-vrf-route-map-match)# set metric-type type-2</pre>	オプションは次のとおりです。 • type-1 : OSPF 外部タイプ 1 メトリック • type-2 : OSPF 外部タイプ 2 メトリック

	コマンドまたはアクション	目的
ステップ 10	必須: [no] set local-preference value 例 : <pre>apic1(config-leaf-vrf-route-map-match) # set local-preference 64</pre>	BGP ローカルプリファレンス値を設定します。範囲は0～4294967295です。
ステップ 11	必須: [no] set community {regular extended} value {none replace additive} 例 : <pre>apic1(config-leaf-vrf-route-map-match) # set community extended 20:22 additive</pre>	BGP ルート アップデートのコミュニティ属性を設定します。 <i>community-value</i> を <i>aa:nn</i> 形式で指定します。次のいずれかのアクションを指定します。 • additive—既存のコミュニティに追加します • replace—既存のコミュニティを置き換えます • none—コミュニティを変更しません
ステップ 12	必須: [no] set tag name 例 : <pre>apic1(config-leaf-vrf-route-map-match) # set tag 1111</pre>	宛先ルーティングプロトコルのタグ値を設定します。 <i>name</i> パラメータは符号なし整数です。
ステップ 13	必須: [no] set weight value 例 : <pre>apic1(config-leaf-vrf-route-map-match) # set weight 32</pre>	ルーティング プロトコルの BGP 重みを指定します。
ステップ 14	必須: [no] contract {provider consumer} contract-name [imported] 例 : <pre>apic1(config-leaf-vrf-route-map-match) # contract provider prov 1</pre>	この VRF から（このプレフィックスリストと一致する）ルートをリークするために必要なコントラクトを追加します。
ステップ 15	必須: [no] match route group group-name [order number] 例 : <pre>apic1(config-leaf-vrf-route-map) # match route group g1 order 1</pre>	すでに作成されたルートグループを一致させ、一致モードを開始してルートマップを設定します。 手順 8-13 または手順 18 のみを繰り返して、ルートグループのルートマップを設定します。手順 17 を参照して、インラインセットアクションの代わりにルートマップを継承します。

	コマンドまたはアクション	目的
ステップ 16	必須: [no] match bridge-domain <i>bd-name</i> 例 : <pre>apicl(config-leaf-vrf-route-map) # bridge-domain bd1</pre>	プロトコルによってパブリック サブ ネットをエクスポートするため、ブリッジ ドメインと照合します。
ステップ 17	必須: [no] inherit route-profile <i>profile-name</i> 例 : <pre>apicl(config-leaf-vrf-route-map-match) # inherit route-profile rp1</pre>	ブリッジドメインのルートマップを設定します。 (注) ルート マップはすでに template route-profile コマンドを使用してすでに作成されています。
ステップ 18	必須: [no] bridge-domain-match 例 : <pre>apicl(config-leaf-vrf-route-map) # no bridge-domain-match</pre>	ブリッジドメインのルートマップを設定します。 (注) ルート マップでブリッジ ドメイン (BD) 一致を無効にして、ルートマップから BD 設定を削除する必要性をなくします。これは、BD がルートマップで一致して、ルートマップがルート グループ/明示プレフィックスリストを使用して BD サブネットをフィルタするために使用される場合に必要です。

例

次に、ルートマップを作成し、プレフィックスリスト、コミュニティリスト、およびブリッジドメインの追加/一致を行う例を示します。

```
# CREATE A ROUTE-MAP
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf v1
apicl(config-leaf-vrf)# route-map bgpMap

# CREATE A PREFIX-LIST
apicl(config-leaf-vrf-route-map)# ip prefix-list list1 permit 13.13.13.0/24
apicl(config-leaf-vrf-route-map)# ip prefix-list list1 permit 14.14.14.0/24

# MATCH THE PREFIX-LIST
apicl(config-leaf-vrf-route-map)# match prefix-list list1

# CONFIGURE A ROUTE-PROFILE FOR THE PREFIX-LIST
```

```

apic1(config-leaf-vrf-route-map-match)# set metric 128
apic1(config-leaf-vrf-route-map-match)# set metric-type type-2
apic1(config-leaf-vrf-route-map-match)# set local-preference 64
apic1(config-leaf-vrf-route-map-match)# set community extended 20:22 additive
apic1(config-leaf-vrf-route-map-match)# set tag 1111
apic1(config-leaf-vrf-route-map-match)# set weight 32
apic1(config-leaf-vrf-route-map-match)# contract provider prov 1

# CREATE COMMUNITY LIST
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# template community-list standard CL_1 65536:20 tenant exampleCorp

# CREATE ROUTE GROUP
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# template route group g1 tenant exampleCorp
apic1(config-route-group)# ip prefix permit 15.15.15.0/24
apic1(config-route-group)# community-list standard 65535:20

# MATCH ROUTE GROUP
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf)# route-map bgpMap
apic1(config-leaf-vrf-route-map)# match route group g1 order 1

# CONFIGURE ROUTE PROFILE FOR COMMUNITY-LIST
apic1(config-leaf-vrf-route-map-match)# set metric 128
apic1(config-leaf-vrf-route-map-match)# set metric-type type-2
apic1(config-leaf-vrf-route-map-match)# set local-preference 64
apic1(config-leaf-vrf-route-map-match)# set community extended 20:22 additive
apic1(config-leaf-vrf-route-map-match)# set tag 1111
apic1(config-leaf-vrf-route-map-match)# set weight 32

# CONFIGURE ROUTE PROFILE ROUTE GROUP
apic1(config-leaf-vrf-route-map-match)# set metric 128
apic1(config-leaf-vrf-route-map-match)# set metric-type type-2
apic1(config-leaf-vrf-route-map-match)# set local-preference 64
apic1(config-leaf-vrf-route-map-match)# set community extended 20:22 additive
apic1(config-leaf-vrf-route-map-match)# set tag 1111
apic1(config-leaf-vrf-route-map-match)# set weight 32

# Or CREATE A ROUTE PROFILE TEMPLATE AND INHERIT IT FOR ROUTE GROUP
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf)# template route-profile rp1
apic1(config-leaf-vrf-template-route-profile)# set metric 128
apic1(config-leaf-vrf-template-route-profile)# set metric-type type-2
apic1(config-leaf-vrf-template-route-profile)# set local-preference 64
apic1(config-leaf-vrf-template-route-profile)# set community extended 20:22 additive
apic1(config-leaf-vrf-template-route-profile)# set tag 1111
apic1(config-leaf-vrf-template-route-profile)# set weight 32
apic1(config-leaf-vrf-template-route-profile)# exit

apic1(config-leaf-vrf)# route-map bgpMap
apic1(config-leaf-vrf-route-map)# match route group g1 order 1
apic1(config-leaf-vrf-route-map-match)# inherit route-profile rp1

# CREATE A BRIDGE-DOMAIN
apic1# configure
apic1(config)# tenant exampleCorp

```



```

apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# interface bridge-domain bd1
apic1(config-tenant-interface)# ip address 13.13.13.1/24 scope public
apic1(config-tenant-interface)# exit
apic1(config-tenant)# exit

# CREATE A ROUTE-PROFILE FOR THE BRIDGE-DOMAIN
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf)# template route-profile default-export
apic1(config-leaf-vrf-template-route-profile)# set metric 128
apic1(config-leaf-vrf-template-route-profile)# set metric-type type-2
apic1(config-leaf-vrf-template-route-profile)# set local-preference 64
apic1(config-leaf-vrf-template-route-profile)# set community extended 20:22 additive
apic1(config-leaf-vrf-template-route-profile)# set tag 1111
apic1(config-leaf-vrf-template-route-profile)# set weight 20
apic1(config-leaf-vrf-template-route-profile)# exit

# MATCH THE BRIDGE-DOMAIN
apic1(config-leaf-vrf)# route-map bgpMap
apic1(config-leaf-vrf-route-map)# match bridge-domain bd1

# CONFIGURE A ROUTE-PROFILE FOR THE BRIDGE-DOMAIN
apic1(config-leaf-vrf-route-map-match)# inherit route-profile default-export

```

ルーティング プロトコルのルートマップの設定

OSPF、BGP、および EIGRP ルーティング プロトコルでは、インポートおよびエクスポートするルートフィルタ処理にルートマップを使用します。これらのプロトコルを設定するために必要な一般的な手順についてはそれぞれのドキュメントセクションを参照してください。これらのプロトコルでルートマップを設定するには、次のコマンドを使用し、例を参照してください。

プロトコル	ルートマップコマンド
BGP	[no] route-map <i>map-name</i> { in out }
OSPF	[no] area エリア <i>id</i> route-map マップ名 { in out }
EIGRP	[no] ip distribute list default route-map <i>map-name</i> out

例

次に、BGP、OSPF、および EIGRP でルートマップを設定する例を示します。

```

# BGP
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 100

```

```

apic1(config-bgp)# vrf member tenant exampleCorp vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 3.3.3.3
apic1(config-leaf-bgp-vrf-neighbor)# route-map map1 out
apic1(config-leaf-bgp-vrf-neighbor)# route-map map2 in
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-bgp)# exit
apic1(config-leaf)# exit

# OSPF
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant exampleCorp vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.1 route-map map1 out
apic1(config-leaf-ospf-vrf)# area 0.0.0.1 route-map map2 in
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit

#EIGRP
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1
apic1(config-leaf-if)# ip address 13.13.13.13/24
apic1(config-leaf-if)# ip router eigrp default
apic1(config-leaf-if)# ip distribute-list eigrp default route-map map1 out
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit

```

エクスポート マップ（インター VRF ルート リーク）の設定

始める前に

- ルートマップを作成します。
- リークさせる必要があるルートと一致するプレフィックスを含むプレフィックスリストをルートマップに追加します。
- プレフィックスリストを照合し、ルートリークを有効にするコントラクトを追加します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーション モードに入ります。

	コマンドまたはアクション	目的
ステップ 2	leaf <i>node-id</i> 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	[no] vrf context tenant <i>tenant-name</i> vrf <i>vrf-name</i> 例 : apic1(config-leaf)# vrf context tenant exampleCorp vrf v1	ノードのテナント VRF を設定します。
ステップ 4	[no] export map <i>map-name</i> 例 : apic1(config-leaf-vrf)# export map shared-route-map1	この VRF からコンシューマ VRF に (リーク) ルートをエクスポートするようこの VRF でルートマップを設定します。

例

次に、ルートマップを作成およびエクスポートする例を示します。

```
# CREATE A ROUTE-MAP
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf)# router-id 1.2.3.4
apic1(config-leaf-vrf)# route-map shared-route-map1
apic1(config-leaf-vrf-route-map)# ip prefix-list list1 permit 13.13.13.0/24
apic1(config-leaf-vrf-route-map)# match prefix-list list1
apic1(config-leaf-vrf-route-map-match)# contract provider prov1
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit

# EXPORT THE ROUTE-MAP
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf)# export map shared-route-map1
```

双方向ルート転送 (BFD) の設定

BFD について

双方向フォワーディング検出 (BFD) は、メディア タイプ、カプセル化、トポロジ、およびルーティングプロトコルのために短時間での転送パス障害検出を提供するために設計された検

出プロトコルです。BFD を使用することで、さまざまなプロトコルの Hello メカニズムにより、変動速度ではなく一定速度で転送パス障害を検出できます。BFD はプロファイリングおよびプランニングを簡単にし、再コンバージェンス時間の一貫性を保ち、予測可能にします。

双方向フォワーディング検出（BFD）を使用して、ピアリングルータの接続をサポートするように設定された ACI ファブリック境界リーフ スイッチ間の転送パスのサブセカンド障害検出時間を提供します。

BFD グローバルの設定

デバイスのすべての BFD セッションの BFD セッションパラメータを設定できます。BFD セッションパラメータは、スリーウェイ ハンドシェイクの BFD ピア間でネゴシエートされます。

グローバル BFD を設定するには、次の手順を実行します。

- グローバル BFD 構成設定を設定します
- アクセス リーフ ポリシー グループを設定し、以前作成したグローバル BFD ポリシーを継承します
- 以前作成したリーフ ポリシー グループをリーフ スイッチまたはリーフ スイッチのグループに関連付けます

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーションモードに入ります。
ステップ 2	[no] template bfd {ip ipv6} <i>global-policy-name</i> 例 : <code>apic1(config)# template bfd ip bfd_global</code>	BFD ポリシーテンプレートを作成します。
ステップ 3	[no] echo-address <i>ip-address</i> 例 : <code>apic1(config-bfd)# echo-address 192.0.20.123</code> <code>apic1(config-bfd)# echo-address 34::1/64</code>	BFD エコー パケットの送信元アドレスとして使用する IP アドレスを指定します。
ステップ 4	[no] slow-timer <i>milliseconds</i> 例 : <code>apic1(config-bfd)# slow-timer 2000</code>	エコー機能で使用する slow timer を設定します。この値はエコー機能がイネーブルの場合、BFD が新しいセッションを開始する速度および非同期

	コマンドまたはアクション	目的
		セッションが BFD 制御パケットに使用する速度を決定します。slow-timer 値は新しい制御パケット間隔として使用されますが、エコーパケットは設定された BFD 間隔を使用します。エコーパケットはリンク障害検出に使用されますが、低速の制御パケットは BFD セッションを維持します。指定できる範囲は 1000 ～ 30000 ミリ秒です。
ステップ 5	[no] min-tx milliseconds 例 : apicl (config-bfd) # min-tx 100	このデバイスが BFD Hello メッセージを送信する間隔を指定します。有効値は 50 ～ 999 ミリ秒です。
ステップ 6	[no] min-rx milliseconds 例 : apicl (config-bfd) # min-rx 70	このデバイスが別の BFD デバイスからの BFD Hello メッセージを受け付ける最小間隔を指定します。有効値は 50 ～ 999 ミリ秒です。
ステップ 7	[no] multiplier policy-name 例 : apicl (config-bfd) # multiplier 3	転送パスの障害を検出するまでに喪失した、別の BFD デバイスからの BFD Hello メッセージの数を指定します。指定できる範囲は 1 ～ 50 です。
ステップ 8	[no] echo-rx-interval policy-name 例 : apicl (config-bfd) # echo-rx-interval 500	このシステムがサポートできる受信 BFD エコーパケット間の最小間隔を指定します。有効値は 50 ～ 999 ミリ秒です。
ステップ 9	exit 例 : apicl (config-bfd) # exit	グローバル コンフィギュレーション モードに戻ります。
ステップ 10	[no] template leaf-policy-group leaf-policy-name 例 : apicl (config) # template leaf-policy-group leaf_pg1	アクセス スリーフ ポリシー グループを設定します。
ステップ 11	[no] inherit bfd {ip ipv6} global-policy-name 例 : apicl (config-leaf-policy-group) # inherit bfd ip bfd_global	以前作成したグローバルポリシーを継承します。

	コマンドまたはアクション	目的
ステップ 12	exit 例 : apic1(config-leaf-policy-group)# exit	グローバル コンフィギュレーション モードに戻ります。
ステップ 13	[no] leaf-profile leaf-profile-name 例 : apic1(config)# leaf-profile leaf_profile1	リーフ プロファイルを設定します。
ステップ 14	[no] leaf-group leaf-group-name 例 : apic1(config-leaf-profile)# leaf-group leaf_group1	リーフ スイッチのグループを作成または指定します。
ステップ 15	[no] leaf-policy-group leaf-policy-name 例 : apic1(config-leaf-group)# leaf-policy-group leaf_pg1	リーフ スイッチに関連付けられる、以前作成したリーフ ポリシー グループを指定します。
ステップ 16	[no] leaf leaf-range 例 : apic1(config-leaf-group)# leaf 101-102	リーフ スイッチ グループに 1 つ以上のリーフ スイッチを追加します。

例

この例では、グローバル BFD を設定し、リーフ スイッチのグループに適用する方法を示します。

```
# CONFIGURE BFD GLOBAL POLICIES
apic1# configure
apic1(config)# template bfd ip bfd_global
apic1(config-bfd)# echo-address 192.0.20.123
apic1(config-bfd)# slow-timer 2000
apic1(config-bfd)# min-tx 100
apic1(config-bfd)# min-rx 70
apic1(config-bfd)# multiplier 3
apic1(config-bfd)# echo-rx-interval 500
apic1(config-bfd)# exit

# CONFIGURE AN ACCESS LEAF POLICY GROUP AND INHERIT BFD GLOBAL POLICIES
apic1(config)# template leaf-policy-group leaf_pg1
apic1(config-leaf-policy-group)# inherit bfd ip bfd_global
apic1(config-leaf-policy-group)# exit

# CONFIGURE A LEAF GROUP AND ASSOCIATE THE LEAF POLICY GROUP
apic1(config)# leaf-profile leaf_profile1
apic1(config-leaf-profile)# leaf-group leaf_group1
apic1(config-leaf-group)# leaf-policy-group leaf_pg1
```

```
apicl(config-leaf-group)# leaf 101-102
```

NX-OS スタイル CLI を使用したリーフ スイッチでの BFD のグローバルな設定

手順

ステップ 1 NX-OS CLI を使用して BFD IPV4 グローバル設定 (bfdIpv4InstPol) を設定するには :

例 :

```
apicl# configure
apicl(config)# template bfd ip bfd_ipv4_global_policy
apicl(config-bfd)# [no] echo-address 1.2.3.4
apicl(config-bfd)# [no] slow-timer 2500
apicl(config-bfd)# [no] min-tx 100
apicl(config-bfd)# [no] min-rx 70
apicl(config-bfd)# [no] multiplier 3
apicl(config-bfd)# [no] echo-rx-interval 500
apicl(config-bfd)# exit
```

ステップ 2 NX-OS CLI を使用して BFD IPV6 グローバル設定 (bfdIpv6InstPol) を設定するには :

例 :

```
apicl# configure
apicl(config)# template bfd ipv6 bfd_ipv6_global_policy
apicl(config-bfd)# [no] echo-address 34::1/64
apicl(config-bfd)# [no] slow-timer 2500
apicl(config-bfd)# [no] min-tx 100
apicl(config-bfd)# [no] min-rx 70
apicl(config-bfd)# [no] multiplier 3
apicl(config-bfd)# [no] echo-rx-interval 500
apicl(config-bfd)# exit
```

ステップ 3 NX-OS CLI を使用してアクセス リーフ ポリシー グループ (infraAccNodePGrp) を設定し、以前に作成した BFD グローバル ポリシーを継承するには:

例 :

```
apicl# configure
apicl(config)# template leaf-policy-group test_leaf_policy_group
apicl(config-leaf-policy-group)# [no] inherit bfd ip bfd_ipv4_global_policy
apicl(config-leaf-policy-group)# [no] inherit bfd ipv6 bfd_ipv6_global_policy
apicl(config-leaf-policy-group)# exit
```

ステップ 4 NX-OS CLI を使用して以前に作成したリーフ ポリシー グループを リーフに関連付けるには:

例 :

```
apicl(config)# leaf-profile test_leaf_profile
apicl(config-leaf-profile)# leaf-group test_leaf_group
apicl(config-leaf-group)# leaf-policy-group test_leaf_policy_group
```

```
apic1(config-leaf-group)# leaf 101-102
apic1(config-leaf-group)# exit
```

NX-OS スタイル CLI を使用したスパイン スイッチ上の BFD のグローバル設定

次の手順を使用して、NX-OS スタイル CLI を使用してスパイン スイッチの BFD をグローバルに設定します。

手順

ステップ 1 NX-OS CLI を使用して BFD IPV4 グローバル設定 (bfdIpv4InstPol) を設定するには :

例 :

```
apic1# configure
apic1(config)# template bfd ip bfd_ipv4_global_policy
apic1(config-bfd)# [no] echo-address 1.2.3.4
apic1(config-bfd)# [no] slow-timer 2500
apic1(config-bfd)# [no] min-tx 100
apic1(config-bfd)# [no] min-rx 70
apic1(config-bfd)# [no] multiplier 3
apic1(config-bfd)# [no] echo-rx-interval 500
apic1(config-bfd)# exit
```

ステップ 2 NX-OS CLI を使用して BFD IPV6 グローバル設定 (bfdIpv6InstPol) を設定するには :

例 :

```
apic1# configure
apic1(config)# template bfd ipv6 bfd_ipv6_global_policy
apic1(config-bfd)# [no] echo-address 34::1/64
apic1(config-bfd)# [no] slow-timer 2500
apic1(config-bfd)# [no] min-tx 100
apic1(config-bfd)# [no] min-rx 70
apic1(config-bfd)# [no] multiplier 3
apic1(config-bfd)# [no] echo-rx-interval 500
apic1(config-bfd)# exit
```

ステップ 3 NX-OS CLI を使用してスパイン ポリシー グループを設定し以前作成した BFD グローバル ポリシーを継承するには :

例 :

```
apic1# configure
apic1(config)# template spine-policy-group test_spine_policy_group
apic1(config-spine-policy-group)# [no] inherit bfd ip bfd_ipv4_global_policy
apic1(config-spine-policy-group)# [no] inherit bfd ipv6 bfd_ipv6_global_policy
apic1(config-spine-policy-group)# exit
```

ステップ 4 NX-OS を使用して以前作成したスパイン ポリシー グループをスパイン スイッチに関連付けるには ;

例：

```
apicl# configure
apicl(config)# spine-profile test_spine_profile
apicl(config-spine-profile)# spine-group test_spine_group
apicl(config-spine-group)# spine-policy-group test_spine_policy_group
apicl(config-spine-group)# spine 103-104
apicl(config-leaf-group)# exit
```

グローバル BFD 設定のオーバーライド

BFD インターフェイス オーバーライド ポリシーの設定

明示的な BFD を設定できる、3 つのサポート対象のインターフェイス（ルーテッド L3 インターフェイス、外部インターフェイス SVI とルーテッドサブインターフェイス）があります。グローバルコンフィギュレーションを使用しないで、さらに特定のインターフェイスの明示的な設定をしたい場合、特定のスイッチまたは一連のすべてのインターフェイスに適用される独自のグローバルコンフィギュレーションを作成できます。特定のインターフェイス上の特定のスイッチの粒度がさらに必要な場合、このインターフェイス オーバーライド設定を使用する必要があります。

始める前に

テナントはすでに作成されています。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apicl# configure	コンフィギュレーションモードに入ります。
ステップ 2	tenant tenant-name 例： apicl(config)# tenant exampleCorp	設定するテナントを指定します。
ステップ 3	vrf context vrf-name 例： apicl(config-tenant)# vrf context vrf1	テナントと VRF を関連付けます。
ステップ 4	exit 例： apicl(config-tenant-vrf)# exit	テナント コンフィギュレーションモードに戻ります。

	コマンドまたはアクション	目的
ステップ 5	exit 例 : <code>apic1(config-tenant)# exit</code>	グローバル コンフィギュレーション モードに戻ります。
ステップ 6	leaf node-id 例 : <code>apic1(config)# leaf 101</code>	設定するリーフを指定します。
ステップ 7	[no] vrf context tenant tenant-name vrf vrf-name 例 : <code>apic1(config-leaf)# vrf context tenant exampleCorp vrf vrf1</code>	ノードのテナント VRF を設定します。
ステップ 8	exit 例 : <code>apic1(config-leaf-vrf)# exit</code>	リーフ コンフィギュレーション モードに戻ります。
ステップ 9	[no] interface type 例 : <code>apic1(config-leaf)# interface eth 1/18</code>	インターフェイス設定モードを開始します。
ステップ 10	[no] vrf member tenant tenant-name vrf vrf-name 例 : <code>apic1(config-leaf-if)# vrf member tenant exampleCorp vrf vrf1</code>	
ステップ 11	exit 例 : <code>apic1(config-leaf-if)# exit</code>	リーフ コンフィギュレーション モードに戻ります。
ステップ 12	[no] template bfd template-name tenant tenant-name 例 : <code>apic1(config-leaf)# template bfd bfdIfPol1 tenant exampleCorp</code>	BFD インターフェイス ポリシーを設定します。
ステップ 13	[no] echo-mode enable 例 : <code>apic1(config-template-bfd-pol)# echo-mode enable</code>	BFD 制御パケットに加えて BFD エコーパケットを送信をイネーブルまたはディセーブルにします。

	コマンドまたはアクション	目的
ステップ 14	[no] echo-rx-interval <i>policy-name</i> 例 : <pre>apicl(config-template-bfd-pol)# echo-rx-interval 500</pre>	このシステムがサポートできる受信 BFD エコー パケット間の最小間隔を指定します。有効値は 50 ～ 999 ミリ秒です。
ステップ 15	[no] min-tx milliseconds 例 : <pre>apicl(config-template-bfd-pol)# min-tx 100</pre>	このデバイスが、BFD hello メッセージを送信する間隔を指定します。有効値は 50 ～ 999 ミリ秒です。
ステップ 16	[no] min-rx milliseconds 例 : <pre>apicl(config-template-bfd-pol)# min-rx 70</pre>	このデバイスが別の BFD デバイスからの BFD Hello メッセージを受け付ける最小間隔を指定します。有効値は 50 ～ 999 ミリ秒です。
ステップ 17	[no] multiplier <i>policy-name</i> 例 : <pre>apicl(config-template-bfd-pol)# multiplier 5</pre>	転送パスの障害を検出するまでに喪失した、別の BFD デバイスからの BFD Hello メッセージの数を指定します。指定できる範囲は 1 ～ 50 です。
ステップ 18	[no] optimize subinterface 例 : <pre>apicl(config-template-bfd-pol)# optimize subinterface</pre>	サブインターフェイスの最適化をイネーブ爾またはディセーブ爾にします。BFD により、設定されているすべてのサブインターフェイスのセッションが作成されます。BFD により、設定されている最小の VLAN ID を持つサブインターフェイスがマスターサブインターフェイスとして設定され、そのサブインターフェイスは親インターフェイスの BFD セッションパラメータを使用します。残りのサブインターフェイスは slow timer を使用します。最適化されたサブインターフェイスセッションでエラーが検出されると、BFD により、その物理インターフェイスのすべてのサブインターフェイスがダウンとマークされます。

例

この例では、BFD オーバーライドポリシーを作成し、インターフェイスに適用する方法を示します。

```
apicl# configure
```

```

apic1(config)# tenant exampleCorp
apic1(config-tenant)# vrf context vrf1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf vrf1
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface eth 1/18
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf vrf1
apic1(config-leaf-if)# exit

# CONFIGURE BFD INTERFACE OVERRIDE POLICY
apic1(config-leaf)# template bfd bfdIfPol1 tenant exampleCorp
apic1(config-template-bfd-pol)# echo-mode enable
apic1(config-template-bfd-pol)# echo-rx-interval 500
apic1(config-template-bfd-pol)# min-tx 100
apic1(config-template-bfd-pol)# min-rx 70
apic1(config-template-bfd-pol)# multiplier 5
apic1(config-template-bfd-pol)# optimize subinterface

```

インターフェイスへの BFD インターフェイス オーバーライド ポリシーの適用

ルーテッド L3 インターフェイス、外部 SVI インターフェイスとルーテッドサブインターフェイスに、BFD インターフェイス オーバーライド ポリシーを適用できます。

始める前に

BFD インターフェイス オーバーライド ポリシーがすでに作成されています。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーション モードに入ります。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	[no] interface type 例 : apic1(config-leaf)# interface Ethernet 1/15	インターフェイス設定モードを開始します。サポートされているインターフェイスは、ルーテッド L3 インターフェイス、外部 SVI インターフェイスとルーテッドサブインターフェイスです。
ステップ 4	[no] ipv6 address ipv6-address [preferred] 例 : apic1(config-leaf-if)# ipv6 address 2001::10:1/64 preferred	インターフェイスからのトラフィックのデフォルトの発信元アドレスである IP アドレスを指定します。

	コマンドまたはアクション	目的
		(注) このコマンドは、インターフェイスが IPv6 インターフェイスである場合にのみ使用されます。
ステップ 5	[no] vrf member tenant <i>tenant-name</i> vrf <i>vrf-name</i> 例 : <pre>apic1(config-leaf-if)# vrf member tenant exampleCorp vrf vrf1</pre>	テナント VRF にインターフェイスを接続します。 (注) このコマンドは、インターフェイスが VLAN インターフェイスである場合にのみ使用されます。
ステップ 6	bfd {ip ipv6} tenant mode 例 : <pre>apic1(config-leaf-if)# bfd ip tenant mode</pre>	BFD テナント モードをイネーブルにします。
ステップ 7	bfd {ip ipv6} inherit interface-policy <i>policy-name</i> 例 : <pre>apic1(config-leaf-if)# bfd ip inherit interface-policy bfdIfPoll</pre>	指定された BFD インターフェイス テンプレート ポリシーを継承します。
ステップ 8	bfd {ip ipv6} authentication keyed-sha1 <i>keyid</i> <i>keyid</i> <i>key</i> <i>key</i> 例 : <pre>apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password</pre>	キー付き SHA-1 として BFD 認証を設定します。

例

この例では、IPv4 アドレスを持つ L3 インターフェイス上に以前作成した BFD インターフェイス ポリシーを継承する方法を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/15
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdIfPoll
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password
```

この例では、IPv6 アドレスを持つ L3 インターフェイス上に以前作成した BFD インターフェイス ポリシーを継承する方法を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/15
apic1(config-leaf-if)# ipv6 address 2001::10:1/64 preferred
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdIfPoll
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password

```

この例では、IPv4 アドレスを持つ VLAN インターフェイス上で BFD を設定する方法を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 15
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf vrf1
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdIfPoll
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password

```

この例では、IPv6 アドレスを持つ VLAN インターフェイス上で BFD を設定する方法を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 15
apic1(config-leaf-if)# ipv6 address 2001::10:1/64 preferred
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf vrf1
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdIfPoll
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password

```

コンシューマ プロトコル上の BFD の有効化

これらの手順は、BFD 機能を利用する 4 つのコンシューマ プロトコル（BGP、EIGRP、OSPF、静的ルート）で BFD を有効にする手順を紹介します。

BGP コンシューマ プロトコルの BFD の有効化

始める前に

テナントはすでに作成されています。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	bgp-fabric 例 : <code>apic1(config-bgp-fabric)# bgp-fabric</code>	ファブリックの BGP 設定モードを開始します。
ステップ 3	asn <i>asn-number</i> 例 : <code>apic1(config-bgp-fabric)# asn 200</code>	BGP 自律システム番号 (ASN) を指定します。
ステップ 4	exit 例 : <code>apic1(config-bgp-fabric)# exit</code>	グローバル コンフィギュレーション モードに戻ります。
ステップ 5	leaf <i>node-id</i> 例 : <code>apic1(config)# leaf 101</code>	設定するリーフを指定します。
ステップ 6	router bgp <i>asn-number</i> 例 : <code>apic1(config-leaf)# router bgp 200</code>	BGP ポリシー設定を入力します。
ステップ 7	vrf member tenant <i>tenant-name</i> vrf <i>vrf-name</i> 例 : <code>apic1(config-bgp)# vrf member tenant exampleCorp vrf v100</code>	後続のポリシー設定モード コマンドと関連付ける VRF インスタンスを指定します。
ステップ 8	neighbor <i>ip-address</i>[/<i>masklength</i>] 例 : <code>apic1(config-leaf-bgp-vrf)# neighbor 1.2.3.4</code>	ネイバーの IP アドレスを指定します。 マスク長は 32 です。
ステップ 9	[no] bfd enable 例 : <code>apic1(config-leaf-bgp-vrf-neighbor)# bfd enable</code>	BGP コンシューマ プロトコルで BFD を有効または無効にします。

例

次に、BGP コンシューマ プロトコルで BFD を有効にする例を示します。

```
apic1# configure
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 200
apic1(config-bgp-fabric)# exit
```

```

apic1(config)# leaf 101
apic1(config-leaf)# router bgp 200
apic1(config-bgp)# vrf member tenant exampleCorp vrf v100
apic1(config-leaf-bgp-vrf)# neighbor 1.2.3.4
apic1(config-leaf-bgp-vrf-neighbor)# bfd enable

```

EIGRP コンシューマ プロトコルの BFD の有効化

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーション モードに入ります。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	[no] interface type 例 : apic1(config-leaf)# interface Ethernet 1/15	インターフェイス設定モードを開始します。
ステップ 4	[no] {ip ipv6} bfd eigrp enable 例 : apic1(config-leaf-if)# ip bfd eigrp enable	EIGRP コンシューマ プロトコルで BFD を有効または無効にします。

例

次に、EIGRP コンシューマ プロトコルで BFD を有効にする例を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/15
apic1(config-leaf-if)# ip bfd eigrp enable

```


OSPF コンシューマ プロトコルの BFD の有効化

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーション モードに入ります。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	[no] interface type 例 : apic1(config-leaf)# interface vlan 123	インターフェイス設定モードを開始します。
ステップ 4	[no] ip ospf bfd enable 例 : apic1(config-leaf-if)# ip ospf bfd enable	OSPF コンシューマ プロトコルで BFD を有効または無効にします。

例

次に、OSPF コンシューマ プロトコルで BFD を有効にする例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 123
apic1(config-leaf-if)# ip ospf bfd enable
```

静的ルート コンシューマ プロトコルの BFD の有効化

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーション モードに入ります。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。

	コマンドまたはアクション	目的
ステップ 3	[no] vrf context tenant <i>tenant-name</i> vrf <i>vrf-name</i> 例 : <pre>apic1(config-leaf)# vrf context tenant exampleCorp vrf vrf1</pre>	ノードのテナント VRF を設定します。
ステップ 4	[no] {ip ipv6} route <i>ip-prefix/masklen</i> <i>next-hop-address</i> bfd 例 : <pre>apic1(config-leaf-vrf)# ip route 10.0.0.1/16 10.0.0.5 bfd</pre>	静的ルートのコンシューマ プロトコルで BFD を有効または無効にします。

例

次に、静的ルートのコンシューマ プロトコルで BFD を有効にする例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf vrf1
apic1(config-leaf-vrf)# ip route 10.0.0.1/16 10.0.0.5 bfd
```

NX-OS スタイルの CLI を使用した BFD コンシューマ プロトコルの設定

手順

ステップ 1 NX-OS は、CLI を使用して、BGP コンシューマ プロトコルを BFD をイネーブルにします。

例 :

```
apic1# configure
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 200
apic1(config-bgp-fabric)# exit
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 200
apic1(config-bgp)# vrf member tenant t0 vrf v0
apic1(config-leaf-bgp-vrf)# neighbor 1.2.3.4
apic1(config-leaf-bgp-vrf-neighbor)# [no] bfd enable
```

ステップ 2 NX-OS は、CLI を使用して、EIGRP コンシューマ プロトコルを BFD をイネーブルにします。

例 :

```
apic1(config-leaf-if)# [no] ip bfd eigrp enable
```

ステップ 3 NX-OS は、CLI を使用して、OSPF コンシューマ プロトコルを BFD をイネーブルにします。

例：

```
apic1(config-leaf-if)# [no] ip ospf bfd enable

apic1# configure
apic1(config)# spine 103
apic1(config-spine)# interface ethernet 5/3.4
apic1(config-spine-if)# [no] ip ospf bfd enable
```

ステップ 4 NX-OS は、CLI を使用して、スタティック ルート コンシューマ プロトコルを BFD をイネーブルにします。

例：

```
apic1(config-leaf-vrf)# [no] ip route 10.0.0.1/16 10.0.0.5 bfd

apic1(config)# spine 103
apic1(config-spine)# vrf context tenant infra vrf overlay-1
apic1(config-spine-vrf)# [no] ip route 21.1.1.1/32 32.1.1.1 bfd
```

ステップ 5 NX-OS は、CLI を使用して、IS-IS コンシューマ プロトコルを BFD をイネーブルにします。

例：

```
apic1(config)# leaf 101
apic1(config-spine)# interface ethernet 1/49
apic1(config-spine-if)# isis bfd enabled
apic1(config-spine-if)# exit
apic1(config-spine)# exit

apic1(config)# spine 103
apic1(config-spine)# interface ethernet 5/2
apic1(config-spine-if)# isis bfd enabled
apic1(config-spine-if)# exit
apic1(config-spine)# exit
```

レイヤ 3 マルチキャストの設定

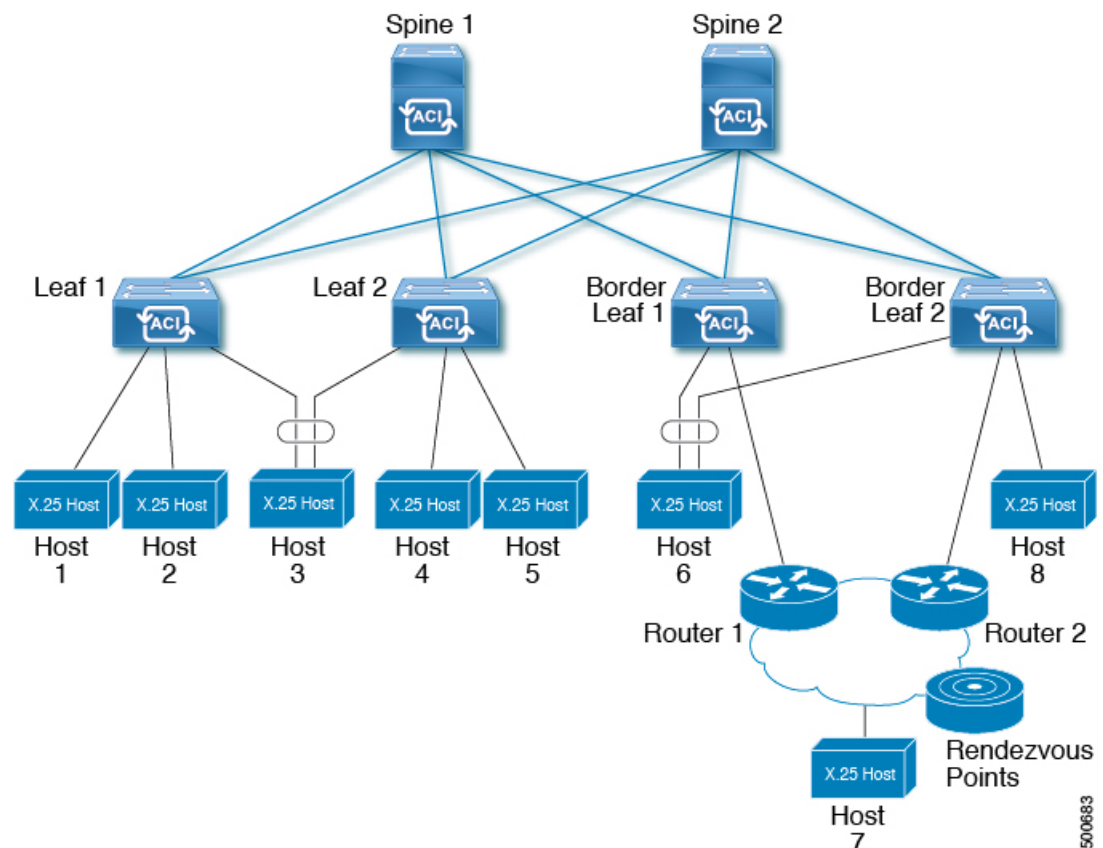
レイヤ 3 マルチキャスト

ACI ファブリックでは、ほとんどのユニキャストとマルチキャスト ルーティングが同じ境界 リーフ スイッチで稼働しており、ユニキャスト ルーティング プロトコル上でマルチキャスト プロトコルが稼働しています。

このアーキテクチャでは、ボーダー リーフ スイッチのみが完全な Protocol Independent Multicast (PIM) プロトコルを実行します。非ボーダー リーフ スイッチは、インターフェイス上でパッシブ モードの PIM を実行します。これらは、その他の PIM ルータとピアリングしません。ボーダー リーフ スイッチは、L3 Out を介してそれらの接続された他の PIM ルータとピアリングし、またそれら相互にもピアリングします。

次の図に、マルチキャストクラウド内のルータ（R1 と R2）に接続しているボーダー リーフ（BL）スイッチを示します。マルチキャストルーティングを必要とするファブリック内の各 Virtual Routing and Forwarding（VRF）は、それぞれ別に外部マルチキャストルータとピアリングします。

図 22: マルチキャストクラウドの概要



レイヤ 3 マルチキャストの設定に関するガイドライン

次のガイドラインを参照してください。

- レイヤ 3 マルチキャストの設定は VRF レベルで実行されます。そのため、VRF 内とマルチキャスト内のプロトコル機能が VRF で有効になり、各マルチキャスト VRF を個別にオンまたはオフにすることができます。
- マルチキャストで VRF が有効になると、有効になった VRF の個別のブリッジドメイン (BD) と L3 Out を有効にしてマルチキャストを設定できます。デフォルトでは、マルチキャストはすべての BD およびレイヤ 3 Out で無効になっています。
- 現時点では、レイヤ 3 マルチキャストは、共有 L3 Out で設定された VRF ではサポートされていません。
- Any Source Multicast (ASM) と Source-Specific Multicast (SSM) はサポートされています。

- 現時点では、双方向 PIM、ACI ファブリック内のランデブー ポイント (RP)、および PIM IPv6 はサポートされていません。
- IGMP スヌーピングは、マルチキャストルーティングが有効になっているパーペシブブリッジドメインでは無効にできません。
- マルチキャストルータは、パーペシブブリッジドメインではサポートされていません。
- 次の -EX でレイヤ 3 のマルチキャスト機能がサポートされているリーフ スイッチのモデルします。

- N9K-93180YC-EX

- N9K-93108TC-EX

- N9K-93180LC-EX

- レイヤ 3 ポートとサブインターフェイスはサポートされていますが、外部 SVI はサポートされていません。外部 SVI がサポートされていないため、PIM を L3-VPC で有効にできません。
- マルチポッドのレイヤ 3 マルチキャスト サポートについて、イングレス リーフ スイッチはマルチキャストルーティングに対応しているブリッジドメインに接続されたソースからパケットを受信するとき、イングレスリーフスイッチはファブリックにルート済み VRF コピーのみ送信します（ルート済みとは TTL が 1 に減少し、source-mac が拡散型サブネット MAC に再度書き込みされます）。また、出力リーフ スイッチも、関連するすべてのブリッジドメイン内の受信者へパケットをルーティングします。そのため、受信者のブリッジドメインが送信元と同じで、リーフ スイッチが送信元とは異なる場合、その受信者は同じブリッジドメイン内であっても、ルーティングされたコピーを受け取り続けます。

詳細については、次のリンクで、既存のレイヤ 2 設計を活用する multipod サポート レイヤ 3 マルチキャストに関する詳細情報を参照してください。 [追加ポッド](#)。

- リリース 3.1(1x) で始まる、FEX にマルチキャストのレイヤ 3 はサポートされています。マルチキャストのソースまたは FEX ポートに接続されているレシーバがサポートされません。詳細については、テスト環境で FEX を追加する方法について、設定、次の URL をアプリケーションセントリック インフラストラクチャとファブリック エクステンダを参照してください: <https://www.cisco.com/c/en/us/support/docs/cloud-systems-management/application-policy-infrastructure-controller-apic/200529-Configure-a-Fabric-Extender-with-Applica.html>。リリース 3.1(1x) 以降のレイヤ 3 マルチキャストでは FEX がサポートされていません。マルチキャストのソースまたは FEX ポートに接続されているレシーバはサポートされていません。



- (注) Cisco ACI は、IP フラグメンテーションをサポートしていません。したがって、外部ルータへのレイヤ 3 Outside (L3Out) 接続、または Inter-Pod Network (IPN) を介した multipod 接続を設定する場合は、MTU が両側で適切に設定されていることが重要です。ACI、Cisco NX-OS、Cisco IOS などの一部のプラットフォームでは、設定された MTU 値は IP ヘッダーを考慮に入れています（結果として、最大パケットサイズは、ACI で 9216 バイト、NX-OS および IOS で 9000 バイトに設定されます）。ただし、IOS XR などの他のプラットフォームは、パケットヘッダーのを除く MTU 値を設定します（結果として最大パケットサイズは 8986 バイトになります）。

各プラットフォームの適切な MTU 値については、それぞれの設定ガイドを参照してください。

CLI ベースのコマンドを使用して MTU をテストすることを強く推奨します。たとえば、Cisco NX-OS CLI で `ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1` などのコマンドを使用します。

レイヤ 3 マルチキャストの設定手順

ここでは、レイヤ 3 マルチキャストの設定手順を示します。手順は次のとおりです。

1. テナント VRF の PIM オプションを設定します。
2. VRF の IGMP オプションを設定します。
3. テナントに L3 Out 設定し、PIM を有効にしてリーフ インターフェイスを設定します。
4. 目的のブリッジ ドメイン内で PIM を有効にします。

レイヤ 3 マルチキャストの PIM オプションの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： <code>apic1# configure</code>	コンフィギュレーションモードに入ります。
ステップ 2	tenant tenant-name 例： <code>apic1(config)# tenant exampleCorp</code>	設定するテナントを指定します。
ステップ 3	vrf context vrf-name 例： <code>apic1(config-tenant)# vrf context exampleCorp_vrf1</code>	テナントと VRF を関連付けます。

	コマンドまたはアクション	目的
ステップ 4	[no] ip pim 例 : <pre>apicl(config-tenant-vrf)# ip pim</pre>	Protocol Independent Multicast (PIM) を設定します。
ステップ 5	(任意) [no] ip pim auto-rp {forward [listen] listen mapping-agent-policy マッピングエージェントポリシー名} 例 : <pre>apicl(config-tenant-vrf)# ip pim auto-rp forward listen</pre>	PIM 自動 RP (ランデブー ポイント) オプションを設定します。Auto-RP は、PIM ネットワークにおけるグループから RP へのマッピングの配信を自動化します。自動 RP メッセージの転送、自動 RP メッセージのリッスン、またはマッピングエージェントメッセージのフィルタリングに対するルートマップ ポリシーの関連付けを選択できます。
ステップ 6	(任意) [no] ip pim bsr {forward [listen] listen bsr-policy mapping-agent-policy-name} 例 : <pre>apicl(config-tenant-vrf)# ip pim bsr forward listen</pre>	PIM ブートストラップ ルータ (BSR) オプションを設定します。BSR は、RP 機能およびグループの RP 情報のリレーに候補ルータを使用するという点において自動 RP と同様に動作します。RP 情報は、PIM メッセージ内で伝送される BSR メッセージを通じて配信されます。ブートストラップ/RP 候補メッセージの転送、ブートストラップ/RP 候補メッセージのリッスン、または BSR メッセージのフィルタリングに対するルートマップ ポリシーの関連付けを選択できます。
ステップ 7	(任意) [no] ip pim fast-convergence 例 : <pre>apicl(config-tenant-vrf)# ip pim fast-convergence</pre>	PIM 高速コンバージェンス機能を有効にして、スイッチが応答しないネイバーを迅速に検出できるようにします。
ステップ 8	(任意) [no] ip pim mtu mtu-size 例 : <pre>apicl(config-tenant-vrf)# ip pim mtu 1500</pre>	PIM メッセージの最大サイズを設定します。有効な範囲は 1500 ~ 65536 バイトです。
ステップ 9	(任意) [no] ip pim register-policy register-policy-name 例 : <pre>apicl(config-tenant-vrf)# ip pim register-policy regPolicy1</pre>	登録メッセージのフィルタリングにポリシーの名前を指定します。

	コマンドまたはアクション	目的
ステップ 10	(任意) [no] ip pim register-rate-limit <i>mtu-size</i> 例 : <pre>apic1(config-tenant-vrf)# ip pim register-rate-limit 1024</pre>	PIM データ レジスタにレート制限を指定します。範囲は毎秒 0 ～ 65535 パケットです。
ステップ 11	(任意) [no] ip pim register-source <i>ip-address</i> 例 : <pre>apic1(config-tenant-vrf)# ip pim register-source 192.0.20.123</pre>	PIM メッセージの送信元 IP アドレスを設定します。
ステップ 12	(任意) [no] ip pim rp-address <i>ip-address</i> [<i>route-map route-map-name</i>] 例 : <pre>apic1(config-tenant-vrf)# ip pim rp-address 192.0.20.99</pre>	マルチキャストグループ範囲に、静的ルートプロセッサ (RP) アドレスを設定します。
ステップ 13	(任意) [no] ip pim sg-expiry-timer <i>ip-address</i> [<i>sg-list route-map-name</i>] 例 : <pre>apic1(config-tenant-vrf)# ip pim sg-expiry-timer 4096</pre>	PIM スパース モード (PIM-SM) (S,G) マルチキャスト ルートに (S,G) 期限切れタイマーの間隔を設定します。値の範囲は 180 ～ 604801 秒です。The optional sg-list parameter specifies S,G values to which the timer applies. デフォルトは 4096 です。
ステップ 14	(任意) [no] ip pim ssm route-map <i>route-map-name</i> 例 : <pre>apic1(config-tenant-vrf)# ip pim ssm route-map SSMRtMap</pre>	IP マルチキャストの拡張機能である Source Specific Multicast (SSM) を設定します。この機能を使用すると、受信者に転送されるデータグラムトラフィックは、その受信者が明示的に加入しているマルチキャスト送信元からのトラフィックだけになります。ルートマップポリシーは、グループのプレフィックスのリストを表示します。
ステップ 15	(任意) [no] ip pim state-limit <i>max-entries</i> [reserved <i>route-map-name</i> [<i>maximum-reserve-state-entries</i>]] 例 : <pre>apic1(config-tenant-vrf)# ip pim state-limit 100000 reserved myReservedPolicy 40000</pre>	現在の VRF インスタンスの PIM の状態エントリの最大数を設定します。最大状態エントリ数の範囲は 0 ～ 4294967295 です。オプションで、ポリシーマップに指定されるルート用に予約する状態エントリ数を指定することができ、また、この VRF で許可される最大予約済み (*,G) と (S,G) エントリを指定できます。子の数は、許可

	コマンドまたはアクション	目的
		される最大状態数以下である必要があります。範囲は 1 ～ 4294967295 です。
ステップ 16	(任意) [no] ip pim use-shared-tree-only group-list policy-name 例 : <pre>apicl(config-tenant-vrf)# ip pim use-shared-tree-only group-list myGroup1</pre>	PIM (*,G) 状態のみを作成します (送信元の状態を作成しない場合)。このポリシーで、この機能を適用するグループプレフィックスを定義します。
ステップ 17	exit 例 : <pre>apicl(config-tenant-vrf)# exit</pre>	テナント コンフィギュレーションモードに戻ります。

次のタスク

VRF の IGMP オプションを設定します。

レイヤ 3 マルチキャストの VRF での IGMP オプションの設定

始める前に

テナント VRF の PIM オプションを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>apicl# configure</pre>	コンフィギュレーションモードに入ります。
ステップ 2	tenant tenant-name 例 : <pre>apicl(config)# tenant exampleCorp</pre>	設定するテナントを指定します。
ステップ 3	vrf context vrf-name 例 : <pre>apicl(config-tenant)# vrf context vrf1</pre>	テナントと VRF を関連付けます。
ステップ 4	[no] ip igmp 例 : <pre>apicl(config-tenant-vrf)# ip igmp</pre>	インターネットグループ管理プロトコル (IGMP) を有効にします。

	コマンドまたはアクション	目的
ステップ 5	exit 例 : <code>apic1(config-tenant-vrf)# exit</code>	テナント コンフィギュレーション モードに戻ります。
ステップ 6	interface bridge-domain <i>bd-name</i> 例 : <code>apic1(config-tenant)# interface bridge-domain exampleCorp_bd1</code>	テナント インターフェイス コンフィギュレーション モードを開始してブリッジ ドメインを設定します。
ステップ 7	[no] ip multicast 例 : <code>apic1(config-tenant-interface)# ip multicast</code>	インターフェイスで IP マルチキャスト ルーティングを有効にします。
ステップ 8	[no] ip igmp allow-v3-asm 例 : <code>apic1(config-tenant-interface)# ip igmp allow-v3-asm</code>	Any Source Multicast (ASM) グループに関する IGMPv3 レポートの送信元アドレスのフィルタリングを許可します。
ステップ 9	[no] ip igmp fast-leave 例 : <code>apic1(config-tenant-interface)# ip igmp fast-leave</code>	IP IGMP スヌーピングの高速脱退処理を有効にします。この機能は、IGMPv2 プロトコルのホスト レポート抑制メカニズムのために明示的に追跡できない IGMPv2 ホストをサポートします。高速脱退が有効になっている場合、IGMP ソフトウェアは、各ポートに接続されたホストが 1 つだけであると見なします。
ステップ 10	[no] ip igmp group-timeout <i>seconds</i> 例 : <code>apic1(config-tenant-interface)# ip igmp group-timeout 260</code>	IGMPv2 のグループ メンバーシップ タイムアウトを設定します。値の範囲は 3 ～ 65535 秒です。デフォルト値は 260 秒です。
ステップ 11	[no] ip igmp inherit interface-policy <i>policy-name</i> 例 : <code>apic1(config-tenant-interface)# ip igmp inherit interface-policy MyIfPolicy</code>	このインターフェイスに IGMP インターフェイス ポリシーを関連付けます。
ステップ 12	[no] ip igmp join-group route-map <i>route-map-name</i> 例 :	1 つ以上のマルチキャスト グループをインターフェイスに静的にバインドします。ルート マップ ポリシーは、グループのプレフィックス、グループの範

	コマンドまたはアクション	目的
	<code>apicl(config-tenant-interface)# ip igmp join-group route-map MyGroupsRMap</code>	囲、および送信元プレフィックスのリストを表示します。
ステップ 13	[no] ip igmp last-member-query-count <i>count</i> 例 : <code>apicl(config-tenant-interface)# ip igmp last-member-query-count 2</code>	ホストの Leave メッセージを受信してから、IGMP クエリーが送信される回数を設定します。範囲は 1 ～ 5 クエリーです。デフォルトは 2 クエリーです。
ステップ 14	[no] ip igmp last-member-query-response-time <i>seconds</i> 例 : <code>apicl(config-tenant-interface)# ip igmp last-member-query-response-time 1</code>	メンバーシップレポートを送信してから、ソフトウェアがグループステートを解除するまでのクエリーインターバルを設定します。値の範囲は 1 ～ 25 秒です。デフォルト値は 1 秒です。
ステップ 15	[no] ip igmp querier-timeout <i>seconds</i> 例 : <code>apicl(config-tenant-interface)# ip igmp querier-timeout 255</code>	前クエリアがクエリーを停止してから、自身がクエリアとして処理を引き継ぐまで、ソフトウェアが待機する秒数を設定します。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は 255 秒です。
ステップ 16	[no] ip igmp query-interval <i>seconds</i> 例 : <code>apicl(config-tenant-interface)# ip igmp query-interval 125</code>	IGMP ホスト クエリー メッセージの送信頻度を設定します。大きな値を設定すると、ソフトウェアによる IGMP クエリーの送信頻度が低くなるため、ネットワーク上の IGMP メッセージ数を調整できます。指定できる範囲は 1 ～ 18000 秒です。デフォルト値は 125 秒です。
ステップ 17	[no] ip igmp query-max-response-time <i>seconds</i> 例 : <code>apicl(config-tenant-interface)# ip igmp query-max-response-time 10</code>	IGMP クエリーでアドバタイズされる応答時間を設定します。大きな値を設定すると、ホストの応答時間が延長されるため、ネットワークの IGMP メッセージのバースト性を調整できます。この値は、クエリーインターバルよりも短く設定する必要があります。値の範囲は 1 ～ 25 秒です。デフォルトは 10 秒です。
ステップ 18	[no] ip igmp report-link-local-groups 例 : <code>apicl(config-tenant-interface)# ip igmp report-link-local-groups</code>	224.0.0.0/24 に含まれるグループに対して、レポート送信をイネーブルにします。リンクローカルアドレスは、ローカルネットワークプロトコルだけで使

	コマンドまたはアクション	目的
		用されます。非リンク ローカルグループには、常にレポートが送信されます。デフォルトでは、リンク ローカルグループにレポートは送信されません。
ステップ 19	[no] ip igmp report-policy <i>policy-name</i> 例 : <pre>apic1(config-tenant-interface)# ip igmp report-policy MyReportPolicy</pre>	ルートマップポリシーに基づく、IGMP レポートのアクセスポリシーを設定します。
ステップ 20	[no] ip igmp robustness-variable <i>value</i> 例 : <pre>apic1(config-tenant-interface)# ip igmp robustness-variable 2</pre>	輻輳ネットワーク上のパケット損失を補うようにロバストネス変数を設定します。ロバストネス値は、IGMP ソフトウェアがメッセージ送信回数を確認するために使用されます。ネットワークのパケット損失が多い場合は、この値を大きくします。指定できる範囲は 1 ～ 7 です。デフォルトは 2 です。
ステップ 21	[no] ip igmp snooping 例 : <pre>apic1(config-tenant-interface)# ip igmp snooping</pre>	インターフェイスに対して IGMP スヌーピングを有効にします。
ステップ 22	[no] ip igmp snooping fast-leave 例 : <pre>apic1(config-tenant-interface)# ip igmp snooping fast-leave</pre>	ソフトウェアが IGMP Leave レポートを受信した場合に、IGMP クエリーメッセージを送信することなく、グループステートを解除できるようにします。このパラメータは、IGMPv2 ホストに関して、各ポート上のホストが 1 つしか存在しない場合に使用されます。
ステップ 23	[no] ip igmp snooping last-member-query-interval 例 : <pre>apic1(config-tenant-interface)# ip igmp snooping last-member-query-interval 5</pre>	時間間隔を秒単位で設定します。この時間が経過しても IGMP クエリーメッセージにホストが 1 つも応答しない場合は、関連するポートからこのグループが削除されます。値の範囲は 1 ～ 25 秒です。デフォルトは 5 秒です。
ステップ 24	[no] ip igmp snooping policy <i>policy-name</i> 例 : <pre>apic1(config-tenant-interface)# ip igmp snooping policy MySnoopingPolicy</pre>	IGMP スヌーピングポリシーとブリッジドメインを関連付けます。

	コマンドまたはアクション	目的
ステップ 25	[no] ip igmp snooping querier 例 : <pre>apicl(config-tenant-interface) # ip igmp snooping querier</pre>	IP IGMP スヌーピング クエリアを有効にして、IP マルチキャストトラフィックを受信するホストから IGMP レポートメッセージをトリガーする IGMP クエリーを定期的に送信します。IGMP スヌーピングはこれらの IGMP レポートを待ち受けて、適切な転送を確立します。
ステップ 26	[no] ip igmp snooping query-interval <i>seconds</i> 例 : <pre>apicl(config-tenant-interface) # ip igmp snooping query-interval 125</pre>	マルチキャストトラフィックをルーティングする必要がないため、PIM をイネーブルにしていない場合に、スヌーピングクエリーインターバルを設定します。指定できる範囲は 1 ～ 18000 秒です。デフォルト値は 125 秒です。
ステップ 27	[no] ip igmp snooping query-max-response-time <i>seconds</i> 例 : <pre>apicl(config-tenant-interface) # ip igmp snooping query-max-response-time 10</pre>	マルチキャストトラフィックをルーティングする必要がないため PIM を無効にしている場合に対して、クエリーメッセージのスヌーピング最大応答時間を設定します。値の範囲は 1 ～ 25 秒です。デフォルトは 10 秒です。
ステップ 28	[no] ip igmp snooping startup-query-count <i>count</i> 例 : <pre>apicl(config-tenant-interface) # ip igmp snooping startup-query-count 5</pre>	マルチキャストトラフィックをルーティングする必要がないため、PIM をイネーブルにしていない場合に、起動時に送信されるクエリー数に対してスヌーピングを設定します。範囲は 1 ～ 10 クエリーです。デフォルトは 5 クエリーです。
ステップ 29	[no] ip igmp snooping startup-query-interval <i>seconds</i> 例 : <pre>apicl(config-tenant-interface) # ip igmp snooping startup-query-interval 15000</pre>	マルチキャストトラフィックをルーティングする必要がないため、PIM をイネーブルにしていない場合に、起動時のスヌーピングクエリーインターバルを設定します。指定できる範囲は 1 ～ 18000 秒です。デフォルト値は 15000 秒です。
ステップ 30	[no] ip igmp startup-query-count <i>count</i> 例 : <pre>apicl(config-tenant-interface) # ip igmp startup-query-count 2</pre>	スタートアップクエリーインターバル中に送信される起動時のクエリー数を設定します。範囲は 1 ～ 10 クエリーです。デフォルトは 2 クエリーです。

	コマンドまたはアクション	目的
ステップ 31	[no] ip igmp startup-query-interval <i>seconds</i> 例 : <pre>apic1(config-tenant-interface)# ip igmp startup-query-interval 31</pre>	ソフトウェアの起動時に使用されるクエリーインターバルを設定します。デフォルトでは、ソフトウェアができるだけ迅速にグループステートを確立できるように、このインターバルはクエリーインターバルより短く設定されています。指定できる範囲は 1 ～ 18000 秒です。デフォルト値は 260 秒です。デフォルト値は 31 秒です。
ステップ 32	[no] ip igmp state-limit <i>max-states</i> [reserved <i>route-map-name</i> [max-reserved-gsg-entries]] 例 : <pre>apic1(config-tenant-interface)# ip igmp state-limit 100000 reserved myReservedPolicy 40000</pre>	IGMP メンバーシップレポート (IGMP 加入) の結果として作成される mroute ステートの数に対するインターフェイスごとの制限を設定します。許可される状態の範囲は 1 ～ 4294967295 です。オプションで、ポリシーマップに指定されるルート用に予約する状態エントリの数を指定することができ、また、インターフェイスで許可される最大予約済み (*, G) と (S, G) エントリを指定できます。予約済み状態の数は、許可される最大状態数以下である必要があります。範囲は 1 ～ 4294967295 です。
ステップ 33	[no] ip igmp static-oif route-map <i>route-map-name</i> 例 : <pre>apic1(config-tenant-interface)# ip igmp static-oif route-map MyOifMap</pre>	マルチキャスト グループを発信インターフェイス (OIF) に静的にバインドし、デバイスハードウェアで処理します。ルートマップは、この機能を適用するグループプレフィックスを定義します。
ステップ 34	[no] ip igmp version {v1 v2 v3} 例 : <pre>apic1(config-tenant-interface)# ip igmp version v3</pre>	インターフェイスの IGMP バージョン番号を設定します。デフォルトのバージョンは v2 です。
ステップ 35	exit 例 : <pre>apic1(config-tenant-interface)# exit</pre>	テナントコンフィギュレーションモードに戻ります。

次のタスク

テナントに L3 Out を設定し、PIM を有効にしてリーフ インターフェイスを設定します。

レイヤ 3 マルチキャストの L3 Out の設定

始める前に

- テナント VRF の PIM オプションを設定します。
- テナント VRF の IGMP を設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	コンフィギュレーションモードに入ります。
ステップ 2	tenant tenant-name 例 : apicl(config)# tenant exampleCorp	設定するテナントを指定します。
ステップ 3	l3out l3out-name 例 : apicl(config-tenant)# l3out exampleCorp_l3out	テナントに L3 Out のインターフェイスを設定します。
ステップ 4	ip pim 例 : apicl(config-tenant-l3out)# ip pim	インターフェイスで PIM をイネーブルにします。
ステップ 5	exit 例 : apicl(config-tenant-l3out)#	テナント コンフィギュレーションモードに戻ります。
ステップ 6	exit 例 : apicl(config-tenant)# exit	グローバル コンフィギュレーションモードに戻ります。
ステップ 7	leaf node-id 例 : apicl(config)# leaf 101	リーフ コンフィギュレーションモードを開始します。
ステップ 8	interface ethernet slot/port 例 : apicl(config-leaf)# interface ethernet 1/3	設定するインターフェイスを指定します。

	コマンドまたはアクション	目的
ステップ 9	[no] ip igmp allow-v3-asm 例 : <pre>apic1(config-leaf-if)# ip igmp allow-v3-asm</pre>	Any Source Multicast (ASM) グループに関する IGMPv3 レポートの送信元アドレスのフィルタリングを許可します。
ステップ 10	[no] ip igmp fast-leave 例 : <pre>apic1(config-leaf-if)# ip igmp fast-leave</pre>	IP IGMP スヌーピングの高速脱退処理を有効にします。この機能は、IGMPv2 プロトコルのホストレポート抑制メカニズムのために明示的に追跡できない IGMPv2 ホストをサポートします。高速脱退が有効になっている場合、IGMP ソフトウェアは、各ポートに接続されたホストが 1 つだけであると見なします。
ステップ 11	[no] ip igmp group-timeout <i>seconds</i> 例 : <pre>apic1(config-leaf-if)# ip igmp group-timeout 260</pre>	IGMPv2 のグループメンバーシップ タイムアウトを設定します。値の範囲は 3 ~ 65535 秒です。デフォルト値は 260 秒です。
ステップ 12	[no] ip igmp inherit interface-policy <i>policy-name</i> 例 : <pre>apic1(config-leaf-if)# ip igmp inherit interface-policy MyIfPolicy</pre>	このインターフェイスに IGMP インターフェイス ポリシーを関連付けます。
ステップ 13	[no] ip igmp join-group route-map <i>route-map-name</i> 例 : <pre>apic1(config-leaf-if)# ip igmp join-group route-map MyGroupsRMap</pre>	1 つ以上のマルチキャスト グループをインターフェイスに静的にバインドします。ルート マップ ポリシーは、グループのプレフィクス、グループの範囲、および送信元プレフィクスのリストを表示します。
ステップ 14	[no] ip igmp last-member-query-count <i>count</i> 例 : <pre>apic1(config-leaf-if)# ip igmp last-member-query-count 2</pre>	ホストの Leave メッセージを受信してから、IGMP クエリーが送信される回数を設定します。範囲は 1 ~ 5 クエリーです。デフォルトは 2 クエリーです。
ステップ 15	[no] ip igmp last-member-query-response-time <i>seconds</i> 例 : <pre>apic1(config-leaf-if)# ip igmp last-member-query-response-time 1</pre>	メンバーシップレポートを送信してから、ソフトウェアがグループステートを解除するまでのクエリーインターバルを設定します。値の範囲は 1 ~ 25 秒です。デフォルト値は 1 秒です。

	コマンドまたはアクション	目的
ステップ 16	[no] ip igmp querier-timeout <i>seconds</i> 例 : <pre>apicl(config-leaf-if)# ip igmp querier-timeout 255</pre>	前クエリアがクエリーを停止してから、自身がクエリアとして処理を引き継ぐまで、ソフトウェアが待機する秒数を設定します。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は 255 秒です。
ステップ 17	[no] ip igmp query-interval <i>seconds</i> 例 : <pre>apicl(config-leaf-if)# ip igmp query-interval 125</pre>	IGMP ホスト クエリー メッセージの送信頻度を設定します。大きな値を設定すると、ソフトウェアによる IGMP クエリーの送信頻度が低くなるため、ネットワーク上の IGMP メッセージ数を調整できます。指定できる範囲は 1 ～ 18000 秒です。デフォルト値は 125 秒です。
ステップ 18	[no] ip igmp query-max-response-time <i>seconds</i> 例 : <pre>apicl(config-leaf-if)# ip igmp query-max-response-time 10</pre>	IGMP クエリーでアドバタイズされる応答時間を設定します。大きな値を設定すると、ホストの応答時間が延長されるため、ネットワークの IGMP メッセージのバースト性を調整できます。この値は、クエリーインターバルよりも短く設定する必要があります。値の範囲は 1 ～ 25 秒です。デフォルトは 10 秒です。
ステップ 19	[no] ip igmp report-link-local-groups 例 : <pre>apicl(config-leaf-if)# ip igmp report-link-local-groups</pre>	224.0.0.0/24 に含まれるグループに対して、レポート送信をイネーブルにします。リンクローカルアドレスは、ローカルネットワークプロトコルだけで使用されます。非リンクローカルグループには、常にレポートが送信されます。デフォルトでは、リンクローカルグループにレポートは送信されません。
ステップ 20	[no] ip igmp report-policy <i>policy-name</i> 例 : <pre>apicl(config-leaf-if)# ip igmp report-policy MyReportPolicy</pre>	ルートマップポリシーに基づく、IGMP レポートのアクセスポリシーを設定します。
ステップ 21	[no] ip igmp robustness-variable <i>value</i> 例 : <pre>apicl(config-leaf-if)# ip igmp robustness-variable 2</pre>	輻輳ネットワーク上のパケット損失を補うようにロバストネス変数を設定します。ロバストネス値は、IGMP ソフトウェアがメッセージ送信回数を確認

	コマンドまたはアクション	目的
		するために使用されます。ネットワークのパケット損失が多い場合は、この値を大きくします。指定できる範囲は 1 ～ 7 です。デフォルトは 2 です。
ステップ 22	[no] ip igmp startup-query-count <i>count</i> 例 : <pre>apic1(config-leaf-if)# ip igmp startup-query-count 2</pre>	スタートアップクエリーインターバル中に送信される起動時のクエリー数を設定します。範囲は 1 ～ 10 クエリーです。デフォルトは 2 クエリーです。
ステップ 23	[no] ip igmp startup-query-interval <i>seconds</i> 例 : <pre>apic1(config-leaf-if)# ip igmp startup-query-interval 31</pre>	ソフトウェアの起動時に使用されるクエリーインターバルを設定します。デフォルトでは、ソフトウェアができるだけ迅速にグループステートを確立できるように、このインターバルはクエリーインターバルより短く設定されています。指定できる範囲は 1 ～ 18000 秒です。デフォルト値は 260 秒です。デフォルト値は 31 秒です。
ステップ 24	[no] ip igmp state-limit <i>max-states</i> [reserved <i>route-map-name</i> [max-reserved-gsg-entries]] 例 : <pre>apic1(config-leaf-if)# ip igmp state-limit 100000 reserved myReservedPolicy 40000</pre>	IGMP メンバーシップレポート (IGMP 加入) の結果として作成される mroute ステートの数に対するインターフェイスごとの制限を設定します。許可される状態の範囲は 1 ～ 4294967295 です。オプションで、ポリシーマップに指定されるルート用に予約する状態エントリの数を指定することができ、また、インターフェイスで許可される最大予約済み (*, G) と (S, G) エントリを指定できます。予約済み状態の数は、許可される最大状態数以下である必要があります。範囲は 1 ～ 4294967295 です。
ステップ 25	[no] ip igmp static-oif route-map <i>route-map-name</i> 例 : <pre>apic1(config-leaf-if)# ip igmp static-oif route-map MyOifMap</pre>	マルチキャスト グループを発信インターフェイス (OIF) に静的にバインドし、デバイスハードウェアで処理します。ルートマップは、この機能を適用するグループプレフィックスを定義します。
ステップ 26	[no] ip igmp version {v1 v2 v3} 例 : <pre>apic1(config-leaf-if)# ip igmp version v3</pre>	インターフェイスの IGMP バージョン番号を設定します。デフォルトのバージョンは v2 です。

	コマンドまたはアクション	目的
ステップ 27	exit 例： apicl(config-leaf-if)# exit	テナント コンフィギュレーション モードに戻ります。

例：レイヤ 3 マルチキャストの設定

```
# CONFIGURE PIM OPTIONS ON A TENANT VRF

apicl# configure
apicl(config)# tenant exampleCorp
apicl(config-tenant)# vrf context exampleCorp_vrf1
apicl(config-tenant-vrf)# ip pim
apicl(config-tenant-vrf)# ip pim fast-convergence
apicl(config-tenant-vrf)# ip pim bsr forward

# ENABLE AND CONFIGURE IGMP ON THE TENANT VRF AND BRIDGE DOMAIN

apicl(config-tenant-vrf)# ip igmp
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# interface bridge-domain exampleCorp_bd
apicl(config-tenant-interface)# ip multicast
apicl(config-tenant-interface)# ip igmp allow-v3-asm
apicl(config-tenant-interface)# ip igmp fast-leave
apicl(config-tenant-interface)# exit

# CREATE AN L3OUT AND CONFIGURE PIM

apicl(config-tenant)# l3out exampleCorp_l3out
apicl(config-tenant-l3out)# ip pim
apicl(config-tenant-l3out)# exit
apicl(config-tenant)# exit

# CONFIGURE AN EXTERNAL INTERFACE AND CONFIGURE IGMP ON THE INTERFACE

apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/125
apicl(config-leaf-if)# ip igmp fast-leave
apicl(config-leaf-if)# ip-igmp join-group
```

外部 L3 EPG の設定

外部 L3 EPG はテナント VRF 下に分類されます。CLI で、外部 L3 EPG はテナント モードで定義され、個々のノードに展開されます。VRF のすべてのノードではなく一部のノードに外部 L3 EPG を配置することができます。

外部 L3 EPG はそれぞれ複数のコントラクトのプロデューサまたはコンシューマになることができ、外部 L3 EPG にはそれぞれファブリック内の DSCP マーキングおよびキューイングプライオリティに関する独自の QoS ポリシーがあります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーションモードに入ります。
ステップ 2	tenant tenant-name 例 : apic1(config)# tenant exampleCorp	テナントコンフィギュレーションモードを開始します。
ステップ 3	external-l3 epg epg-name 例 : apic1(config-tenant)# external-l3 epg epgExtern1	外部 L3 EPG コンフィギュレーションモードを開始します。
ステップ 4	vrf member vrf-name 例 : apic1(config-tenant-l3ext-epg)# vrf member v1	EPG を VRF に関連付けます。
ステップ 5	match {ip ipv6} ip-address/masklength 例 : apic1(config-tenant-l3ext-epg)# match ip 192.0.20.0/24 apic1(config-tenant-l3ext-epg)# match ipv6 2001::1/64	サブネットに一致するルールを作成します。
ステップ 6	set qos-class class 例 : apic1(config-tenant-l3ext-epg)# set qos-class level1	EPG の QoS レベルを指定します。
ステップ 7	set dscp dscp-value 例 : apic1(config-tenant-l3ext-epg)# set dscp af31	EPG の DSCP 値を指定します。
ステップ 8	contract consumer contract-name 例 : apic1(config-tenant-l3ext-epg)# contract consumer cConsumer1	EPG のコンシューマコントラクトを指定します。
ステップ 9	contract provider contract-name 例 :	EPG のプロバイダーコントラクトを指定します。

	コマンドまたはアクション	目的
	<code>apicl(config-tenant-l3ext-epg) # contract provider cProvider1</code>	
ステップ 10	contract deny contract-name 例 : <code>apicl(config-tenant-l3ext-epg) # contract deny cDeny1</code>	EPG の拒否コントラクトを指定します。
ステップ 11	exit 例 : <code>apicl(config-tenant-l3ext-epg) # exit</code>	
ステップ 12	exit 例 : <code>apicl(config-tenant) # exit</code>	
ステップ 13	leaf node-id 例 : <code>apicl(config) # leaf 101</code>	設定するリーフを指定します。
ステップ 14	vrf context tenant tenant-name vrf vrf-name 例 : <code>apicl(config-leaf) # vrf context tenant exampleCorp vrf v1</code>	ノードのテナント VRF を設定します。
ステップ 15	external-l3 epg epg-name 例 : <code>apicl(config-leaf-vrf) # external-l3 epg epgExtern1</code>	VRF の外部レイヤ 3 EPG を関連付けます。

例

次に、外部レイヤ 3 EPG を設定し、リーフに EPG を展開する例を示します。

```
apicl# configure
apicl(config)# tenant exampleCorp

# CONFIGURE EXTERNAL L3 EPG
apicl(config-tenant) # external-l3 epg epgExtern1
apicl(config-tenant-l3ext-epg) # vrf member v1
apicl(config-tenant-l3ext-epg) # match ip 192.0.20.0/24
apicl(config-tenant-l3ext-epg) # match ipv6 2001::1/64
apicl(config-tenant-l3ext-epg) # set qos-class level1
apicl(config-tenant-l3ext-epg) # set dscp af31
apicl(config-tenant-l3ext-epg) # contract consumer cConsumer1
apicl(config-tenant-l3ext-epg) # contract provider cProvider1
apicl(config-tenant-l3ext-epg) # contract deny cDeny1
```

```

apic1(config-tenant-l3ext-epg) # exit
apic1(config-tenant) # exit

# DEPLOY EXTERNAL L3 EPG ON A LEAF
apic1(config) # leaf 101
apic1(config-leaf) # vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf) # external-l3 epg epgExtern1

```

名前付きモードを使用したレイヤ 3 外部接続の設定

名前付き L3Out の作成

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーションモードに入ります。
ステップ 2	tenant tenant-name 例 : apic1(config)# tenant exampleCorp	テナントコンフィギュレーションモードを開始します。
ステップ 3	vrf context vrf-name 例 : apic1(config-tenant)# vrf context v1	テナントを VRF に関連付けます。
ステップ 4	l3out l3out-name 例 : apic1(config-tenant)# l3out out1	名前付き L3Out を作成します。
ステップ 5	vrf member vrf-name 例 : apic1(config-tenant-l3out)# vrf member v1	L3Out をテナント VRF と関連付けます。
ステップ 6	exit 例 : apic1(config-tenant-l3out)# exit	テナントコンフィギュレーションモードに戻ります。
ステップ 7	exit 例 : apic1(config-tenant)# exit	グローバル コンフィギュレーションモードに戻ります。

	コマンドまたはアクション	目的
ステップ 8	leaf node-id 例 : apicl(config)# leaf 101	ノード
ステップ 9	vrf context tenant tenant-name vrf vrf-name l3out l3out-name 例 : apicl(config-leaf)# vrf context tenant exampleCorp vrf v1 l3out out1	ノードのテナント VRF を設定します。
ステップ 10	必須: [no] router-id ipv4-address 例 : apicl(config-leaf-vrf)# router-id 1.2.3.4	VRF で実行されているルーティングプロトコルに対してルータ ID を割り当てます。
ステップ 11	[no] {ip ipv6} route ip-prefix/masklen next-hop-address [preferred] 例 : apicl(config-leaf-vrf)# ip route 21.1.1.1/32 32.1.1.1 apicl(config-leaf-vrf)# ipv6 route 5001::1/128 6002::1	VRF の静的ルートの情報を設定します。

例

次に、テナントの下に名前付き L3Out を作成し、テナント VRF に割り当て、境界リーフスイッチに導入する例を示します。

```

apicl# configure
apicl(config)# tenant exampleCorp
apicl(config-tenant)# vrf context v1
apicl(config-tenant)# l3out out1
apicl(config-tenant-l3out)# vrf member v1
apicl(config-tenant-l3out)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf v1 l3out out1
apicl(config-leaf-vrf)# router-id 1.2.3.4
apicl(config-leaf-vrf)# ip route 21.1.1.1/32 32.1.1.1

```

次のタスク

名前付き L3Out のレイヤ 3 インターフェイスを設定します。

名前付き L3Out のレイヤ 3 インターフェイスの設定

この手順では、名前付き L3Out にレイヤ 3 ポート インターフェイスを設定する方法を示します。例では、名前付き L3Out にサブインターフェイスまたは SVI を設定する方法を示します。

- A given interface can be added to multiple L3Outs by providing multiple L3Out names after the **l3out** keyword.
- An SVI can be configured using the **switchport trunk allowed vlan** command under any of the following interface types:
 - **interface Ethernet**
 - **interface port-channel**
 - **interface vpc**

始める前に

名前付き L3Out を作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーション モードに入ります。
ステップ 2	leaf node-id 例 : <code>apic1(config)# leaf 101</code>	設定するリーフを指定します。
ステップ 3	interface type 例 : <code>apic1(config-leaf)# interface eth 1/20</code>	外部インターフェイスのポートを指定します。
ステップ 4	no switchport 例 : <code>apic1(config-leaf-if)# no switchport</code>	インターフェイスをレイヤ 3 インターフェイスとして設定し、設定オプションでレイヤ 3 コマンドを公開します。
ステップ 5	vrf member tenant tenant-name vrf vrf-name l3out l3out-name 例 : <code>apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1</code>	テナント VRF にインターフェイスを接続します。

	コマンドまたはアクション	目的
ステップ 6	[no] {ip ipv6} address <i>ip-prefix/masklen</i> [eui64] [secondary] [preferred] 例 : <pre>apic1(config-leaf-if)# ip address 10.1.1.1/24 apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred</pre>	インターフェイスに IP アドレスを設定します。指定されたアドレスは次のいずれかとして宣言できます。 • preferred—インターフェイスからトラフィックのデフォルト送信元アドレス。 • secondary—インターフェイスのセカンダリ アドレス。 With the optional eui64 keyword, the host can assign itself a 64-bit Extended Unique Identifier (EUI). In this mode, you can also configure ipv6 link-local, mac address, mtu, and other layer 3 properties on the interface.

例

次に、名前付き L3Out にレイヤ 3 ポートを割り当てる例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/20
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
```

次に、名前付き L3Out にレイヤ 3 サブインターフェイスを割り当てる例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/5
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vlan-domain member d1
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/5.1000
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
```

次に、名前付き L3Out にレイヤ 3 SVI を割り当てる例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 200
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1
```

```

apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/4
apic1(config-leaf-if)# vlan-domain member d1
apic1(config-leaf-if)# switchport trunk allowed vlan 200 tenant t1 external-svi l3out
out1

```

名前付き L3Out のルート マップの設定

- ルートマップはリーフ、VRF モードで設定します。
- 次のルートマップは、すべての名前付き L3Out に対して作成されます。
 - エクスポート：L3Out で有効なルーティングプロトコルからアドバタイズされるルートのルートマップ。By default, no routes are exported out until you explicitly enable them in the route-map through one or more of **match bridge-domain**, **match prefix-list** and **match community-list** statements.
 - インポート：L3Out のルーティングプロトコルにインポートされるルートのルートマップ。デフォルトでは、すべてのルートがインポートされます。You can control specific routes to be imported by using one or more **match prefix-list** or **match community-list** statements.
 - 共有：ルートと、この VRF からコントラクトの関連付けがあるその他の VRF へのルートをリークするために使用されるコントラクトプロバイダー/コンシューマ ポリシーを含むルートマップ。

These route-maps are created when you associate a leaf to the L3Out through the **vrf context tenant tenant-name vrf vrf-name l3out l3out-name** command.

- 名前付き L3Out 下のルートマップの範囲は常にグローバルであり、名前付き L3Out が展開されているすべてのノードに適用できます。
- All commands under the route-map (such as **match prefix-list**, **match community-list**, **match bridge-domain**) are the same as the route-map configuration for the Basic Mode discussed in the previous sections.

始める前に

名前付き L3Out を作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apic1# configure	コンフィギュレーションモードに入ります。

	コマンドまたはアクション	目的
ステップ 2	leaf node-id 例 : <pre>apic1(config)# leaf 101</pre>	設定するリーフを指定します。
ステップ 3	[no] vrf context tenant tenant-name vrf vrf-name l3out l3out-name 例 : <pre>apic1(config-leaf)# vrf context tenant exampleCorp vrf v1 l3out out1</pre>	ノードのテナント VRF を設定します。
ステップ 4	必須: [no] route-map name 例 : <pre>apic1(config-leaf-vrf)# route-map out1_in</pre>	ルートマップを作成し、ルートマップ コンフィギュレーションを開始します。これがインポートルートマップになります。
ステップ 5	必須: [no] ip prefix-list list-name permit prefix/masklen [le {32 128}] 例 : <pre>apic1(config-leaf-vrf-route-map)# ip prefix-list p1 permit 15.1.1.0/24</pre>	ルートマップにプレフィックスリストを作成します。
ステップ 6	必須: [no] match prefix-list list-name 例 : <pre>apic1(config-leaf-vrf-route-map)# match prefix-list p1</pre>	作成済みのプレフィックスリストと照合し、一致モードを開始して、プレフィックスリストのルート制御プロファイルを設定します。
ステップ 7	必須: exit 例 : <pre>apic1(config-leaf-vrf-route-map-match)# exit</pre>	ルートマップ コンフィギュレーション モードに戻ります。
ステップ 8	必須: exit 例 : <pre>apic1(config-leaf-vrf-route-map)# exit</pre>	リーフ VRF コンフィギュレーション モードに戻ります。
ステップ 9	必須: [no] route-map name 例 : <pre>apic1(config-leaf-vrf)# route-map out1_out</pre>	ルートマップを作成し、ルートマップ コンフィギュレーションを開始します。これがエクスポートルートマップになります。
ステップ 10	必須: [no] ip prefix-list list-name permit prefix/masklen [le {32 128}] 例 :	ルートマップにプレフィックスリストを作成します。

	コマンドまたはアクション	目的
	<code>apic1(config-leaf-vrf-route-map) # ip prefix-list p2 permit 16.1.1.0/24</code>	
ステップ 11	必須: [no] match prefix-list list-name 例 : <code>apic1(config-leaf-vrf-route-map) # match prefix-list p2</code>	作成済みのプレフィックスリストと照合し、一致モードを開始して、プレフィックスリストのルート制御プロファイルを設定します。
ステップ 12	必須: set tag name 例 : <code>apic1(config-leaf-vrf-route-map-match) # set tag 100</code>	タグ値を設定します。name パラメータは符号なし整数です。
ステップ 13	必須: exit 例 : <code>apic1(config-leaf-vrf-route-map-match) # exit</code>	ルートマップコンフィギュレーションモードに戻ります。
ステップ 14	必須: [no] match bridge-domain list-name 例 : <code>apic1(config-leaf-vrf-route-map) # match bridge-domain bdl</code>	プロトコルによってパブリック サブネットをエクスポートするため、ブリッジ ドメインと照合します。
ステップ 15	必須: exit 例 : <code>apic1(config-leaf-vrf-route-map-match) # exit</code>	ルートマップコンフィギュレーションモードに戻ります。
ステップ 16	必須: [no] route-map name 例 : <code>apic1(config-leaf-vrf) # route-map out1_shared</code>	ルートマップを作成し、ルートマップコンフィギュレーションを開始します。これが共有ルートマップになります。
ステップ 17	必須: [no] ip prefix-list list-name permit prefix/masklen [le {32 128}] 例 : <code>apic1(config-leaf-vrf-route-map) # ip prefix-list p3 permit 16.10.1.0/24</code>	ルートマップにプレフィックスリストを作成します。
ステップ 18	必須: [no] match prefix-list list-name 例 : <code>apic1(config-leaf-vrf-route-map) # match prefix-list p3</code>	作成済みのプレフィックスリストと照合し、一致モードを開始して、プレフィックスリストのルート制御プロファイルを設定します。

	コマンドまたはアクション	目的
ステップ 19	必須: contract provider name 例 : <pre>apicl(config-leaf-vrf-route-map-match)# contract provider default</pre>	この VRF から（このプレフィックスリストと一致する）ルートをリークするために必要なコントラクトを追加します。

例

次に、名前付き L3Out のルート マップを設定する例を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf v1 l3out out1

# CREATE IMPORT ROUTE-MAP
apicl(config-leaf-vrf)# route-map out1_in
apicl(config-leaf-vrf-route-map)# ip prefix-list p1 permit 15.1.1.0/24
apicl(config-leaf-vrf-route-map)# match prefix-list p1
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit

# CREATE EXPORT ROUTE-MAP
apicl(config-leaf-vrf)# route-map out1_out
apicl(config-leaf-vrf-route-map)# ip prefix-list p2 permit 16.1.1.0/24
apicl(config-leaf-vrf-route-map)# match prefix-list p2
apicl(config-leaf-vrf-route-map-match)# set tag 100
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# match bridge-domain bd1
apicl(config-leaf-vrf-route-map-match)# exit

# CREATE SHARED ROUTE-MAP
apicl(config-leaf-vrf)# route-map out1_shared
apicl(config-leaf-vrf-route-map)# ip prefix-list p3 permit 16.10.1.0/24
apicl(config-leaf-vrf-route-map)# match prefix-list p3
apicl(config-leaf-vrf-route-map-match)# contract provider default
```

名前付き L3Out のルーティング プロトコルの設定

名前付き L3Out の BGP の設定

- All commands under the BGP neighbor with the exception of **route-map** are identical to those in the Basic Mode of L3Out configuration. BGP テンプレート設定およびテンプレートの継承は、基本モードと同じです。
- L3Out 設定の名前付きモードでは、**route-map** は L3Out レベルで適用されます。L3Out とネイバーを関連付けることにより、**route-map** は L3Out のプロトコルに自動的に適用されます。したがって、**route-map** オプションは適用できず、BGP ネイバーでは使用できません。同じ理由で、**route-map** オプションは OSPF エリアで使用できず、**distribute-list EIGRP** オプションはインターフェイスで使用できません。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーション モードに入ります。
ステップ 2	leaf node-id 例 : <code>apic1(config)# leaf 101</code>	設定するリーフを指定します。
ステップ 3	router bgp asn-number 例 : <code>apic1(config-leaf)# router bgp 100</code>	BGP ポリシー設定を入力します。
ステップ 4	vrf member tenant tenant-name vrf vrf-name 例 : <code>apic1(config-bgp)# vrf member tenant exampleCorp vrf v100</code>	後続のポリシー設定モード コマンドと関連付ける VRF インスタンスを指定します。
ステップ 5	neighbor ip-address[/masklength] l3out l3out-name 例 : <code>apic1(config-leaf-bgp-vrf)# neighbor 192.0.2.229 l3out out1</code>	ネイバーの IP アドレスを指定します。
ステップ 6	remote-as asn 例 : <code>apic1(config-leaf-bgp-vrf-neighbor)# remote-as 300</code>	ネイバーの自律システム番号を指定します。
ステップ 7	allow-self-as-count count 例 : <code>apic1(config-leaf-bgp-vrf-neighbor)# allow-self-as-count 5</code>	指定できる数は 1 ～ 10 です。デフォルトは 3 です。
ステップ 8	update-source ethernet interface-range 例 : <code>apic1(config-leaf-bgp-vrf-neighbor)# update-source ethernet 1/3</code>	BGP パケットの送信元 IP を、ループバック、物理、サブインターフェイス、または SVI インターフェイスのいずれかに更新します。

例

次に、名前付き L3Out の BGP ルーティング プロトコルを設定する例を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant exampleCorp vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 192.0.2.229 l3out out1
apic1(config-leaf-bgp-vrf-neighbor)# remote-as 300
apic1(config-leaf-bgp-vrf-neighbor)# allow-self-as-count 5
apic1(config-leaf-bgp-vrf-neighbor)# update-source ethernet 1/3

```

名前付き L3Out の OSPF の設定

All commands under the **router ospf default** command, with the exception of **area area-id route-map map-name out**, are identical to those in the Basic Mode of L3Out configuration. インターフェイス下の OSPF コマンドおよび OSPF テンプレート継承コマンドも、基本モードと同じです。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーションモードに入ります。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	router ospf default 例 : apic1(config-leaf)# router ospf default	OSPF ルーティングプロセスを作成し、OSPF ポリシー設定を開始します。
ステップ 4	vrf member tenant tenant-name vrf vrf-name 例 : apic1(config-leaf-ospf)# vrf member tenant exampleCorp vrf v100	OSPF セッションで VRF を有効にします。
ステップ 5	area エリア id l3out l3out: 名前 例 : apic1(config-leaf-ospf-vrf)# area 0.0.0.1 l3out out1	L3Out で OSPF を有効にします。
ステップ 6	area area-id loopback loopback-address 例 : apic1(config-leaf-ospf-vrf)# area 0.0.0.1 loopback 192.0.20.11	OSPF が BGP の接続プロトコルとして使用される場合、OSPF は BGP セッションの送信元として使用されるループバックアドレスをアドバタイズしま

	コマンドまたはアクション	目的
		す。ループバック ID ではなくループバック IP アドレスが使用されることに注意してください。この場合、OSPF に依存している BGP セッションは <code>update-source</code> コマンドで同じループバック IP アドレスを使用します。
ステップ 7	area <i>area-id</i> nssa [no-redistribution] [default-information-originate] 例 : <pre>apic1(config-leaf-ospf-vrf)# area 0.0.0.1 nssa</pre>	Not-So-Stubby Area (NSSA) を定義します。
ステップ 8	exit 例 : <pre>apic1(config-leaf-ospf-vrf)# exit</pre>	OSPF コンフィギュレーションモードに戻ります。
ステップ 9	exit 例 : <pre>apic1(config-leaf-ospf)# exit</pre>	リーフ コンフィギュレーションモードに戻ります。
ステップ 10	interface <i>type</i> 例 : <pre>apic1(config-leaf)# interface eth 1/20</pre>	外部インターフェイスのポートを指定します。
ステップ 11	vlan-domain member <i>domain-name</i> 例 : <pre>apic1(config-leaf-if)# vlan-domain member dom1</pre>	インターフェイスに VLAN ドメインを割り当てます。The VLAN domain must have already been created using the vlan-domain command in the global configuration mode.
ステップ 12	no switchport 例 : <pre>apic1(config-leaf-if)# no switchport</pre>	インターフェイスをレイヤ 3 インターフェイスとして設定し、設定オプションでレイヤ 3 コマンドを公開します。
ステップ 13	vrf member tenant <i>tenant-name</i> vrf <i>vrf-name</i> l3out <i>l3out-name</i> 例 : <pre>apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1</pre>	テナント VRF にインターフェイスを接続します。
ステップ 14	[no] {ip ipv6} address <i>ip-prefix/masklen</i> [eui64] [secondary] [preferred] 例 :	インターフェイスに IP アドレスを設定します。指定されたアドレスは次のいずれかとして宣言できます。

	コマンドまたはアクション	目的
	<pre>apicl(config-leaf-if) # ip address 10.1.1.1/24 apicl(config-leaf-if) # ipv6 address 2001::1/64 preferred</pre>	• preferred—インターフェイスからトラフィックのデフォルト送信元アドレス。 • secondary—インターフェイスのセカンダリ アドレス。 With the optional eui64 keyword, the host can assign itself a 64-bit Extended Unique Identifier (EUI). In this mode, you can also configure ipv6 link-local, mac address, mtu, and other layer 3 properties on the interface.
ステップ 15	{ip ipv6} router ospf default area area-id 例 : <pre>apicl(config-leaf-if) # ip router ospf default area 0.0.0.1</pre>	OSPF ルーティングプロセスを作成し、OSPF ポリシー設定を開始します。

例

次に、名前付き L3Out の OSPF ルーティング プロトコルを設定する例を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant exampleCorp vrf v1
apicl(config-leaf-ospf-vrf)# area 0.0.0.1 l3out out1
apicl(config-leaf-ospf-vrf)# area 0.0.0.1 loopback 192.0.20.11
apicl(config-leaf-ospf-vrf)# area 0.0.0.1 nssa
apicl(config-leaf-ospf-vrf)# exit
apicl(config-leaf-ospf)# exit
apicl(config-leaf)# interface eth 1/20
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1
apicl(config-leaf-if)# ip address 10.1.1.1/24
apicl(config-leaf-if)# ip router ospf default area 0.0.0.1
```

名前付き L3Out の EIGRP の設定

All EIGRP commands under vrf mode and interface mode, with the exception of **ip distribute-list**, are identical to those in the Basic Mode of L3Out configuration. This includes the EIGRP **template** and **inherit** commands. The **ip distribute-list** commands are not applicable to the Named Mode of L3Out configuration, as the route-maps are defined at the L3Out level and by associating an interface with the L3Out, the route-map distribute-list is automatically associated. For this reason, **ip distribute-list** is not available in the CLI as an option.

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーションモードに入ります。
ステップ 2	leaf node-id 例 : apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	router eigrp default 例 : apic1(config-leaf)# router eigrp default	EIGRP ポリシー設定を入力します。
ステップ 4	vrf member tenant tenant-name vrf vrf-name 例 : apic1(config-eigrp)# vrf member tenant exampleCorp vrf v100	後続のコンフィギュレーションモードコマンドと関連付ける VRF インスタンスを指定します。
ステップ 5	autonomous-systemasn l3out l3out: 名前 例 : apic1(config-eigrp-vrf)# autonomous-system 500 l3out out1	EIGRP の自律システム設定を入力します。
ステップ 6	exit 例 : apic1(config-eigrp-vrf)# exit	EIGRP コンフィギュレーションモードに戻ります。
ステップ 7	exit 例 : apic1(config-eigrp)# exit	リーフ コンフィギュレーションモードに戻ります。
ステップ 8	interface type 例 : apic1(config-leaf)# interface eth 1/5	外部インターフェイスのポートを指定します。
ステップ 9	vlan-domain member domain-name 例 : apic1(config-leaf-if)# vlan-domain member dom1	インターフェイスに VLAN ドメインを割り当てます。The VLAN domain must have already been created using the vlan-domain command in the global configuration mode.

	コマンドまたはアクション	目的
ステップ 10	no switchport 例 : <pre>apic1(config-leaf-if)# no switchport</pre>	インターフェイスをレイヤ 3 インターフェイスとして設定し、設定オプションでレイヤ 3 コマンドを公開します。
ステップ 11	vrf member tenant tenant-name vrf vrf-name l3out l3out-name 例 : <pre>apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1</pre>	テナント VRF にインターフェイスを接続します。
ステップ 12	[no] {ip ipv6} address ip-prefix/masklen [eui64] [secondary] [preferred] 例 : <pre>apic1(config-leaf-if)# ip address 10.1.1.1/24 apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred</pre>	インターフェイスに IP アドレスを設定します。指定されたアドレスは次のいずれかとして宣言できます。 • preferred—インターフェイスからトラフィックのデフォルト送信元アドレス。 • secondary—インターフェイスのセカンダリ アドレス。 With the optional eui64 keyword, the host can assign itself a 64-bit Extended Unique Identifier (EUI). In this mode, you can also configure ipv6 link-local, mac address, mtu, and other layer 3 properties on the interface.
ステップ 13	{ip ipv6} router eigrp default 例 : <pre>apic1(config-leaf-if)# ip router eigrp default</pre>	EIGRP ポリシーをデフォルトに設定します。

例

次に、名前付き L3Out の EIGRP ルーティング プロトコルを設定する例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router eigrp default
apic1(config-eigrp)# vrf member tenant exampleCorp vrf v1
apic1(config-eigrp-vrf)# autonomous-system 500 l3out out1
apic1(config-eigrp-vrf)# exit
apic1(config-eigrp)# exit
apic1(config-leaf)# interface eth 1/5
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
```

```
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ip router eigrp default
```

名前付き L3Out の外部 L3 EPG の設定

外部 L3 EPG はテナント VRF 下に分類されます。

All commands under the **config-tenant-l3ext-epg** mode are identical to those in the Basic Mode of L3Out configuration with the following differences:

- VRF は自動的に EPG と関連付けられます。L3Out は VRF に関連付けられ、EPG は L3Out に関連付けられます。
- The **external-l3 epg** command is not available under the leaf **vrf context tenant tenant-name vrf vrf-name l3out l3out-name** command, as this configuration is not applicable for Named L3Outs. The **external-l3 epg** is automatically deployed on the leaf, when the **external-l3 epg** is created within a named L3Out and a leaf is associated with the same L3Out through the **vrf context tenant tenant-name vrf vrf-name l3out l3out-name** command.

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーション モードに入ります。
ステップ 2	tenant tenant-name 例 : apic1(config)# tenant exampleCorp	テナント コンフィギュレーション モードを開始します。
ステップ 3	external-l3 epg epg: l3out名前l3out 例 : apic1(config-tenant)# external-l3 epg epg1 l3out out1	外部 L3 EPG コンフィギュレーション モードを開始します。
ステップ 4	match {ip ipv6} ip-address/masklength 例 : apic1(config-tenant-l3ext-epg)# match ip 192.0.20.0/24 apic1(config-tenant-l3ext-epg)# match ipv6 2001::1/64	サブネットに一致するルールを作成します。
ステップ 5	contract consumer contract-name 例 :	EPG のコンシューマ コントラクトを指定します。

	コマンドまたはアクション	目的
	<code>apic1(config-tenant-l3ext-epg)# contract consumer cConsumer1</code>	
ステップ 6	contract provider contract-name 例 : <code>apic1(config-tenant-l3ext-epg)# contract provider cProvider1</code>	EPG のプロバイダー コントラクトを指定します。

例

次に、名前付き L3Out に外部レイヤ 3 EPG を設定する例を示します。

```
apic1# configure
apic1(config)# tenant exampleCorp
apic1(config-tenant)# external-l3 epg epg1 l3out out1
apic1(config-tenant-l3ext-epg)# match ip 192.0.20.0/24
apic1(config-tenant-l3ext-epg)# match ipv6 2001::1/64
apic1(config-tenant-l3ext-epg)# contract consumer cConsumer1
apic1(config-tenant-l3ext-epg)# contract provider cProvider1
```

IPv6 ネイバー探索

ネイバー探索

IPv6 ネイバー探索 (ND) は、ノードのアドレスの自動設定、リンク上の他のノードの探索、他のノードのリンク層アドレスの判別、重複アドレスの検出、使用可能なルータと DNS サーバの検出、アドレス プレフィックスの探索、および他のアクティブなネイバー ノードへのパスに関する到達可能性情報の維持を担当します。

ND 固有のネイバー要求/ネイバー アドバタイズメント (NS/NA) およびルータ要求/ルータ アドバタイズメント (RS/RA) パケット タイプは、物理、層 3 サブ インターフェイス、および SVI (外部およびパーベイス) を含むすべての ACI ファブリックのレイヤ 3 インターフェイスでサポートされます。APIC リリース 3.1(1x) まで、RS/RA パケットはすべてのレイヤ 3 インターフェイスの自動設定のために使用されますが、拡散型 SVI の設定のみ可能です。

APIC リリース 3.1(2x) より、RS/RA パケットは自動設定のため使用され、ルーテッドインターフェイス、レイヤ 3 サブ インターフェイス、SVI (外部および拡散) を含むレイヤ 3 インターフェイスで設定できます。

ACI のブリッジ ドメイン ND は常にフラッド モードで動作します。ユニキャスト モードはサポートされません。

ACI ファブリック ND サポートに含まれるもの :

- インターフェイス ポリシー (nd:IfPol) は、NS/NA メッセージに関する ND タイマーと動作を制御します。
- ND プレフィックス ポリシー (nd:PfxPol) コントロール RA メッセージ。
- ND の IPv6 サブネット (fv:Subnet) の設定。
- 外部ネットワークの ND インターフェイス ポリシー。
- 外部ネットワークの設定可能 ND サブネットおよびパーベイスブ ブリッジ ドメインの任意サブネット設定はサポートされません。

設定可能なオプションは次のとおりです。

- 隣接関係
 - 設定可能な静的 Adjacencies : (<vrf, L3Iface < ipv6 address> --> mac address)
 - 動的 Adjacencies : NS/NA パケットの交換経由で学習
- インターフェイス単位
 - ND パケットの制御 (NS/NA)
 - ネイバー要求間隔
 - ネイバー要求再試行回数
 - RA パケットの制御
 - RA の抑制
 - RA MTU の抑制
 - RA 間隔、RA 最小間隔、再送信時間
- プレフィックス単位 (RA でアドバタイズ) の制御
 - ライフタイム、優先ライフタイム
 - プレフィックス コントロール (自動設定、リンク上)
- ネイバー検索重複アドレスの検出 (DAD)

NX-OS スタイル CLI を使用したブリッジ ドメイン上の IPv6 ネイバー検索によるテナント、VRF、ブリッジ ドメインの設定

手順

ステップ 1 IPv6 ネイバー検索インターフェイス ポリシーを設定し、ブリッジ ドメインに割り当てます。

- a) IPv6 ネイバー検索インターフェイス ポリシーを作成します。

例 :

```
apic1(config)# tenant ExampleCorp
apic1(config-tenant)# template ipv6 nd policy NDPol001
apic1(config-tenant-template-ipv6-nd)# ipv6 nd mtu 1500
```

- b) VRF およびブリッジ ドメインを作成します:

例 :

```
apic1(config-tenant)# vrf context pvn1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member pvn1
apic1(config-tenant-bd)# exit
```

- c) IPv6 ネイバー検索ポリシーをブリッジ ドメインに割り当てます。

例 :

```
apic1(config-tenant)# interface bridge-domain bd1
apic1(config-tenant-interface)# ipv6 nd policy NDPol001
apic1(config-tenant-interface)#exit
```

ステップ 2 サブネット上で IPV6 ブリッジ ドメイン サブネットおよびネイバー検索プレフィックス ポリシーを作成します。

例 :

```
apic1(config-tenant)# interface bridge-domain bd1
apic1(config-tenant-interface)# ipv6 address 34::1/64
apic1(config-tenant-interface)# ipv6 address 33::1/64
apic1(config-tenant-interface)# ipv6 nd prefix 34::1/64 1000 1000
apic1(config-tenant-interface)# ipv6 nd prefix 33::1/64 4294967295 4294967295
```

注意事項と制約事項

次のガイドラインと制限事項に適用ネイバー探索ルータ アドバタイズメント (ND RA) のプレフィックスのレイヤ 3 インターフェイス。

- NDRA 設定は、IPv6 プレフィックスにのみ適用されます。IPv4 プレフィックスで ND ポリシーを設定しようとするは適用に失敗します。

NX-OS スタイル CLI を使用したレイヤ 3 インターフェイス上の RA による IPv6 ネイバー探索インターフェイス ポリシーの設定

この例では、IPv6 ネイバー探索インターフェイス ポリシーを設定し、レイヤ 3 インターフェイスに割り当てます。次に、IPv6 レイヤ 3 アウトインターフェイス、ネイバー探索プレフィックス ポリシーを設定し、インターフェイスにネイバー探索ポリシーを関連付けます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーションモードに入ります。
ステップ 2	tenant <i>tenant_name</i> 例 : <code>apic1(config)# tenant ExampleCorp</code> <code>apic1(config-tenant)#</code>	テナントを作成し、テナントモードを開始します。
ステップ 3	template ipv6 nd policy <i>policy_name</i> 例 : <code>apic1(config-tenant)# template ipv6 nd policy NDPol001</code>	IPv6 ND ポリシーを作成します。
ステップ 4	ipv6 nd mtu <i>mtu value</i> 例 : <code>apic1(config-tenant-template-ipv6-nd)# ipv6 nd mtu 1500</code> <code>apic1(config-tenant-template-ipv6)# exit</code> <code>apic1(config-tenant-template)# exit</code> <code>apic1(config-tenant)#</code>	IPv6 ND ポリシーに MTU 値を割り当てます。
ステップ 5	vrf context <i>VRF_name</i> 例 : <code>apic1(config-tenant)# vrf context pvn1</code>	VRF を作成します。

	コマンドまたはアクション	目的
	<code>apicl(config-tenant-vrf) # exit</code>	
ステップ 6	<code>l3out VRF_name</code> 例 : <pre> apicl(config-tenant) # l3out l3extOut001 </pre>	レイヤ 3 アウトを作成します。
ステップ 7	<code>vrf member VRF_name</code> 例 : <pre> apicl(config-tenant-l3out) # vrf member pvn1 apicl(config-tenant-l3out) # exit </pre>	VRF をレイヤ 3 アウトインターフェイスに関連付けます。
ステップ 8	<code>external-l3 epg instp l3out l3extOut001</code> 例 : <pre> apicl(config-tenant) # external-l3 epg instp l3out l3extOut001 apicl(config-tenant-l3ext-epg) # vrf member pvn1 apicl(config-tenant-l3ext-epg) # exit </pre>	レイヤ 3 アウトおよび VRF をレイヤ 3 インターフェイスに割り当てます。
ステップ 9	<code>leaf 2011</code> 例 : <pre> apicl(config) # leaf 2011 </pre>	リーフスイッチモードを開始します。
ステップ 10	<code>vrf context tenant ExampleCorp vrf pvn1 l3out l3extOut001</code> 例 : <pre> apicl(config-leaf) # vrf context tenant ExampleCorp vrf pvn1 l3out l3extOut001 apicl(config-leaf-vrf) # exit </pre>	VRF をリーフスイッチに関連付けます。
ステップ 11	<code>int eth 1/1</code> 例 : <pre> apicl(config-leaf) # int eth 1/1 apicl(config-leaf-if) # </pre>	インターフェイスモードに入ります。

	コマンドまたはアクション	目的
ステップ 12	vrf member tenant ExampleCorp vrf pvn1 l3out l3extOut001 例 : <pre>apic1(config-leaf-if)# vrf member tenant ExampleCorp vrf pvn1 l3out l3extOut001</pre>	インターフェイスで関連付けられているテナント、VRF、レイヤ 3 Out を指定します。
ステップ 13	ipv6 address 2001:20:21:22::2/64 preferred 例 : <pre>apic1(config-leaf-if)# ipv6 address 2001:20:21:22::2/64 preferred</pre>	プライマリまたは優先 Ipv6 アドレスを指定します。
ステップ 14	ipv6 nd prefix 2001:20:21:22::2/64 1000 1000 例 : <pre>apic1(config-leaf-if)# ipv6 nd prefix 2001:20:21:22::2/64 1000 1000</pre>	レイヤ 3 インターフェイス下で IPv6 ND プレフィックス ポリシーを設定します。
ステップ 15	inherit ipv6 nd NDPol001 例 : <pre>apic1(config-leaf-if)# inherit ipv6 nd NDPol001 apic1(config-leaf-if)# exit apic1(config-leaf)# exit</pre>	レイヤ 3 インターフェイス下で ND ポリシーを設定します。

設定が完了します。

『Configuring HSRP』

NX-OS スタイル CLI での Cisco APIC を使用してインラインパラメータで HSRP の設定

リーフ スイッチが設定されている場合、HSRP が有効になっています。

始める前に

- テナントと VRF が設定されています。
- VLAN プールは、適切な VLAN 範囲が定義され、レイヤ 3 ドメインが作成されて VLAN プールに接続されている状態で設定される必要があります。
- エンティティプロファイルの接続も、レイヤ 3 ドメインに関連付けられている必要があります。
- リーフスイッチのインターフェイスプロファイルは必要に応じて設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>apic1# configure</pre>	コンフィギュレーション モードに入ります。
ステップ 2	インラインパラメータを作成することにより、HSRP を設定します。 例 : <pre>apic1(config)# leaf 101 apic1(config-leaf)# interface ethernet 1/17 apic1(config-leaf-if)# hsrp version 1 apic1(config-leaf-if)# hsrp use-bia apic1(config-leaf-if)# hsrp delay minimum 30 apic1(config-leaf-if)# hsrp delay reload 30 apic1(config-leaf-if)# hsrp 10 ipv4 apic1(config-if-hsrp)# ip 182.16.1.2 apic1(config-if-hsrp)# ip 182.16.1.3 secondary apic1(config-if-hsrp)# ip 182.16.1.4 secondary apic1(config-if-hsrp)# mac-address 5000.1000.1060 apic1(config-if-hsrp)# timers 5 18 apic1(config-if-hsrp)# priority 100 apic1(config-if-hsrp)# preempt apic1(config-if-hsrp)# preempt delay minimum 60 apic1(config-if-hsrp)# preempt delay reload 60 apic1(config-if-hsrp)# preempt delay sync 60 apic1(config-if-hsrp)# authentication none apic1(config-if-hsrp)# authentication simple apic1(config-if-hsrp)# authentication</pre>	

	コマンドまたはアクション	目的
	<pre>md5 apicl(config-if-hsrp)# authentication-key <mypassword> apicl(config-if-hsrp)# authentication-key-timeout <timeout></pre>	

NX-OS スタイル CLI のテンプレートとポリシーを使用した Cisco APIC の HSRP の設定

リーフ スイッチが設定されている場合、HSRP が有効になっています。

始める前に

- テナントと VRF が設定されています。
- VLAN プールは、適切な VLAN 範囲が定義され、レイヤ 3 ドメインが作成されて VLAN プールに接続されている状態で設定される必要があります。
- エンティティプロファイルの接続も、レイヤ 3 ドメインに関連付けられている必要があります。
- リーフ スイッチのインターフェイス プロファイルは必要に応じて設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	<pre>configure</pre> 例 : <pre>apicl# configure</pre>	コンフィギュレーション モードに入ります。
ステップ 2	HSRP ポリシーテンプレートを設定します。 例 : <pre>apicl(config)# leaf 101 apicl(config-leaf)# template hsrp interface-policy hsrp-intfPol1 tenant t9 apicl(config-template-hsrp-if-pol)# hsrp use-bia apicl(config-template-hsrp-if-pol)# hsrp delay minimum 30 apicl(config-template-hsrp-if-pol)# hsrp delay reload 30</pre>	

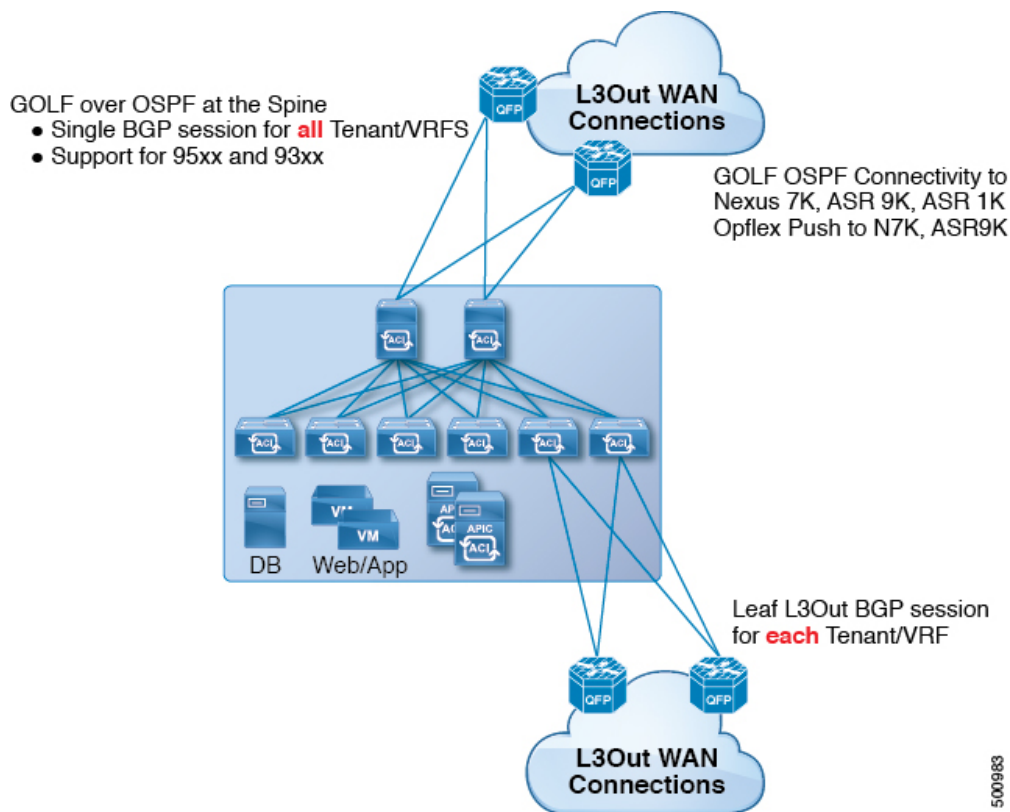
	コマンドまたはアクション	目的
	<pre> apic1(config)# leaf 101 apic1(config-leaf)# template hsrp group-policy hsrp-groupPol1 tenant t9 apic1(config-template-hsrp-group-pol)# timers 5 18 apic1(config-template-hsrp-group-pol)# priority 100 apic1(config-template-hsrp-group-pol)# preempt apic1(config-template-hsrp-group-pol)# preempt delay minimum 60 apic1(config-template-hsrp-group-pol)# preempt delay reload 60 apic1(config-template-hsrp-group-pol)# preempt delay sync 60 </pre>	
ステップ 3	設定されているポリシー テンプレートを使用します。 例 : <pre> apic1(config)# leaf 101 apic1(config-leaf)# interface ethernet 1/17 apic1(config-leaf-if)# hsrp version 1 apic1(config-leaf-if)# inherit hsrp interface-policy hsrp-intfPol1 apic1(config-leaf-if)# hsrp 10 ipv4 apic1(config-if-hsrp)# ip 182.16.1.2 apic1(config-if-hsrp)# ip 182.16.1.3 secondary apic1(config-if-hsrp)# ip 182.16.1.4 secondary apic1(config-if-hsrp)# mac-address 5000.1000.1060 apic1(config-if-hsrp)# inherit hsrp group-policy hsrp-groupPol1 </pre>	

Cisco ACI GOLF

Cisco ACI GOLF

Cisco ACI GOLF 機能 (ファブリック WAN のレイヤ 3 EVPN サービス機能とも呼ばれる) では、より効率的かつスケーラブルな ACI ファブリック WAN 接続が可能になります。スパインスイッチに接続されている WAN に OSPF 経由で BGP EVPN プロトコルが使用されます。

図 23 : Cisco ACI GOLF のトポロジ



すべてのテナント WAN 接続が、WAN ルータが接続されたスパイン スイッチ上で単一のセッションを使用します。データセンター相互接続ゲートウェイ (DCIG) へのテナント BGP セッションのこの集約では、テナント BGP セッションの数と、それらすべてに必要な設定の量を低減することによって、コントロールプレーンのスケールが向上します。ネットワークは、スパイン ファブリック ポートに設定された レイヤ 3 サブインターフェイスを使用して拡張されます。GOLFを使用した、共有サービスを伴うトランジットルーティングはサポートされていません。

スパイン スイッチでの GOLF 物理接続のためのレイヤ 3 外部外側ネットワーク (L3extOut) は、*infra* テナントの下で指定され、次のものを含みます:

- LNodeP (*infra* テナントの L3Out では、L3extInstP は必要ありません)。
- *infra* テナントの GOLF 用の L3extOut のプロバイダ ラベル。
- OSPF プロトコル ポリシー
- BGP プロトコル ポリシー

すべての通常テナントが、上記で定義した物理接続を使用します。通常のテナントで定義した L3extOut では、次が必要です:

- サブネットとコントラクトを持つ `L3extInstP` (EPG)。サブネットの範囲を使用して、ルート制御ポリシーとセキュリティポリシーのインポートまたはエクスポートを制御します。ブリッジドメインサブネットは外部的にアドバタイズするように設定される必要があります。アプリケーション EPG および GOLF L3Out EPG と同じ VRF に存在する必要があります。
- アプリケーション EPG と GOLF L3Out EPG の間の通信は、(契約優先グループではなく) 明示的な契約によって制御されます。
- `L3extConsLbl` コンシューマ ラベル。これは `infra` テナントの GOLF 用の L3Out の同じプロバイダラベルと一致している必要があります。ラベルを一致させることにより、他のテナント内のアプリケーション EPG が LNodeP 外部 L3Out EPG を利用することが可能になります。
- `infra` テナント内のマッチング プロバイダ L3extOut の BGP EVPN セッションは、この L3Out で定義されたテナント ルートをアドバタイズします。

次に示す GOLF のガイドラインおよび制限事項に従ってください。

- すべての Cisco Nexus 9000 シリーズ ACI モードのスイッチと、すべての Cisco Nexus 9500 プラットフォーム ACI モード スイッチライン カードおよびファブリック モジュールが GOLF をサポートします。Cisco APIC、リリース 3.1(x) 以降では、これに N9K-C9364C スイッチが含まれます。
- 現時点では、ファブリック全体のスパインスイッチインターフェイスに展開できるのは、単一の GOLF プロバイダ ポリシーだけです。
- APIC リリース 2.0(2) までは、GOLF はマルチポッドでサポートされていません。リリース 2.0 (2) では、同じファブリックでの 2 つの機能を、スイッチ名の末尾に「EX」のない Cisco Nexus N9000K スイッチ上でのみサポートしています。たとえば N9K-9312TX です。2.1(1) リリース以降では、2 つの機能を、マルチポッドおよび EVPN トポロジで使用されているすべてのスイッチとともに展開できるようになりました。
- スパイン スイッチで GOLF を設定する場合、コントロールプレーンがコンバージするまでは、別のスパイン スイッチで GOLF の設定を行わないでください。
- スパイン スイッチは複数のプロバイダの GOLF 外側ネットワーク (GOLF L3Outs) に追加できますが、GOLF L3Out ごとのプロバイダ ラベルは異なっている必要があります。また、この例では、OSPF エリアも L3extOut ごとに異なっていて、異なるルーブバックアドレスを使用する必要があります。
- `infra` テナント内のマッチング プロバイダ L3Out の BGPEVPN セッションは、この L3extOut で定義されたテナント ルートをアドバタイズします。
- 3 つの GOLF Outs を展開する場合、1 つだけが GOLF, and 0/0 エクスポート集約のプロバイダ/コンシューマラベルを持っているなら、APIC はすべてのルートをエクスポートします。これは、テナントのリーフ スイッチ上の既存の L3extOut と同じです。
- スパインスイッチとデータセンター相互接続 (DCI) ルータ間に直接ピアリングがある場合、リーフ スイッチから ASR へのトランジットルートには、リーフ スイッチの PTEP として次のホップが存在することになります。この場合、その ACI ポッドの TEP 範囲に対

して ASR の静的ルートを定義します。また、DCI が同じポッドにデュアルホーム接続されている場合は、静的ルートの優先順位（管理距離）は、他のリンクを通じて受信するルートと同じである必要があります。

- デフォルトの `bgpPeerPfxPol` ポリシーは、ルートを 20,000 に制限します。ACI WAN インターコネクト ピアの場合には、必要に応じてこれを増やしてください。
- 1 つのスパイン スイッチ上に 2 つの `L3extOut` が存在し、そのうちの一方のプロバイダ ラベルが `prov1` で DCI 1 とピアリングしており、もう一方の `L3extOut` のプロバイダ ラベルが `prov2` で DCI 2 とピアリングしているという、展開シナリオを考えます。テナント VRF に、プロバイダ ラベルのいずれか一方 (`prov1` または `prov2`) をポイントしているコンシューマ ラベルがある場合、テナントルートは DCI 1 と DCI 2 の両方に送信されます。
- GOLF OpFlex Vrf を集約する場合、ACI ファブリックまたは GOLF OpFlex VRF とシステム内のその他の VRF 間の GOLF デバイスでは、ルートのリーキングは発生しません。VRF リーキングのためには、(GOLF ルータではなく) 外部デバイスを使用する必要があります。



(注) Cisco ACI は、IP フラグメンテーションをサポートしていません。したがって、外部ルータへのレイヤ 3 Outside (L3Out) 接続、または Inter-Pod Network (IPN) を介した multipod 接続を設定する場合は、MTU が両側で適切に設定されていることが重要です。ACI、Cisco NX-OS、Cisco IOS などの一部のプラットフォームでは、設定された MTU 値は IP ヘッダーを考慮に入れています（結果として、最大パケットサイズは、ACI で 9216 バイト、NX-OS および IOS で 9000 バイトに設定されます）。ただし、IOS XR などの他のプラットフォームは、パケットヘッダーのを除く MTU 値を設定します（結果として最大パケット サイズは 8986 バイトになります）。

各プラットフォームの適切な MTU 値については、それぞれの設定ガイドを参照してください。

CLI ベースのコマンドを使用して MTU をテストすることを強く推奨します。たとえば、Cisco NX-OS CLI で `ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1` などのコマンドを使用します。

NX-OS スタイル CLI を使用して Cisco ACI GOLF サービスを設定する設定タスク

次のタスクを実行して、NX-OS スタイル CLI で GOLF サービスを設定します（BGP EVPN プロトコルを使用）。

- VLAN ドメイン、VRF、インターフェイス IP アドレッシング、OSPF を含む BGP EVPN の内部テナントを設定します。
- スパインの BPG を設定して BGP EVPN をサポートします。
- BGP EVPN のテナントを設定します。

- テナントの BGP EVPN ルート ターゲット、ルート マップ、プレフィックス EPG を設定します。
- BGP アドレスファミリを設定して、**host-rt-enable** コマンドで DCIG に BGP EVPN type-2 (MAC-IP) ホスト ルートを配布することを有効にします。

スパインを設定し、NX-OS スタイル CLI を使用して、BGP EVPN のテナント インフラ

このタスクを設定する方法を説明する、次の手順で VLAN ドメイン、VRF、インターフェイスの IP アドレッシングを含む、BGP EVPN および OSPF のテナント インフラ。

手順

	コマンドまたはアクション	目的
ステップ 1	configure	グローバル コンフィギュレーション モードを開始します。 apic1# configure
ステップ 2	vlan-domain <i>vlan-domain-name</i> dynamic	Creates a VLAN domain. apic1(config)# vlan-domain evpn-dom dynamic
ステップ 3	spine <i>spine-name</i>	Creates the spine or enters spine configuration mode. apic1(config)# spine 111
ステップ 4	vrf context <i>tenant</i> <i>tenant-name</i> vrf <i>vrf-name</i>	テナントに、VRF を関連付けます。 apic1(config-spine)# vrf context tenant infra vrf overlay-1
ステップ 5	router-id <i>A.B.C.D</i>	VRF のルータ ID を設定します。 apic1(config-spine-vrf)# router-id 10.10.3.3
ステップ 6	exit	Returns to spine configuration mode. apic1(config-spine-vrf)# exit
ステップ 7	interface ethernet <i>slot/port</i>	スパイン ノードのインターフェイスを設定します。 apic1(config-spine)# interface ethernet 1/33
ステップ 8	vlan-domain member <i>vlan-domain-name</i>	Associates the interface with the VLAN domain. apic1(config-spine-if)# vlan-domain member evpn-dom

スパインを設定し、NX-OS スタイル CLI を使用して、BGP EVPN のテナントインフラ

	コマンドまたはアクション	目的
ステップ 9	exit	Returns to spine configuration mode. <code>apic1(config-spine-if) # exit</code>
ステップ 10	interface ethernet <i>sub-interface-id</i>	Creates a sub-interface. <code>apic1(config-spine) # interface ethernet 1/33.4</code>
ステップ 11	vrf member tenant <i>tenant-name</i> vrf <i>vrf-name</i>	インターフェイスをオーバーレイ 1 VRF に関連付けると、テナントインフラ。 <code>apic1(config-spine-if) # vrf member tenant infra vrf overlay-1</code>
ステップ 12	mtu <i>mtu-value</i>	Sets the maximum transmission unit (MTU) for the interface. <code>apic1(config-spine-if) # mtu 1500</code>
ステップ 13	ip address <i>A.B.C.D/LEN</i>	インターフェイスの IP アドレスを設定します。 <code>apic1(config-spine-if) # ip address 5.0.0.1/24</code>
ステップ 14	ip router ospf default area <i>ospf エリア id</i>	インターフェイスのデフォルトの OSPF エリア ID を設定します。 <code>apic1(config-spine-if) # ip router ospf default area 0.0.0.150</code>
ステップ 15	exit	Returns to spine configuration mode. <code>apic1(config-spine-if) # exit</code>
ステップ 16	interface ethernet <i>slot/port</i>	スパイン ノードのインターフェイスを設定します。 <code>apic1(config-spine) # interface ethernet 1/34</code>
ステップ 17	vlan-domain member <i>vlan-domain-name</i>	Associates the interface with the VLAN domain. <code>apic1(config-spine-if) # vlan-domain member evpn-dom</code>
ステップ 18	exit	Returns to spine configuration mode. <code>apic1(config-spine-if) # exit</code>
ステップ 19	interface ethernet <i>sub-interface-id</i>	Creates a sub-interface. <code>apic1(config-spine) # interface ethernet 1/34.4</code>
ステップ 20	vrf member tenant <i>tenant-name</i> vrf <i>vrf-name</i>	インターフェイスをオーバーレイ 1 VRF に関連付けると、テナントインフラ。

	コマンドまたはアクション	目的
		<code>apic1(config-spine-if) # vrf member tenant infra vrf overlay-1</code>
ステップ 21	<code>mtu mtu-value</code>	Sets the maximum transmission unit (MTU) for the interface. <code>apic1(config-spine-if) # mtu 1500</code>
ステップ 22	<code>ip address A.B.C.D/LEN</code>	インターフェイスの IP アドレスを設定します。 <code>apic1(config-spine-if) # ip address 2.0.0.1/24</code>
ステップ 23	<code>ip router ospf default area ospf エリア id</code>	インターフェイスのデフォルトの OSPF エリア ID を設定します。 <code>apic1(config-spine-if) # ip router ospf default area 0.0.0.200</code>
ステップ 24	<code>exit</code>	Returns to spine configuration mode. <code>apic1(config-spine-if) # exit</code>
ステップ 25	<code>router ospf default</code>	スパインの OSPF を設定します。 <code>apic1(config-spine) # router ospf default</code>
ステップ 26	<code>vrf member tenant tenant-name vrf vrf-name</code>	テナントをオーバーレイ 1 の VRF とインフラ Router OSPF ポリシーに関連付けます。 <code>apic1(config-spine-ospf) # vrf member tenant infra vrf overlay-1</code>
ステップ 27	<code>area area-id loopback loopback-ip-address</code>	Configure an OSPF area for the OSPF policy. <code>apic1(config-spine-ospf-vrf) # area 0.0.0.150 loopback 10.10.5.3</code>
ステップ 28	<code>area area-id loopback loopback-ip-address</code>	OSPF ポリシーのもう 1 つの OSPF エリアを設定します。 <code>apic1(config-spine-ospf-vrf) # area 0.0.0.200 loopback 10.10.4.3</code>
ステップ 29	<code>exit</code>	Returns to spine OSPF configuration mode. <code>apic1(config-spine-ospf-vrf) # exit</code>
ステップ 30	<code>exit</code>	Returns to spine configuration mode. <code>apic1(config-spine-ospf) # exit</code>

複数のサイトで共有 APIC ゴルフ接続

トポロジでは、複数のサイト、APIC サイトの拡大 Vrf は、ゴルフ接続を共有している場合、リスクのクロス VRF トラフィックの問題を回避する次のガイドラインに従います。

スパインスイッチと、DCI の間でルート ターゲットの設定

ゴルフ Vrf の EVPN ルート ターゲット (RTs) を設定する 2 つの方法があります: 手動 RT と自動 RT. ルート ターゲットは、ACI 背表紙と OpFlex を介して DCIs の間で同期されます。ゴルフ Vrf の自動 RT は、形式に組み込まれて Fabric ID: - ASN : [FabricID] VNID

2 つのサイトには、次の図のように導入の Vrf がある、Vrf 間のトラフィックを混在させることができます。

サイト 1	サイト 2
ASN: 100、ファブリック ID: 1	ASN: 100、ファブリック ID: 1
VRF A : VNID 1000 インポート/エクスポートルートターゲット : 100 : [1] 1000	VRF A : VNID 2000 インポート/エクスポートルートターゲット : 100 : [1] 2000
VRF B : VNID 2000 インポート/エクスポートルートターゲット : 100 : [1] 2000	VRF B : VNID 1000 インポート/エクスポートルートターゲット : 100 : [1] 1000

Dci のために必要なルート マップ

トンネルは、中継ルートは、[DCI を介してリークとサイト間では作成されません、ため、コントロール プレーンの手間をも削減する必要があります。もう 1 つのサイトでゴルフ スパインに、DCI への 1 つのサイトでゴルフ スパインから送信される EVPN タイプ 5 およびタイプ 2 ルートを送信できませんする必要があります。これが発生スパイン スイッチに dci のために次のタイプの BGP セッションが必要がある場合。

Site1: IBGP--DCI--EBGP--サイト 2

Site1: EBGP--DCI--IBGP--サイト 2

Site1:--DCI--EBGP EBGP--サイト 2

Site1: IBGP RR クライアント--DCI (RR)---IBGP サイト 2

Dci のためにこの問題を避けるためには、ルートマップは、インバウンドおよびアウトバウンドのピア ポリシーのさまざまな BGP コミュニティで使用されます。

ルートを 1 つのサイト、もう 1 つのサイト フィルタ着信ピア ポリシーでコミュニティに基づくルートでゴルフ スパインへのアウトバウンドピア ポリシー ゴルフ スパインから受信します。別のアウトバウンドピア ポリシーは、WAN へコミュニティを取り除き。すべてのルートマップは、ピアのレベルです。

NX-OS スタイル CLI を使用した推奨される共有 GOLF 設定

マルチサイトで管理されている複数の APIC サイト間で、DCI による GOLF 接続を共有する場合、ルートマップと BPG を設定し VRF 間のトラフィックの問題を避けるために次の手順を使用します。

手順

ステップ 1 インバウンドルート マップ

例 :

```
Inbound peer policy to attach community:
|
| route-map multi-site-in permit 10
|
| set community 1:1 additive
```

ステップ 2 アウトバウンドピア ポリシーを設定し、インバウンドピア ポリシーのコミュニティに基づいてルートをフィルタします。

例 :

```
ip community-list standard test-com permit 1:1

route-map multi-site-out deny 10

    match community test-com exact-match

route-map multi-site-out permit 11
```

ステップ 3 アウトバウンドピア ポリシーを設定し、WAN へのコミュニティをフィルタします。

例 :

```
ip community-list standard test-com permit 1:1

route-map multi-site-wan-out permit 11

    set comm-list test-com delete
```

ステップ 4 BGP を設定します。

例 :

```
router bgp 1

    address-family l2vpn evpn

    neighbor 11.11.11.11 remote-as 1

    update-source loopback0

    address-family l2vpn evpn

        send-community both

    route-map multi-site-in in

    neighbor 13.0.0.2 remote-as 2
```

```

address-family l2vpn evpn

send-community both

route-map multi-site-out out

```

NX-OS スタイル CLI を使用して、スパインで BGP EVPN をサポートする BGP の設定

このタスクは、次の手順で BGP EVPN をサポートするスパインで BGP を設定する方法を示しています。

手順

	コマンドまたはアクション	目的
ステップ 1	configure	グローバル コンフィギュレーション モードを開始します。 apic1# configure
ステップ 2	spine <i>spine-name</i>	Creates the spine or enters spine configuration mode. apic1(config)# spine 111
ステップ 3	router bgp <i>AS-number</i>	スパイン ノードの BGP を設定します。 apic1(config-spine)# router bgp 100
ステップ 4	vrf context tenant <i>tenant-name vrf vrf-name</i>	Router BGP ポリシーに関連付けます、インフラ テナントとオーバーレイ 1 VRF。 apic1(config-spine-bgp)# vrf context tenant infra vrf overlay-1
ステップ 5	vrf context tenant <i>tenant-name vrf vrf-name</i>	Router BGP ポリシーに関連付けます、インフラ テナントとオーバーレイ 1 VRF。 apic1(config-spine-bgp-vrf)# vrf context tenant infra vrf overlay-1
ステップ 6	neighbor neighbor-ip-address evpn	Configures the IP address for an EVPN BGP neighbor. apic1(config-spine-bgp-vrf)# neighbor 10.10.4.1 evpn
ステップ 7	label <i>label-name</i>	ネイバーにラベルを割り当てます。

	コマンドまたはアクション	目的
		<pre>apic1(config-spine-bgp-vrf-neighbor)# label evpn-aci</pre>
ステップ 8	update-source loopback ループバック IP アドレス vrf vrf	ネイバーループバック IP アドレスを更新の送信元を設定します。 <pre>apic1(config-spine-bgp-vrf-neighbor)# update-source loopback 10.10.4.3</pre>
ステップ 9	remote-as <i>AS-number</i>	Specifies the autonomous system (AS) number of the neighbor. The valid value can be from 1 to 4294967295. <pre>apic1(config-spine-bgp-vrf-neighbor)# remote-as 100</pre>
ステップ 10	exit	Returns to BGP VRF configuration mode. <pre>apic1(config-spine-bgp-vrf-neighbor)# exit</pre>
ステップ 11	neighbor neighbor-ip-address evpn	Configures the IP address for an EVPN BGP neighbor. <pre>apic1(config-spine-bgp-vrf)# neighbor 10.10.5.1 evpn</pre>
ステップ 12	label <i>label-name</i>	ネイバーにラベルを割り当てます。 <pre>apic1(config-spine-bgp-vrf-neighbor)# label evpn-aci2</pre>
ステップ 13	update-source loopback <i>loopback-ip-address</i> vrf vrf-name	ネイバーループバック IP アドレスを更新の送信元を設定します。 <pre>apic1(config-spine-bgp-vrf-neighbor)# update-source loopback 10.10.5.3</pre>
ステップ 14	remote-as <i>AS-number</i>	Specifies the autonomous system (AS) number of the neighbor. The valid value can be from 1 to 4294967295. <pre>apic1(config-spine-bgp-vrf-neighbor)# remote-as 100</pre>

NX-OS スタイル CLI を使用した BG EVPN のテナントの設定

このタスクは、次の手順で BGP EVPN のテナントを設定する方法を示しています。

手順

	コマンドまたはアクション	目的
ステップ 1	configure	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
		<code>apic1# configure</code>
ステップ 2	<code>tenant tenant-name</code>	テナントを作成するか、テナント設定モードを開始します。 <code>apic1(config)# tenant sky</code>
ステップ 3	<code>vrf context vrf-name</code>	テナントの VRF を作成します。 <code>apic1(config-tenant)# vrf context vrf-sky</code>
ステップ 4	<code>exit</code>	テナントコンフィギュレーションモードに戻ります。 <code>apic1(config-tenant-vrf)# exit</code>
ステップ 5	<code>bridge-domain bd-name</code>	ブリッジドメインを作成します <code>apic1(config-tenant)# bridge-domain bd-sky</code>
ステップ 6	<code>vrf member vrf-name</code>	VRF およびテナントにブリッジドメインを関連付けます。 <code>apic1(config-tenant-bd)# vrf member vrf-sky</code>
ステップ 7	<code>exit</code>	テナントコンフィギュレーションモードに戻ります。 <code>apic1(config-tenant-bd)# exit</code>
ステップ 8	<code>interface bridge-domain bd-name</code>	ブリッジドメインにインターフェイスを作成します。 <code>apic1(config-tenant)# interface bridge-domain bd_sky</code>
ステップ 9	<code>ip address A.B.C.D/LEN</code>	IP アドレスと長さをブリッジドメインインターフェイスに割り当てます。 <code>apic1(config-tenant-interface)# ip address 59.10.1.1/24</code>
ステップ 10	<code>exit</code>	テナントコンフィギュレーションモードに戻ります。 <code>apic1(config-tenant-interface)# exit</code>
ステップ 11	<code>bridge-domain bd-name</code>	ブリッジドメインを作成します <code>apic1(config-tenant)# bridge-domain bd-sky2</code>
ステップ 12	<code>vrf member vrf-name</code>	VRF およびテナントにブリッジドメインを関連付けます。

	コマンドまたはアクション	目的
		<code>apic1(config-tenant-bd) # vrf member vrf-sky</code>
ステップ 13	exit	テナント コンフィギュレーション モードに戻ります。 <code>apic1(config-tenant-bd) # exit</code>
ステップ 14	interface bridge-domain <i>bd-name</i>	ブリッジドメインにインターフェイスを作成します。 <code>apic1(config-tenant) # interface bridge-domain bd_sky2</code>
ステップ 15	ip address <i>A.B.C.D/LEN</i>	IP アドレスと長さをブリッジドメイン インターフェイスに割り当てます。 <code>apic1(config-tenant-interface) # ip address 59.11.1.1/24</code>
ステップ 16	exit	テナント コンフィギュレーション モードに戻ります。 <code>apic1(config-tenant-interface) # exit</code>

ルート マップの設定

このタスクは、BGP EVPN を介してブリッジドメインサブネットをアドバタイズするルートマップを設定する方法を示します。各ブリッジドメインは、一意のプロバイダラベルで、スパイン上のさまざまな BGP EVPN セッションを通してアドバタイズされます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure	コンフィギュレーションモードに入ります。 <code>apic1# configure</code>
ステップ 2	spine <i>spine-name</i>	スパインを作成するか、スパイン設定モードを開始します。 <code>apic1(config) # spine 111</code>
ステップ 3	vrf context tenant <i>tenant-name</i> vrf <i>vrf-name</i>	VRF を作成するか、VRF 設定モードを開始します。 <code>apic1(config-spine) # vrf context tenant sky vrf vrf_sky</code>
ステップ 4	address-family { ipv4 ipv6 } unicast	VRF の IPv4 または IPv6 ユニキャスト アドレス ファミリを設定します。

	コマンドまたはアクション	目的
		<pre>apic1(config-spine-vrf) # address-family ipv4 unicast</pre>
ステップ 5	route-target mode extended-community-number	アドレスファミリにエクスポートルートターゲットを割り当てます。 <pre>apic1(config-spine-vrf-af) # route-target export 100:1</pre>
ステップ 6	route-target mode extended-community-number	アドレスファミリにインポートルートターゲットを割り当てます。 <pre>apic1(config-spine-vrf-af) # route-target import 100:1</pre>
ステップ 7	exit	スパイン VRF 設定モードに戻ります。 <pre>apic1(config-spine-vrf-af) # exit</pre>
ステップ 8	route-map route-map-name	EVPN にルート マップを作成します (トランジットネットワークから学習したプレフィックス)。 <pre>apic1(config-spine-vrf) # route map rmap</pre>
ステップ 9	ip prefix-list ip-pl-name permit A.B.C.D/LEN	ルート マップに IP プレフィックス リストを追加して、指定されたサブネットからのトラフィックを許可します。 <pre>apic1(config-spine-vrf-route-map) # ip prefix-list pl permit 11.10.10.0/24</pre>
ステップ 10	match bridge-domain bd-name	ブリッジドメインに属するトラフィックを照合するルートマップを設定します。 <pre>apic1(config-spine-vrf-route-map) # match bridge-domain bd_sky</pre>
ステップ 11	exit	スパイン VRF ルート マップ設定モードに戻ります。 <pre>apic1(config-spine-vrf-route-map-match) # exit</pre>
ステップ 12	match prefix-list pl-name	指定されたプレフィックスリストを照合するルート マップを設定します。 <pre>apic1(config-spine-vrf-route-map) # match prefix-list pl</pre>
ステップ 13	exit	スパイン VRF ルート マップ設定モードに戻ります。 <pre>apic1(config-spine-vrf-route-map-match) # exit</pre>

	コマンドまたはアクション	目的
ステップ 14	exit	スパイン VRF 設定モードに戻ります。 apic1(config-spine-vrf-route-map)# exit
ステップ 15	evpn export map <i>route-map-name label consumer-label-name</i>	VRF にコンシューマ ラベルを割り当てます。 apic1(config-spine-vrf)# evpn export map rmap label evpn-aci
ステップ 16	route-map <i>route-map-name</i>	EVPN にルート マップを作成します (トランジットネットワークから学習したプレフィックス)。 apic1(config-spine-vrf)# route map rmap2
ステップ 17	match bridge-domain <i>bd-name</i>	ブリッジドメインに属するトラフィックを照合するルートマップを設定します。 apic1(config-spine-vrf-route-map)# match bridge-domain bd_sky
ステップ 18	exit	スパイン VRF ルート マップ設定モードに戻ります。 apic1(config-spine-vrf-route-map-match)# exit
ステップ 19	match prefix-list <i>pl-name</i>	指定されたプレフィックスリストを照合するルート マップを設定します。 apic1(config-spine-vrf-route-map)# match prefix-list pl
ステップ 20	exit	スパイン VRF ルート マップ設定モードに戻ります。 apic1(config-spine-vrf-route-map-match)# exit
ステップ 21	exit	スパイン VRF 設定モードに戻ります。 apic1(config-spine-vrf-route-map)# exit
ステップ 22	evpn export map <i>route-map-name label consumer-label-name</i>	VRF にコンシューマ ラベルを割り当てます。 apic1(config-spine-vrf)# evpn export map rmap label evpn-aci2
ステップ 23	external-l3 <i>epgepg-name</i>	apic1(config-spine-vrf)# external-l3 epg l3_sky

	コマンドまたはアクション	目的
ステップ 24	vrf member <i>vrf-name</i>	apic1(config-spine-vrf-l3ext-epg) # vrf member vrf_sky
ステップ 25	match ip <i>A.B.C.D/LEN</i>	EPG の一部であるホストを識別するサブネットを設定します。 apic1(config-spine-vrf-l3ext-epg) # match ip 80.10.1.0/24

NX-OS スタイル CLI を使用して DCIG への配布の BGP EVPN タイプ 2 のホスト ルートの有効化

手順

	コマンドまたはアクション	目的
ステップ 1	BGP アドレス ファミリ configuration mode(設定モード、コンフィギュレーションモード) で、次のコマンドを DCIG に配布 EVPN タイプ 2 のホスト ルートを設定します。 例 : <pre>apic1(config)# leaf 101 apic1(config-leaf)# template bgp address-family bgpAf1 tenant bgp_t1 apic1(config-bgp-af)# distance 250 240 230 apic1(config-bgp-af)# host-rt-enable apic1(config-bgp-af)# exit</pre>	このテンプレートは、テナント bgp_t1 は VRF の導入を持つすべてのノードで利用可能になります。配布 EVPN タイプ 2 のホスト ルートを無効にするには、次のように入力します。、 no ホスト-rt-enable コマンド。

NX-OS スタイル CLI を使用した Cisco ACI GOLF 設定の例:

次の例を設定する CLI コマンドの show GOLF サービスで、OSPF over スパイン スイッチに接続されている WAN ルータの BGP EVPN プロトコルを使用します。

設定、BGP EVPN のテナント インフラ

次の例を設定する方法を示しています、インフラ VLAN ドメイン、VRF、インターフェイスの IP アドレッシングを含む、BGP EVPN および OSPF のテナントします。

```
configure
vlan-domain evpn-dom dynamic
exit
```

```

spine 111
  # Configure Tenant Infra VRF overlay-1 on the spine.
  vrf context tenant infra vrf overlay-1
    router-id 10.10.3.3
    exit

  interface ethernet 1/33
    vlan-domain member golf_dom
    exit
  interface ethernet 1/33.4
    vrf member tenant infra vrf overlay-1
    mtu 1500
    ip address 5.0.0.1/24
    ip router ospf default area 0.0.0.150
    exit
  interface ethernet 1/34
    vlan-domain member golf_dom
    exit
  interface ethernet 1/34.4
    vrf member tenant infra vrf overlay-1
    mtu 1500
    ip address 2.0.0.1/24
    ip router ospf default area 0.0.0.200
    exit

  router ospf default
    vrf member tenant infra vrf overlay-1
    area 0.0.0.150 loopback 10.10.5.3
    area 0.0.0.200 loopback 10.10.4.3
    exit
  exit

```

スパイン ノード上の BGP の設定

次の例では、BGP EVPN をサポートする BGP を設定する方法を示します。

```

Configure
spine 111
router bgp 100
  vrf member tenant infra vrf overlay- 1
    neighbor 10.10.4.1 evpn
    label golf_aci
    update-source loopback 10.10.4.3
    remote-as 100
    exit
  neighbor 10.10.5.1 evpn
    label golf_aci2
    update-source loopback 10.10.5.3
    remote-as 100
    exit
  exit
exit

```

BGP EVPN のテナントの設定

次の例では、BGPEVPN、BGPEVPNセッションで提供されるゲートウェイ サブネットを含むのテナントを設定する方法を示します。

```

configure
  tenant sky
    vrf context vrf_sky

```

```

exit
bridge-domain bd_sky
  vrf member vrf_sky
exit
interface bridge-domain bd_sky
  ip address 59.10.1.1/24
exit
bridge-domain bd_sky2
  vrf member vrf_sky
exit
interface bridge-domain bd_sky2
  ip address 59.11.1.1/24
exit
exit

```

BGP EVPN ルート ターゲット、ルートマップと、テナントのプレフィックス EPG の設定

次の例では、BGP EVPN を介してブリッジ ドメイン サブネットをアドバタイズするルートマップを設定する方法を示します。

```

configure
spine 111
  vrf context tenant sky vrf vrf_sky
    address-family ipv4 unicast
      route-target export 100:1
      route-target import 100:1
    exit

    route-map rmap
      ip prefix-list p1 permit 11.10.10.0/24
      match bridge-domain bd_sky
      exit
      match prefix-list p1
      exit

    evpn export map rmap label golf_aci

    route-map rmap2
      match bridge-domain bd_sky
      exit
      match prefix-list p1
      exit
    exit

    evpn export map rmap label golf_aci2

  external-l3 epg l3_sky
    vrf member vrf_sky
    match ip 80.10.1.0/24
    exit

```

DCIG への EVPN タイプ2 ルート配信のトラブルシューティング

EVPN トポロジで最適なトラフィックの転送を行うには、ファブリック スパインを有効にし、EVPN タイプ2 (MAC-IP) ルートを使用して、BGP EVPN タイプ5 (IP プレフィックス) ルートの形式で公的BDサブネットとともにホストルートを Data Center Interconnect Gateway (DCIG) に配信できます。HostLeak オブジェクトを使用して有効になります。ルート配信に問題が発生した場合、このトピックでのトラブルシューティングの手順を使用します。

手順

- ステップ 1** スパインスイッチ CLI で次のようなコマンドを入力して、問題の VRF AF で HostLeak オブジェクトが有効になっていることを確認します。

例 :

```
spinel# ls /mit/sys/bgp/inst/dom-apple/af-ipv4-ucast/
ctrl-l2vpn-evpn ctrl-vpnv4-ucast hostleak summary
```

- ステップ 2** スパインスイッチ CLI で次のようなコマンドを入力して、設定 MO が BGP により正常に処理されていることを確認します。

例 :

```
spinel# show bgp process vrf apple
```

次と同様の出力を検索します。

```
Information for address family IPv4 Unicast in VRF apple
Table Id           : 0
Table state        : UP
Table refcount     : 3
Peers              Active-peers  Routes    Paths    Networks  Aggregates
0                  0          0         0        0         0

Redistribution
None

Wait for IGP convergence is not configured
GOLF EVPN MAC-IP route is enabled
EVPN network next-hop 192.41.1.1
EVPN network route-map map_pfxleakctrl_v4
Import route-map rtctrlmap-apple-v4
EVPN import route-map rtctrlmap-evpn-apple-v4
```

- ステップ 3** 公的 BD サブネットが EVPN タイプ 5 ルートとして DCIG に提供されたことを確認します。

例 :

```
spinel# show bgp l2vpn evpn 10.6.0.0 vrf overlay-1
Route Distinguisher: 192.41.1.5:4123 (L3VNI 2097154)
BGP routing table entry for [5]:[0]:[0]:[16]:[10.6.0.0]:[0.0.0.0]/224, version 2088
Paths: (1 available, best #1)
Flags: (0x000002 00000000) on xmit-list, is not in rib/evpn
Multipath: eBGP iBGP

Advertised path-id 1
Path type: local 0x4000008c 0x0 ref 1, path is valid, is best path
AS-Path: NONE, path locally originated
192.41.1.1 (metric 0) from 0.0.0.0 (192.41.1.5)
Origin IGP, MED not set, localpref 100, weight 32768
Received label 2097154
Community: 1234:444
Extcommunity:
RT:1234:5101
4BYTEAS-GENERIC:T:1234:444

Path-id 1 advertised to peers:
50.41.50.1
```

[パス タイプ] エントリで、**ref 1** は 1 個のルートが送信されたことその 1 つのルートが送信されたことを示します。

ステップ4 EVPN ピアにアドバタイズされたホスト ルートが EVPN タイプ2 MAC IP ルートであることを確認します。

例：

```
spine1# show bgp l2vpn evpn 10.6.41.1 vrf overlay-1
Route Distinguisher: 10.10.41.2:100 (L2VNI 100)
BGP routing table entry for [2]:[0]:[2097154]:[48]:[0200.0000.0002]:[32]:[10.6.41.1]/272, version 1146
Shared RD: 192.41.1.5:4123 (L3VNI 2097154)
Paths: (1 available, best #1)
Flags: (0x00010a 00000000) on xmit-list, is not in rib/evpn
Multipath: eBGP iBGP

Advertised path-id 1
Path type: local 0x4000008c 0x0 ref 0, path is valid, is best path
AS-Path: NONE, path locally originated
EVPN network: [5]:[0]:[0]:[16]:[10.6.0.0]:[0.0.0.0] (VRF apple)
  10.10.41.2 (metric 0) from 0.0.0.0 (192.41.1.5)
    Origin IGP, MED not set, localpref 100, weight 32768
    Received label 2097154 2097154
    Extcommunity:
      RT:1234:16777216

Path-id 1 advertised to peers:
50.41.50.1
```

[共有 RD] ラインは、EVPN タイプ2 ルートと BD サブネットとで共有された RD/VNI を示します。

[EVPN ネットワーク] ラインは、BD サブネットの EVPN タイプ5 のルートを表示します。

[ピアにアドバタイズされたパス ID] は、EVPN ピアにパスがアドバタイズされたことを示します。

ステップ5 DCIG デバイスで次のようなコマンドを入力して（DCIG が下の例で Cisco ASR 9000 スイッチであることを仮定します）、EVPN ピア（DCIG）が正常なタイプ2 MAC IP ルートを受信し、ホスト ルートが正常に特定の VRF にインポートされたことを確認します。

例：

```
RP/0/RSP0/CPU0:asr9k#show bgp vrf apple-2887482362-8-1 10.6.41.1
Tue Sep  6 23:38:50.034 UTC
BGP routing table entry for 10.6.41.1/32, Route Distinguisher: 44.55.66.77:51
Versions:
  Process          bRIB/RIB   SendTblVer
  Speaker          2088      2088
Last Modified: Feb 21 08:30:36.850 for 28w2d
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    192.41.1.1 (metric 42) from 10.10.41.1 (192.41.1.5)
    Received Label 2097154
    Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
    imported
    Received Path ID 0, Local Path ID 1, version 2088
```



```
Community: 1234:444
Extended community: 0x0204:1234:444 Encapsulation Type:8 Router
MAC:0200.c029.0101 RT:1234:5101
RIB RNH: table_id 0xe0000190, Encap 8, VNI 2097154, MAC Address: 0200.c029.0101,
IP Address: 192.41.1.1, IP table_id 0x00000000
Source AFI: L2VPN EVPN, Source VRF: default,
Source Route Distinguisher: 192.41.1.5:4123
```

この出力では、受信した RD、ネクスト ホップ、属性はタイプ 2 ルートおよび BD サブネットと同じです。

Multipod_Fabric

マルチポッド ファブリックについて

マルチポッドは、隔離されたコントロールプレーンプロトコルを持つ複数のポッドで構成された、障害耐性の高いファブリックのプロビジョニングを可能にします。また、マルチポッドでは、さらに柔軟にリーフとスパインスイッチ間のフルメッシュ配線を行うことができます。たとえば、リーフスイッチが異なるフロアや異なる建物にまたがって分散している場合、マルチポッドでは、フロアごと、または建物ごとに複数のポッドをプロビジョニングし、スパインスイッチを通じてポッド間を接続することができます。

マルチポッドは、異なるポッドの ACI スパイン間のコントロールプレーン通信プロトコルとして MP-BGP EVPN を使用します。

WAN ルータは、スパイン スwitch に直接接続されるか、またはボーダー リーフ スwitch に接続された IPN 内でプロビジョニングできます。マルチポッドはすべてのポッドに単一の APIC クラスタを使用します。そのため、すべてのポッドが単一のファブリックとして機能します。ポッド全体にわたって個々の APIC コントローラが配置されますが、それらはすべて単一の APIC クラスタの一部です。

マルチポッド ファブリックでのスイッチの割り当て

始める前に

ノード グループ ポリシーと L3Out ポリシーがすでに作成されています。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	[no] system switch-id serial-number switch-id switch-name [pod pod-id] [role {leaf spine}] 例 : <pre>apic1(config)# system switch-id SAL1748H56D 201 ifav4-spine1 pod 1 role spine</pre>	マルチポッドファブリック内の各スイッチに対して、スイッチの関連するポッドとロール（リーフ/スパイン）を宣言します。マルチポッドファブリック内の各リーフとスパイン スイッチにこのコマンドを繰り返します。
ステップ 3	[no] system pod pod-id tep-pool ip-prefix/length 例 : <pre>apic1(config)# system pod 1 tep-pool 10.0.0.0/16</pre>	ポッドにトンネルエンドポイント IP アドレスプールを設定します。マルチポッドファブリック内の各ポッドにこのコマンドを繰り返します。

例

次に、2 ポッドファブリックでスパインとリーフ スイッチを割り当てる例を示します。

```
apic1# configure
apic1(config)# system switch-id SAL1748H56D 201 ifav4-spine1 pod 1 role spine
apic1(config)# system switch-id SAL1938P7A6 202 ifav4-spine3 pod 1 role spine
apic1(config)# system switch-id SAL1819RXP4 101 ifav4-leaf1 pod 1 role leaf
apic1(config)# system switch-id SAL1803L25H 102 ifav4-leaf2 pod 1 role leaf
apic1(config)# system switch-id SAL1934MNY0 103 ifav4-leaf3 pod 1 role leaf
apic1(config)# system switch-id SAL1934MNY3 104 ifav4-leaf4 pod 1 role leaf
apic1(config)# system switch-id SAL1931LA3B 203 ifav4-spine2 pod 2 role spine
apic1(config)# system switch-id FGE173400A9 204 ifav4-spine4 pod 2 role spine
apic1(config)# system switch-id SAL1938PHBB 105 ifav4-leaf5 pod 2 role leaf
apic1(config)# system switch-id SAL1942R857 106 ifav4-leaf6 pod 2 role leaf
apic1(config)# system pod 1 tep-pool 10.0.0.0/16
apic1(config)# system pod 2 tep-pool 10.1.0.0/16
```

次のタスク

ファブリックの外部接続を設定します。

マルチポッドファブリックのファブリックの外部接続の設定

始める前に

- ノードグループポリシーと L3Out ポリシーがすでに作成されています。
- スイッチがポッドに割り当てられています。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apicl# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] fabric-external controller-number 例 : <code>apicl(config)# fabric-external 1</code>	
ステップ 3	[no]bgp password type automatic_with_full_mesh [ピアリング-パスワード] [{ evpn peeringautomatic_with_rr }] 例 : <code>apicl(config-fabric-external)# bgp evpn peering</code>	BGP EVPN ピアリング プロファイルを設定します。ピアリングパスワードを設定できます。また、タイプはフルメッシュに設定するか、ルータリフレクタで設定できます。
ステップ 4	[no] pod pod-id 例 : <code>apicl(config-fabric-external)# pod 1</code>	設定用のポッドを選択します。
ステップ 5	[no] interpod data hardware-proxy ip-addr/mask 例 : <code>apicl(config-fabric-external-pod)# interpod data hardware-proxy 100.11.1.1/32</code>	ポッド間トラフィックの各ポートにユニキャストハードウェアプロキシ IP アドレスを設定します。
ステップ 6	[no] bgp evpn peering [password peering-password] [type {automatic_with_full_mesh automatic_with_rr}] 例 : <code>apicl(config-fabric-external-pod)# bgp evpn peering</code>	
ステップ 7	exit 例 : <code>apicl(config-fabric-external-pod)# exit</code>	BGP EVPN ピアリング プロファイルの設定に戻ります。
ステップ 8	マルチポッドファブリック内の各ポッドにステップ 4～7 を繰り返します。	

	コマンドまたはアクション	目的
ステップ 9	[no] route-map interpod-import 例 : apic1(config-fabric-external)# route-map interpod-import	OSPF プロトコルを介してファブリック内への参加を許可されるポッド間ネットワーク (IPN) のサブネットを含むルート マップを設定します。
ステップ 10	[no] ip prefix-list prefix-list-name [permit ip-address/len] 例 : apic1(config-fabric-external-route-map)# ip prefix-list default permit 0.0.0.0/0	
ステップ 11	exit 例 : apic1(config-fabric-external-route-map)# exit	ファブリック外コンフィギュレーション モードに戻ります。
ステップ 12	[no] route-target extended ASN4:NN 例 : apic1(config-fabric-external)# route-target extended 5:16	ルートターゲットは拡張コミュニティ属性として伝送されます。AA4:NN2 形式でコミュニティ番号を入力します (1-4294967295: 1-65535)。
ステップ 13	exit	

例

次に、マルチポッドファブリックにファブリック外部接続を設定する例を示します。

```
apic1# configure
apic1(config)# fabric-external 1
apic1(config-fabric-external)# bgp evpn peering
apic1(config-fabric-external)# pod 1
apic1(config-fabric-external-pod)# interpod data hardware-proxy 100.11.1.1/32
apic1(config-fabric-external-pod)# bgp evpn peering
apic1(config-fabric-external-pod)# exit
apic1(config-fabric-external)# pod 2
apic1(config-fabric-external-pod)# interpod data hardware-proxy 200.11.1.1/32
apic1(config-fabric-external-pod)# bgp evpn peering
apic1(config-fabric-external-pod)# exit
apic1(config-fabric-external)# route-map interpod-import
apic1(config-fabric-external-route-map)# ip prefix-list default permit 0.0.0.0/0
apic1(config-fabric-external-route-map)# exit
apic1(config-fabric-external)# route-target extended 5:16
apic1(config-fabric-external)# exit
```

次のタスク

スパイン インターフェイスおよび OSPF を設定します。

マルチポッド ファブリックのスパインインターフェイスと OSPF の設定

始める前に

- スイッチがポッドに割り当てられています。
- VLAN ドメインが存在している必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	spine spine-id 例 : apicl(config)# spine 104	スパインスイッチは、101 ~ 4000 の範囲内の ID 番号か、または「spine1」などの名前指定できます。
ステップ 3	[no] vrf context tenant infra vrf vrf-name 例 : apicl(config-spine)# vrf context tenant infra vrf overlay-1	
ステップ 4	[no] router-id A.B.C.D 例 : apicl(config-spine-vrf)# router-id 201.201.201.201	ルータ識別子 (ID) を設定します。
ステップ 5	exit 例 : apicl(config-spine-vrf)# exit	スパイン コンフィギュレーション モードに戻ります。
ステップ 6	[no] interface ethernet slot/port 例 : apicl(config-spine)# interface ethernet 1/1	
ステップ 7	[no] vlan-domain member domain-name 例 : apicl(config-spine)# vlan-domain member 13Dom	The VLAN domain must already exist, having been created using the vlan-domain domain-name command in the global configuration mode.

	コマンドまたはアクション	目的
ステップ 8	exit 例 : apic1(config-spine-if) # exit	スパインコンフィギュレーションモードに戻ります。
ステップ 9	[no] interface ethernet <i>type/slot.subinterface</i> 例 : apic1(config-spine) # interface ethernet 1/1.4	サブインターフェイスのカプセル化は 4 である必要があります。
ステップ 10	[no] vrf member tenant infra vrf vrf-name 例 : apic1(config-spine-if) # vrf member tenant infra vrf overlay-1	テナント VRF のメンバーとしてインターフェイスを設定します。
ステップ 11	[no] ip address ip-address 例 : apic1(config-spine-if) # ip address 201.1.1.1/30	
ステップ 12	[no] ip router ospf default area 0.0.0.0 例 : apic1(config-spine-if) # ip router ospf default area 0.0.0.0	
ステップ 13	[no] ip ospf cost cost 例 : apic1(config-spine-if) # ip ospf cost 1	
ステップ 14	exit 例 : apic1(config-spine-if) # exit	スパインコンフィギュレーションモードに戻ります。
ステップ 15	ステップ 6 (341 ページ) ~ ステップ 14 (342 ページ) を繰り返し、他のインターフェイスを追加します。	
ステップ 16	[no] router ospf default 例 : apic1(config-spine) # router ospf default	
ステップ 17	[no] vrf member tenant infra vrf vrf-name 例 :	

	コマンドまたはアクション	目的
	<code>apicl(config-spine-ospf)# vrf member tenant infra vrf overlay-1</code>	
ステップ 18	[no] area area loopback ip-address 例 : <code>apicl(config-spine-ospf-vrf)# area 0.0.0.0 loopback 201.201.201.201</code>	OSPF を介してループバック アドレスをアドバタイズします。このアドレスは、ピアリングの場合に BGP EVPN セッションで使用されます。
ステップ 19	[no] area area interpod peering 例 : <code>apicl(config-spine-ospf-vrf)# area 0.0.0.0 interpod peering</code>	OSPF エリアでポッド間ピアリングを有効にし、OSPF によってアドバタイズされたループバックアドレスを使用して BGP EVPN セッションを自動的に設定します。
ステップ 20	exit 例 : <code>apicl(config-spine-ospf-vrf)# exit</code>	OSPF コンフィギュレーション モードに戻ります。
ステップ 21	exit 例 : <code>apicl(config-spine-ospf)# exit</code>	スパイン コンフィギュレーション モードに戻ります。
ステップ 22	exit 例 : <code>apicl(config-spine)# exit</code>	グローバル コンフィギュレーション モードに戻ります。
ステップ 23	ステップ 2 (341 ページ) ~ ステップ 22 (343 ページ) を繰り返し、他のスパイン スイッチを設定します。	

例

```

apicl# configure

# CONFIGURE FIRST SPINE

apicl(config)# spine 201
apicl(config-spine)# vrf context tenant infra vrf overlay-1
apicl(config-spine-vrf)# router-id 201.201.201.201
apicl(config-spine-vrf)# exit

apicl(config-spine)# interface ethernet 1/1
apicl(config-spine-if)# vlan-domain member l3Dom
apicl(config-spine-if)# exit
apicl(config-spine)# interface ethernet 1/1.4
apicl(config-spine-if)# vrf member tenant infra vrf overlay-1
apicl(config-spine-if)# ip address 201.1.1.1/30
apicl(config-spine-if)# ip router ospf default area 0.0.0.0

```

```

apic1(config-spine-if)# ip ospf cost 1
apic1(config-spine-if)# exit

apic1(config-spine)# interface ethernet 1/2
apic1(config-spine-if)# vlan-domain member l3Dom
apic1(config-spine-if)# exit
apic1(config-spine)# interface ethernet 1/2.4
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1
apic1(config-spine-if)# ip address 201.2.1.1/30
apic1(config-spine-if)# ip router ospf default area 0.0.0.0
apic1(config-spine-if)# ip ospf cost 1
apic1(config-spine-if)# exit

apic1(config-spine)# router ospf default
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 0.0.0.0 loopback 201.201.201.201
apic1(config-spine-ospf-vrf)# area 0.0.0.0 interpod peering
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)# exit

# CONFIGURE SECOND SPINE

apic1(config)# spine 202
apic1(config-spine)# vrf context tenant infra vrf overlay-1
apic1(config-spine-vrf)# router-id 202.202.202.202
apic1(config-spine-vrf)# exit

apic1(config-spine)# interface ethernet 1/2
apic1(config-spine-if)# vlan-domain member l3Dom
apic1(config-spine-if)# exit
apic1(config-spine)# interface ethernet 1/2.4
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1
apic1(config-spine-if)# ip address 202.1.1.1/30
apic1(config-spine-if)# ip router ospf default area 0.0.0.0
apic1(config-spine-if)# ip ospf cost 1
apic1(config-spine-if)# exit

apic1(config-spine)# router ospf default
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 0.0.0.0 loopback 202.202.202.202
apic1(config-spine-ospf-vrf)# area 0.0.0.0 interpod peering
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)# exit

# CONFIGURE ADDITIONAL SPINES

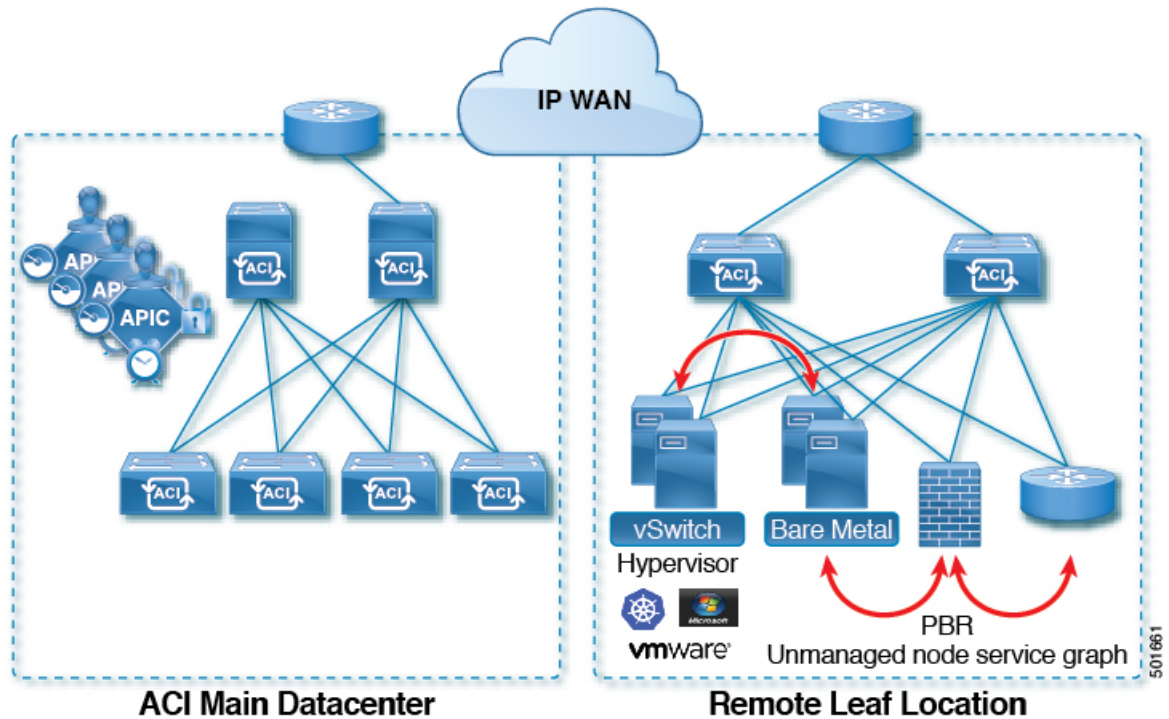
```

リモート リーフ スイッチ

ACI ファブリックのリモート リーフ スイッチについて

ACI ファブリックの展開では、ローカル スパイン スイッチまたは APIC が接続されていない Cisco ACI リーフ スイッチのリモート データセンタに、ACI サービスと APIC 管理を拡張できます。

図 24: リモート リーフ トポロジ



リモート リーフ スイッチがファブリックの既存のポッドに追加されます。メイン データ センタに展開されるすべてのポリシーはリモート スイッチで展開され、ポッドに属するローカル リーフ スイッチのように動作します。このトポロジでは、すべてのユニキャストトラフィックはレイヤ 3 上の VXLAN を経由します。レイヤ 2 ブロードキャスト、不明なユニキャスト、マルチキャスト (BUM) メッセージは、マルチキャストを使用することなく、**Head End Replication (HER)** トンネルを使用して送信されます。リモートサイトのすべてのローカルトラフィックは、物理または仮想にかかわらずエンドポイント間で直接切り替えられます。スパイン スイッチ プロキシを使用する必要があるすべてのトラフィックは、メイン データ センタに転送されます。

APIC システムは、起動時にリモート リーフ スイッチを検出します。その時点から、ファブリックの一部として APIC で管理できます。



- (注)
- VRF 間のすべてのトラフィックは、転送される前にスパイン スイッチに移動します。
 - リモート リーフを解除する前に、vPC を最初に削除する必要があります。

ウィザードを使用するか (使用しない場合も)、REST API または NX-OS スタイル CLI を使用して、APIC GUI のリモート リーフを設定できます。

リモートのリーフ ハードウェアの要件

リモートのリーフ スイッチの機能には、次のスイッチがサポートされています。

ファブリック スパイン スイッチ

WAN ルータに接続された ACI メイン データ センターにスパイン スイッチでの次のスパイン スイッチがサポートされています。

- 固定スパイン スイッチの Cisco Nexus 9000 シリーズ N9K C9364C
N9K-X9732C-EX または N9K-X9736C-FX ラインカードをモジュラ スパイン スイッチ
- 古い生成スパイン スイッチは、固定スパイン スイッチ N9K C9336PQ または N9K X9736PQ ラインカードでモジュラ スパイン スイッチなどのメイン データ センターではサポートが次世代のみのスパイン スイッチは、WAN への接続をサポートします。

リモート リーフ スイッチ

- リモートのリーフ スイッチ、後で (たとえば N9K-C93180LC-EX) EX で終了する名前と Cisco Nexus 9000 シリーズ スイッチのみがサポートされています。



(注) Cisco Nexus 9000 N9K-C9336C-FX スイッチは、リモートのリーフ スイッチのサポートされていません。

- リモートのリーフ スイッチする必要がありますにイメージを実行する、スイッチ 13.1.x 以降 (aci n9000 dk9.13.1.x.x.bin) 検出できる前にします。これにより、リーフ スイッチでの手動アップグレードが必要です。

制約事項と制限



- (注) Cisco APIC のリリースでは、以前にサポートされていませんでした 3.2(x) が、次の機能がサポートされます。
- リモートのリーフ スイッチに接続されている FEX デバイス
 - リモートのリーフ スイッチまたはリモート リーフ スイッチとローカル リーフ スイッチ間でという原子カウンタという
 - Cisco VXLAN での VLAN とシスコの AV で AV
 - VXLAN での VLAN と ACI の仮想エッジで Cisco ACI 仮想エッジ

リモート リーフ機能では、次の導入と設定がサポートされていません。

- APIC コントローラは、リモートのリーフ スイッチに直接接続

- VPC ドメインでのリモート リーフ スイッチで孤立ポート チャンネルまたは物理ポート
- コンシューマ、プロバイダー、およびサービス ノードがすべてスイッチ vPC モードでは、リモートのリーフに接続されている場合に、リモート ロケーション内で転送ローカル トラフィックがサポートのみとサービス ノードの統合がなければ、

このリリースでは、次の機能を除くで、リモートのリーフスイッチでは、ファブリックおよびテナントの完全なポリシーがサポートされています。

- ACI マルチサイト
- レイヤ 2 (スタティック Epg) を除く接続外部
- 802.1 q トンネリング
- EPG のための Q-in-Q カプセル化マッピング
- VzAny 契約とサービスをコピーします。
- リモートのリーフ スイッチの FCoE 接続
- ブリッジ ドメインまたは Epg のカプセル化をフラグディングします。
- 高速リンク フェールオーバー ポリシー
- 遠隔地での管理対象のサービス グラフに接続されたデバイス
- NetFlow
- トラフィック ストーム制御
- クラウド秒および MacSec 暗号化
- ファーストホップ セキュリティ
- PTP
- レイヤ 3 マルチキャスト リモート リーフ スイッチ上のルーティング
- リモートのリーフ スイッチでの PBR トラッキング
- Openstack および Kubernetes VMM ドメイン
- メンテナンス モード
- ウィザードのトラブルシューティング
- 遠隔地での中継 L3Out
- 同じリモート データ センターで同じポッドおよびポッド全体の 2 つのリモート リーフ スイッチ間で直接トラフィックの転送

WAN ルータとリモート リーフ設定の注意事項

リモート リーフが検出され APIC 管理に組み込まれる前に、WAN ルータとリモート リーフ スイッチを設定する必要があります。

次の要件に従い、ファブリック スパイン スイッチの外部インターフェイスとリモート リーフ スイッチ ポートに接続する WAN ルータを接続します。

WAN ルータ

- エリア ID、タイプ、コストなど、同じ詳細を有するインターフェイスで OSPF を有効にします。
- メインファブリックの各 APIC の IP アドレスにつながるインターフェイスで DHCP リレーを設定します。
- スパイン スイッチで VLAN 5 インターフェイスに接続する WAN ルータのインターフェイスは、通常のマルチポッドネットワークに接続するインターフェイス以外に、異なる VRF に存在する必要があります。

リモート リーフ スイッチ

- ファブリック ポートの 1 つから直接接続して、アップストリーム ルータにリモート リーフ スイッチを接続します。アップストリーム ルータへの次の接続がサポートされています。
 - 40 Gbps 以上の接続
 - QSFP-SFP アダプタでは、1/10 G SFP がサポートされています

WAN の帯域幅は最小で 100 Mbps、最大。サポートされている遅延は 300 ミリ秒です。

- 上記が推奨されますが、vPC とリモート リーフ スイッチのペアを接続する必要はありません。vPC の両端にあるスイッチは、同じリモート データセンタのリモート リーフ スイッチである必要があります。
- 一意の IP アドレスを持つ VLAN 4 でレイヤ 3 サブインターフェイスとしてノース バウンド インターフェイスを設定します。

リモートのリーフスイッチからルータに1個以上のインターフェイスを接続する場合、一意の IP アドレスで各インターフェイスを設定します。
- インターフェイスで OSPF を有効にします。
- リモート リーフ スイッチ内の TEP プール サブネットの IP アドレスは、ポッド TEP サブネットプールと重複しないようにする必要があります。使用されるサブネットは /24 以下である必要があります。
- マルチポッドがサポートされますが、リモート リーフ機能は必要ありません。
- 単一ポッドファブリックのポッドをリモート リーフ スイッチに接続するとき、スパイン スイッチから WAN ルータへ、リモート リーフ スイッチから WAN ルータへ L3Out を設定し、これは両方ともスイッチ インターフェイスで VLAN-4 を使用します。

- マルチポッド ファブリックのポッドをリモート リーフ スイッチに接続するとき、スパイン スイッチから WAN ルータへ、リモート リーフ スイッチから WAN ルータへ L3Out を設定し、これは両方ともスイッチ インターフェイスで VLAN-4 を使用します。また、VLAN-5 を使用してマルチポッド内部 L3Out を設定し、リモート リーフ スイッチを宛先としてポッドを通過するトラフィックをサポートします。VLAN 4 および VLAN 5 を使用する限り、通常のマルチポッドおよびマルチポッド内部接続は、同じ物理インターフェイスで設定できます。
- マルチポッド内部 L3Out を設定している場合、通常のマルチポッド L3Out として同じルータ ID を使用しますが、ルータ ID の [ループバック アドレスとしてルータ ID を使用する] オプションを選択解除して、異なるループバック IP アドレスを設定します。これで ECMP が機能します。

NX-OS スタイル CLI を使用したリモート リーフの設定

この例では、リーフ スイッチがメインのファブリック ポッドと通信できるようにするため、スパイン スイッチとリモート リーフ スイッチを設定しています。

始める前に

- IPN ルータとリモート スイッチはアクティブで設定されています。 [WAN ルータとリモート リーフ設定の注意事項 \(348 ページ\)](#) を参照してください。
- リモート リーフ スイッチは、13.1.x 以降 (aci n9000 dk9.13.1.x.x.bin) のスイッチ イメージを実行しています。
- リモート リーフ スイッチを追加する予定のポッドが作成され、設定されています。

手順

ステップ 1 ポッド 2 のリモート ロケーション 5 で TEP プールを定義します。

ネットワーク マスクは /24 以下である必要があります。

次の新しいコマンドを使用します : **system remote-leaf-site site-id pod pod-id tep-pool ip-address-and-netmask**

例 :

```
apic1(config)# system remote-leaf-site 5 pod 2 tep-pool 192.0.0.0/16
```

ステップ 2 ポッド 2 の、リモート リーフ サイト 5 にリモート リーフ スイッチを追加します。

次のコマンドを使用します : **system switch-id serial-number node-id leaf-switch-namepod pod-id remote-leaf-site remote-leaf-site-id node-type remote-leaf-wan**

例 :

```
apic1(config)# system switch-id FDO210805SKD 109 ifav4-leaf9 pod 2
remote-leaf-site 5 node-type remote-leaf-wan
```

ステップ 3 VLAN 4 を含む VLAN で VLAN ドメインを設定します。

例：

```
apic1(config)# vlan-domain ospfDom
apic1(config-vlan)# vlan 4-5
apic1(config-vlan)# exit
```

ステップ 4 インフラ テナントに 2 つの L3Out を設定します。1 つはリモート リーフ接続のために、もう 1 つはマルチポッド IPN のためです。

例：

```
apic1(config)# tenant infra
apic1(config-tenant)# l3out rl-wan
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# l3out ipn-multipodInternal
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# exit
apic1(config)#
```

ステップ 5 L3Out が使用する、スパイン スイッチ インターフェイスとサブインターフェイスを設定します。

例：

```
apic1(config)# spine 201
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-vrf)# exit
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-vrf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36
apic1(config-spine-if)# vlan-domain member ospfDom
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf default
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.4
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-if)# ip router ospf default area 5
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf multipod-internal
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out ipn-multipodInternal
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.5
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-if)# ip router ospf multipod-internal area 5
apic1(config-spine-if)# exit
apic1(config-spine)# exit
apic1(config)#
```

ステップ 6 メインのファブリック ポッドと通信するために使用するリモートのリーフ スイッチ インターフェイスとサブインターフェイスを設定します。

例 :

```
(config)# leaf 101
apic1(config-leaf)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-vrf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49
apic1(config-leaf-if)# vlan-domain member ospfDom
apic1(config-leaf-if)# exit
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-leaf-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49.4
apic1(config-leaf-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-if)# ip router ospf default area 5
apic1(config-leaf-if)# exit
```

例

次の例は、ダウンロード可能な設定を示しています:

```
apic1# configure
apic1(config)# system remote-leaf-site 5 pod 2 tep-pool 192.0.0.0/16
apic1(config)# system switch-id FDO210805SKD 109 ifav4-leaf9 pod 2
remote-leaf-site 5 node-type remote-leaf-wan
apic1(config)# vlan-domain ospfDom
apic1(config-vlan)# vlan 4-5
apic1(config-vlan)# exit
apic1(config)# tenant infra
apic1(config-tenant)# l3out rl-wan-test
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# l3out ipn-multipodInternal
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# exit
apic1(config)#
apic1(config)# spine 201
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-vrf)# exit
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-vrf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36
apic1(config-spine-if)# vlan-domain member ospfDom
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf default
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.4
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
```

```

apic1(config-spine-if)# ip router ospf default area 5
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf multipod-internal
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out ipn-multipodInternal
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.5
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-if)# ip router ospf multipod-internal area 5
apic1(config-spine-if)# exit
apic1(config-spine)# exit
apic1(config)#
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-vrf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49
apic1(config-leaf-if)# vlan-domain member ospfDom
apic1(config-leaf-if)# exit
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-leaf-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49.4
apic1(config-leaf-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-if)# ip router ospf default area 5
apic1(config-leaf-if)# exit

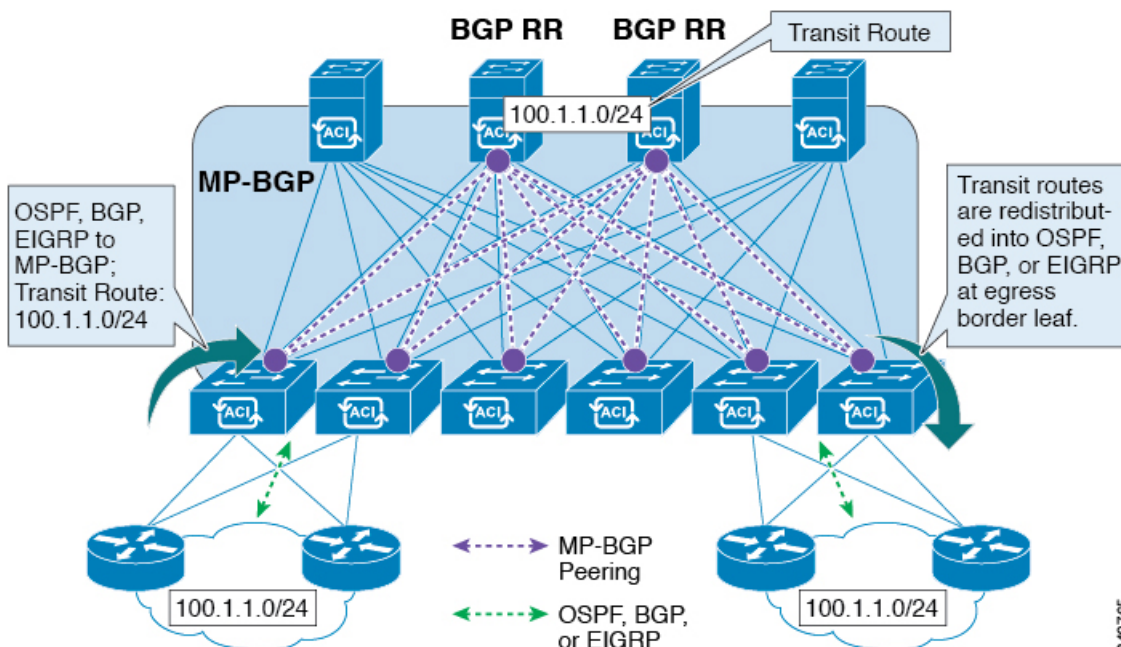
```

トランジット ルーティング

中継 ACI ファブリックのルーティング

Cisco APIC ソフトウェアは、OSPF（NSSA）およびiBGPを使用した外部レイヤ3接続をサポートします。ファブリックは、外部レイヤ3アウトサイド（l3out）接続の外部ルータにテナントブリッジドメインのサブネットをアドバタイズします。外部ルータから学習されたルートは、他の外部ルータにアドバタイズされません。ファブリックはスタブネットワークと同じように動作し、外部レイヤ3ドメイン間のトラフィックの伝送に使用できます。

図 25: ファブリックでルーティング中継



3-497/65

中継のルーティングで1つのテナントとVRF内の複数のL3Out接続がサポートされているし、APICは別のL3Out接続を1つのL3Out接続から学習したルートを実バタイズします。外部レイヤ3ドメインは、境界リーフスイッチのファブリックとピアリングします。ファブリックはピア間のMultiprotocol-Border Gateway Protocol (MP-BGP) 中継ドメインです。

外部L3Out接続の設定は、テナントとVRFレベルで実行されます。外部ピアから学習したルートは、VRFごとに入力リーフのMP-BGPにインポートされます。L3Out接続から学習したプレフィックスは、テナントVRFが存在するリーフスイッチにのみエクスポートされます。



(注) 注意事項と中継ルーティングの設定のガイドラインは、次を参照してください。 [中継ルーティングのガイドライン \(355 ページ\)](#)

トピックに関連するトランジットルーティング

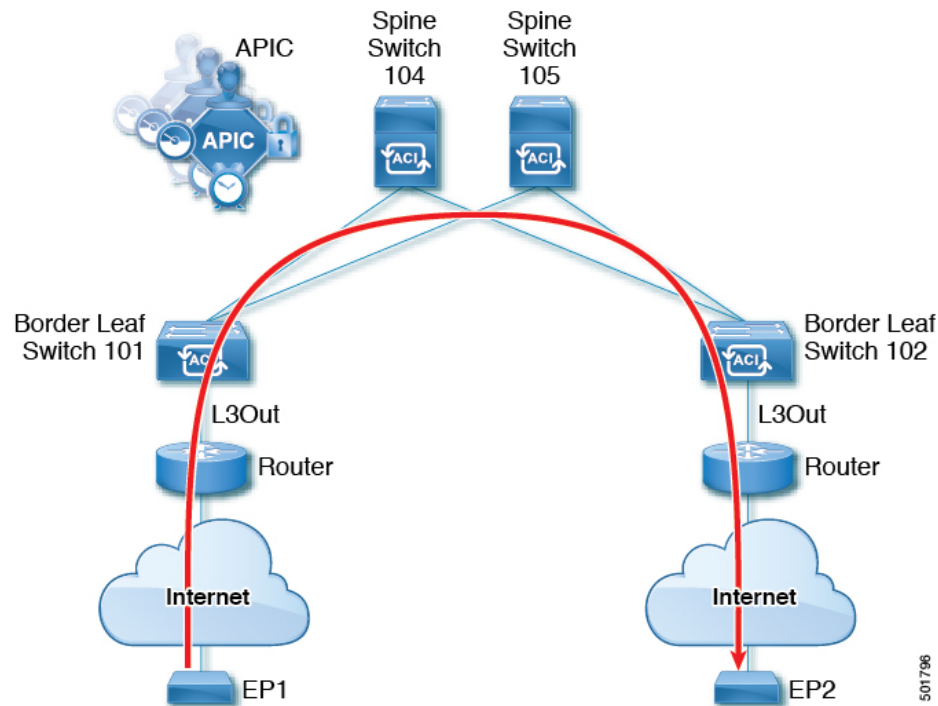
トランジットルーティングの概要、使用例、注意事項、トランジットルーティングのルート制御情報に関する詳細は、「Cisco APIC レイヤ3 ネットワーキング 設定ガイド」を参照してください。

トランジットルーティングの概要

このトピックでは、Cisco APIC を使用する際のトランジットルーティングを設定する方法の一般的な例を説明します。

この章にある例では、次のトポロジを使用します。

図 26:



この章の例では、Cisco ACI ファブリックには APIC クラスタによって制御される 2 個のリーフスイッチと 2 個のスパインスイッチがあります。境界リーフスイッチ 101 と 102 には L3Out があり、2 つのルータに接続することでインターネットにも接続しています。この例の目標は、2 つの L3Out を通して、インターネット上の EP1 から EP2 へ、ファブリック内外をトラフィックが行き来できるようにすることです。

この例では、両方の L3Out に関連付けられているテナントは、t1 と、VRF がつく v1 です。

L3Out を設定する前に、ノード、ポート、機能プロファイル、AEP、レイヤ 3 ドメインを設定します。BGP ルートリフレクタとして 104 と 105 スパインスイッチを設定する必要があります。

トランジットルーティングの設定には、次のコンポーネントの定義が含まれます。

1. テナントおよび VRF
2. リーフ 101 と 102 上のノードおよびインターフェイス
3. 各 L3Out のプライマリ ルーティングプロトコル（境界リーフスイッチと外部ルータ間のルートの交換に使用。この例では BGP）
4. 各 L3Out のルーティングプロトコルの接続性（プライマリ プロトコルへの到達可能性情報の提供。この例では、OSPF）
5. 2 個の外部 EPG
6. 1 個のルートマップ

7. 少なくとも 1 つのフィルタと 1 つのコントラクト
8. 外部 EPG とコントラクトを関連付ける



(注) トランジットルーティングの注意事項については、[中継ルーティングのガイドライン \(355 ページ\)](#) を参照してください。

次の表では、この章で使用される名前を一覧にしています。

プロパティ	ノード 101 の L3Out1 の名前	ノード 102 の L3Out2 の名前
テナント	t1	t1
VRF	v1	v1
ノード	ルータ ID 11.11.11.103 を持つ nodep1	ルータ ID 22.22.22.203 を持つ nodep2
OSPF インターフェイス	Eth/1/3 の ifp1	Eth/1/3 の ifp2
BPG ピア アドレス	15.15.15.2/24	25.25.25.2/24
外部 EPG	192.168.1.0/24 の extnw1	192.168.2.0/24 の extnw2
ルート マップ	Ctx1 を持つ rp1 とルートの宛 先 192.168.1.0/24	ctx2 を持つ rp2 とルートの宛 先 192.168.2.0/24
フィルタ	http-filter	http-filter
コントラクト	extnw1 によって提供される httpCtrect	extnw2 によって消費される httpCtrect

中継ルーティングのガイドライン

作成し、中継ルーティング接続を維持する場合は、次のガイドラインを使用します。

トピック	注意またはガイドライン
中継が 1 つの L3Out プロファイルを使用してルーティング	

トピック	注意またはガイドライン
	前に APIC 2.3(1f) のリリースでは、ルーティングを通過、1 つの L3Out プロファイル内ではサポートされていませんでした。APIC 2.3(1f) のリリース、および以降では、中継が単一 L3Out プロファイルで、次の制限でルーティングを設定できます。 • VRF が適用されるではない場合は、同じ L3EPG を共有ルータ間のトラフィックを許可する 0.0.0.0/0 の外部のサブネット (l3extSubnet) を使用できます。 • 外部デフォルト サブネット (0.0.0.0/0) の場合は、VRF を適用すると、 できません 同一のレイヤ 3 EPG 内のトラフィックの送信元と宛先の両方のプレフィックスに一致するために使用します。同一のレイヤ 3 EPG 内のすべてのトラフィックを一致するには、次のプレフィックスがサポートされています。 • IPv4 • 外部 EPG の 0.0.0.0/1—with 外部サブネット • 外部 EPG の 128.0.0.0/1—with 外部サブネット • 0.0.0.0/0—with インポート ルート制御サブネット、集約のインポート • IPv6 • 0::0/1: 外部 EPG の外部のサブネットを持つ • 8000::0/1: 外部 EPG の外部のサブネットを持つ • 0:0/0: インポート ルート制御のサブネットで集約のインポート • また、VzAny 契約と組み合わせると、1 つのデフォルトのサブネット (0.0.0.0/0) を使用できます。次に例を示します。 • 使用契約とレイヤ 3 EPG の提供 VzAny 消費契約 (一致する 0.0.0.0/0)、または、VzAny には、契約とレイヤ 3 EPG が提供される契約 (0.0.0.0/0 に一致する) が使用された行数。 • サブネット 0.0.0.0/0—with インポート/エクスポート ルート制御サブネット、集約のインポートおよび集約エクスポートを使用します。

トピック	注意またはガイドライン
ハードウェア サポートの違いを共有ルート:	第 2 世代のスイッチの VRF 機能間で正常にルートが共有されます (N9K-93108TC-EX など、スイッチモデル名の最後やその後に「EX」や「FX」がつく Cisco Nexus N9K)。第 1 世代のスイッチですが、ルートを保存する物理的な 3 進コンテンツ対応メモリ (TCAM) にルートの解析を完全にサポートするだけの容量がないため、この設定のパケットは失敗する可能性があります。
背面に戻る設定で EIGRP や OSPF	Cisco APIC では、中継が、L3Out に設定されているエクスポートルート制御ポリシーのルーティングをサポートします。ルート(プレフィックス)を通過するこれらのポリシー制御は、L3Out でルーティングプロトコルに再配送されます。これらの中継ルートは、EIGRP や OSPF に再配布されたが、これらはルーティングループを防ぐためにタグ付けされた 4294967295 です。Cisco ACI ファブリックは、OSPF または EIGRP L3Out で学習すると、このタグに一致するルートを受け入れません。ただし、次の場合、この動作をオーバーライドする必要があります。 • EIGRP や OSPF を使用して、2 つの Cisco ACI ファブリックを接続します。 • EIGRP や OSPF を使用して、同じ Cisco ACI ファブリックで 2 つの異なる Vrf を接続します。 APIC GUI の次の場所にある別のタグのポリシーを使用して、VRF を設定する必要がありますをオーバーライドする必要がある場所: テナント > <i>Tenant_name</i> > ネットワーキング > プロトコル ポリシー > ルート タグ。異なるタグを適用します。 新しいルート タグ ポリシーを作成するには、だけでなく、APIC GUI の次の場所でこのポリシーを使用する VRF を更新: テナント > <i>Tenant_name</i> > ネットワーキング > Vrf > <i>Tenant_VRF</i>。VRF を作成したルート タグ ポリシーを適用します。 (注) 複数 L3Outs または同じ L3Out で複数のインターフェイスは同じリーフスイッチに導入し、中継ルーティングに使用、(、IGP に再配布されません) IGP 内で、ルートをアドバタイズします。この状況では、ルート タグ ポリシーは適用されません。

トピック	注意またはガイドライン
BD サブネットをファブリック外にアドバタイズする	インポートおよびエクスポートのルート制御ポリシーは、中継ルート（他の外部ピアから学習したルート）およびスタティックルートのみに適用されます。テナント BD サブネット上に設定されているファブリック内部のサブネットは、エクスポート ポリシー サブネットを使用して外部にアドバタイズされません。IP プレフィックス リストおよびルート マップを使用すると IP テナント サブネットは許可されますが、これらは別の設定手順を使用して実装されます。テナント サブネットをファブリックの外部にアドバタイズする場合は、次の設定手順を参照してください。 1. [subnet properties] ウィンドウで、テナント サブネットのスコープを [Public Subnet] として設定します。 2. オプション。[subnet properties] ウィンドウで、[Subnet Control] を [ND RA Prefix] として設定します。 3. テナントブリッジドメイン (BD) を外部レイヤ 3 Outside に関連付けます (L3Out)。 4. テナント EPG と外部 EPG 間にコントラクト（プロバイダ/コンシューマ）の関連付けを作成します。 BD サブネットを Public 範囲に設定し、BD をレイヤ 3 Out に関連付けると、BD サブネットプレフィックスの境界リーフに IP プレフィックスおよびルートマップの連続エントリが作成されます。

トピック	注意またはガイドライン
デフォルト ルートのアドバタイズ	デフォルトルートのみを必要とするファブリックへの外部接続の場合、OSPF、EIGRP、および BGP の L3Out 接続をデフォルトルートの起点とすることがサポートされます。外部ピアからデフォルトルートが受信されると、この文書で説明されている中継エクスポートルート制御に従って、このルートを別のピアに再配布できます。 デフォルトルートは、デフォルトルート リーク ポリシーを使用してアドバタイズすることもできます。このポリシーは、デフォルトルートがルーティング テーブル内にあるか、または常にデフォルトルートをアドバタイズすることがサポートされている場合、デフォルトルートのアドバタイズをサポートします。デフォルトルート リーク ポリシーは、L3Out 接続で設定されます。 デフォルトルート リーク ポリシーを作成するときは、以下のガイドラインに従ってください: • BGP の場合、Always プロパティは適用されません。 • BGP の場合、Scope プロパティを設定するときには、Outside を選択します。 • OSPF の場合、範囲の値が Context だとタイプ 5 LSA が作成されるのに対し、Outside だとタイプ 7 LSA が作成されます。選択したは、L3Out で設定されたエリアのタイプによって異なります。エリアタイプが場合 定期的な、範囲を設定します コンテキスト。エリアタイプが場合 NSSA、範囲を設定します 外部。 • EIGRP で、Scope プロパティを選択する場合には、Context を選択する必要があります。

トピック	注意またはガイドライン
MTU	Cisco ACI は、IP フラグメンテーションをサポートしていません。したがって、外部ルータへのレイヤ 3 Outside (L3Out) 接続、または Inter-Pod Network (IPN) を介した multipod 接続を設定する場合は、MTU が両側で適切に設定されていることが重要です。ACI、Cisco NX-OS、Cisco IOS などの一部のプラットフォームでは、設定された MTU 値は IP ヘッダーを考慮に入れています（結果として、最大パケットサイズは、ACI で 9216 バイト、NX-OS および IOS で 9000 バイトに設定されます）。ただし、IOS XR などの他のプラットフォームは、パケットヘッダーのを除く MTU 値を設定します（結果として最大パケットサイズは 8986 バイトになります）。 各プラットフォームの適切な MTU 値については、それぞれの設定ガイドを参照してください。 CLI ベースのコマンドを使用して MTU をテストすることを強く推奨します。たとえば、Cisco NX-OS CLI で <code>ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1</code> などのコマンドを使用します。

NX-OS スタイル CLI を使用したトランジットルーティングの設定

次の手順では、テナントのトランジットルーティングを設定する方法を説明します。この例では、別々にルータに接続された 2 つの境界リーフ スイッチ上の 1 個の VRF で、2 個の L3Outs を展開します。

始める前に

- ノード、ポート、AEP、機能プロファイル、レイヤ 3 ドメインを設定します。
- 使用して VLAN ドメイン設定、`vlan ドメイン` ドメイン および `vlan vlan 範囲` コマンド。
- BGP ルート リフレクタ ポリシーを設定し、ファブリック内でルーテッドを伝達します。

これらの前提条件のコマンド例については、[NX-OS スタイル CLI の例: L3Out の前提条件 \(192 ページ\)](#) を参照してください。

手順

ステップ 1 テナントおよび VRF を設定します。

この例では VRF v1 でテナント t1 を設定します。VRF はまだ展開されていません。

例：

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
```

ステップ 2 ノードおよびインターフェイスを設定します。

この例では、2つの境界リーフスイッチでテナント t1 の2つの L3Outs を設定します。

- 最初の L3Out はノード 101 上にあり、nodep1 という名前です。ノード 101 はルータ ID 11.11.11.103 で設定されます。ルーテッドインターフェイス ifp1 が eth1/3 にあり、IP アドレス 12.12.12.3/24 です。
- 2 番目の L3Out が ノード 102 上にあり、nodep2 という名前です。ノード 102 はルータ ID 22.22.22.203 で設定されます。ルーテッドインターフェイス ifp2 が eth1/3 に存在し、IP アドレスは 23.23.23.1/24 です。

例：

```
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 11.11.11.103
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 12.12.12.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 22.22.22.203
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 23.23.23.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

ステップ 3 両方のリーフスイッチのルーティングプロトコルを設定します。

この例では、両方の境界リーフスイッチに対して、ASN 100 でプライマリ ルーティングプロトコルとして BGP を設定します。BGP ピア 15.15.15.2 を持つノード 101 と BGP ピア 25.25.25.2 を持つノード 102 を設定します。

例：

```
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit
```

```
apicl(config)# leaf 102
apicl(config-leaf)# router bgp 100
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 25.25.25.2
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# exit
```

ステップ 4 接続ルーティング プロトコルを設定します。

この例では、定期的なエリア ID 0.0.0.0 で両方の L3Outs に対して通信プロトコルとして OSPF を設定します。

例：

```
apicl(config)# leaf 101
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant t1 vrf v1
apicl(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 40.40.40.1
apicl(config-leaf-ospf-vrf)# exit
apicl(config-leaf-ospf)# exit
apicl(config-leaf)# exit
apicl(config)# leaf 102
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant t1 vrf v1
apicl(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 60.60.60.1
apicl(config-leaf-ospf-vrf)# exit
apicl(config-leaf-ospf)# exit
apicl(config-leaf)# exit
```

ステップ 5 外部 EPG を設定します。

この例では、ネットワーク 192.168.1.0/24 をノード 101 上の外部ネットワーク extnw1 として、ネットワーク 192.168.2.0/24 をノード 102 上の外部ネットワーク extnw2 として設定します。

例：

```
apicl(config)# tenant t1
apicl(config-tenant)# external-l3 epg extnw1
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# match ip 192.168.1.0/24
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# external-l3 epg extnw2
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# match ip 192.168.2.0/24
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# external-l3 epg extnw1
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# exit
apicl(config)# leaf 102
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# external-l3 epg extnw2
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# exit
```

ステップ 6 オプション。ルート マップを設定します。

この例では、インバウンドおよびアウトバウンド方向で各 BGP ピアのルートマップを設定します。

例：

例：

```

apic1(config)# leaf 101
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp2 out
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 102
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 25.25.25.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp2 in
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 out
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

```

ステップ 7 フィルタ（アクセスリスト）およびコントラクトを作成し、EPGが通信できるようにします。

例：

```

apicl(config)# tenant t1
apicl(config-tenant)# access-list http-filter
apicl(config-tenant-acl)# match ip
apicl(config-tenant-acl)# match tcp dest 80
apicl(config-tenant-acl)# exit
apicl(config-tenant)# contract httpCtrct
apicl(config-tenant-contract)# scope vrf
apicl(config-tenant-contract)# subject subj1
apicl(config-tenant-contract-subj)# access-group http-filter both
apicl(config-tenant-contract-subj)# exit
apicl(config-tenant-contract)# exit
apicl(config-tenant)# exit

```

ステップ 8 コントラクトを設定し、EPG に関連付けます。

例：

```

apicl(config)# tenant t1
apicl(config-tenant)# external-l3 epg extnw1
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# contract provider httpCtrct
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# external-l3 epg extnw2
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# contract consumer httpCtrct
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# exit
apicl(config)#

```

例：中継ルーティング

この例では、中継ルーティングのマージされた設定を提供します。設定は別々のルータに接続されている 2 個の障壁リーフ スイッチで、2 つの L3Outs を持つ単一のテナントと VRF のためにあります。

```

apicl# configure
apicl(config)# tenant t1
apicl(config-tenant)# vrf context v1
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# exit

apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# router-id 11.11.11.103
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# interface ethernet 1/3
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vrf member tenant t1 vrf v1
apicl(config-leaf-if)# ip address 12.12.12.3/24
apicl(config-leaf-if)# exit
apicl(config-leaf)# router bgp 100
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant t1 vrf v1

```

例：中継ルーティング

```

apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 40.40.40.1
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 22.22.22.203
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 23.23.23.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 25.25.25.2/24
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 60.60.60.3
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit

apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 192.168.1.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant-l3ext-epg)# external-l3 epg extnw2
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit

apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw1
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw2
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 101
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0

```

```
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# router bgp 100
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp2 out
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# exit

apicl(config)# leaf 102
apicl(config-leaf)# template route group match-rule1 tenant t1
apicl(config-route-group)# ip prefix permit 192.168.1.0/24
apicl(config-route-group)# exit
apicl(config-leaf)# template route group match-rule2 tenant t1
apicl(config-route-group)# ip prefix permit 192.168.2.0/24
apicl(config-route-group)# exit
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# route-map rp1
apicl(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# route-map rp2
apicl(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# router bgp 100
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 25.25.25.2
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp2 in
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp1 out
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# exit

apicl(config)# tenant t1
apicl(config-tenant)# access-list http-filter
apicl(config-tenant-acl)# match ip
apicl(config-tenant-acl)# match tcp dest 80
apicl(config-tenant-acl)# exit
apicl(config-tenant)# contract httpCtrct
apicl(config-tenant-contract)# scope vrf
apicl(config-tenant-contract)# subject http-subj
apicl(config-tenant-contract-subj)# access-group http-filter both
apicl(config-tenant-contract-subj)# exit
apicl(config-tenant-contract)# exit
apicl(config-tenant)# exit

apicl(config)# tenant t1
apicl(config-tenant)# external-l3 epg extnw1
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# contract provider httpCtrct
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# external-l3 epg extnw2
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# contract consumer httpCtrct
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# exit
apicl(config)#
```

例：中継ルーティング



第 8 章

Cisco ACI QoS の設定

この章の内容は、次のとおりです。

- [L3Outs の QoS \(369 ページ\)](#)
- [CoS の保持 \(370 ページ\)](#)
- [Multipod QoS \(372 ページ\)](#)
- [出力マーキングを QoS 入力マーキングの変換 \(374 ページ\)](#)

L3Outs の QoS

L3Out の QoS ポリシーを設定するには、次のガイドラインを使用します。

- L3Out がある障壁リーフに適用する QoS ポリシーを設定するには、VRF が出力モード（ポリシー制御適用方向は「出力」にする必要があります）である必要があります。
- 適用する QoS ポリシーを有効にするには、VRF ポリシー制御適用設定を「適用」にする必要があります。
- L3Out とその他の EPG 間の通信を制御する契約を設定する際に、契約またはサブジェクトの QoS クラスまたはターゲット DSCP を含めます。



(注) 外部 EPG ではなく、契約の QoS クラスまたはターゲット DSCP のみ設定します（`l3extInstP`）。

NX-OS スタイルの CLI を使用した L3Outs の QoS の設定

L3Out の QoS は、L3Out 設定の一部として設定されます。

手順

ステップ 1 L3Out で QoS 優先順位の適用をサポートするには、テナントと、VRF を設定するときに出力モードの VRF を設定し、次のコマンドを使用して、ポリシーの適用を有効に。

例：

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# contract enforce egress
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)#
```

ステップ 2 フィルタ (アクセス リスト) を作成するときを含める、**dscp を一致** コマンドで、ターゲット DSCP では、この例ではレベル EF。契約を設定するときなど、QoS クラスを含める レベル 1、L3Out でトラフィック ingressing の。または、ターゲットの DSCP 値を定義する可能性があります。QoS ポリシーは、契約またはサブジェクトのいずれかでサポートされます。

例：

```
apic1(config)# tenant t1
apic1(config-tenant)# access-list http-filter
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# match dscp EF
apic1(config-tenant-acl)# exit
apic1(config-tenant)# contract httpCtct
apic1(config-tenant-contract)# scope vrf
apic1(config-tenant-contract)# qos-class level1
apic1(config-tenant-contract)# subject http-subject
apic1(config-tenant-contract-subj)# access-group http-filter both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
```

CoS の保持

802.1 P クラスのサービスの設定の保持

APIC では、ファブリック内で 802.1 P クラスのサービス (CoS) 設定を保持することができます。ファブリック グローバル QoS ポリシーの dot1p-preserve オプションを有効にすることで、ACI ファブリックを入力し中継するパケット CoS 値の保持が保証されます。

802.1 P CoS の保持は単一のポッドおよび multipod トポロジでサポートされます。

Multipod トポロジは、CoS の保持を使用できますポッド 1 を入力して、ポッド 2 外からの 802.1 P トラフィックの優先順位の QoS の設定を保持したいですが、CoS の保持を行わない interpod の DSCP 設定のネットワーク (IPN) トラフィックポッド間。マルチポッドトラフィックが IPN に送信される際に CoS/DSCP 設定を保持するには、DSCP ポリシーを使用します。詳細について

では、マルチポッドファブリックでの QoS 優先度の設定項目の保持 (373 ページ) を参照してください。

次に示す 801.1P CoS 保持のガイドラインおよび制限事項に従ってください。

- 現在のリリースでは、VLAN ヘッダー内の 801.1P 値のみ保持可能です。DEI ビットは保持されません。
- VXLAN カプセル化パケットについては、現在のリリースでは外部ヘッダに含まれる CoS 値を保持しません。
- 次の設定オプションを有効にすると、802.1P の保持は行われません。
 - マルチポッド QoS (DSCP ポリシーを使用) を有効にします。
 - コントラクトが QoS を含むように設定されます。
 - ダイナミック パケットのプライオリティが有効。
 - 発信インターフェイスが FEX 上にある。
- トラフィックが分離が適用されている EPG から分離を適用していない EPG に転送される場合、QoS CoS 優先順位設定の保持はサポートされません。
- DSCP QoS ポリシーが VLAN EPG 上で設定され、パケットに IP ヘッダがあります。次の項目について、最内部～最外部の優先順位を付けて、DSCP マーキングをフィルタ レベルで設定できます。
 - コントラクト
 - サブジェクト
 - インターム
 - アウトターム



(注) コントラクトに vzAny を指定した場合、vzAny はその VRF のすべての EPG の集合であり、EPG の固有の設定が適用できないため、外部 EPG DSCP の値は考慮されません。EPG 固有のターゲット DSCP 値が必要な場合は、外部 EPG で vzAny を使用しないでください。

NX-OS スタイル CLI を使用した QoS CoS 設定の保持

QoS 優先順位設定がトラフィックの受信および単一ポッドファブリックの送信、または1個のポッドによるトラフィックの受信およびマルチポッドファブリック内の別の場所への出力と同じように処理されるには、次の手順のコマンドを使用して CoS 保持を有効にします。



- (注) CoS 保持の有効化により、CoS 優先順位のデフォルト マッピングが、DSCP レベルからさまざまな種類のトラフィックに適用されます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	qos preserve cos 例 : apic1(config)# qos preserve cos	CoS 保持を有効にしています。

Multipod QoS

NX-OS スタイル CLI を使用した DSCP ポリシーによるマルチポッド QoS の有効化

DSCP マップを作成し（APIC GUI で DSCP ポリシーと呼ばれる）、マルチポッド トポロジで QoS 優先順位設定を保証します。マッピングはポリシー内で固有である必要があります。

次の手順を使用して、トラフィック ストリームにカスタム マッピングで DSCP マップを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	tenant <i>tenant-name</i> apic1(config)# tenant infra	インフラ テナントのテナント設定モードを開始します。
ステップ 3	qos dscp-map <i>dscp-translation-policy-name</i> apic1(config-tenant)# qos dscp-map default	DSCP マップを設定します。

	コマンドまたはアクション	目的
ステップ 4	set dscp-code	次の例のように、カスタム DSCP マッピングを設定します。マッピングは DSCP マップ内で固有である必要があります。 (注) IPN を通過するトラフィックは、DSCP 値を COS6 (traceroute トラフィック) にマップしないください。 <pre> apic1(config-qos-cmap# set dscp-code control CS3 apic1(config-qos-cmap# set dscp-code span CS5 apic1(config-qos-cmap# set dscp-code level1 CS0 apic1(config-qos-cmap# set dscp-code level2 CS1 apic1(config-qos-cmap# set dscp-code level3 CS2 apic1(config-qos-cmap# set dscp-code policy CS4 apic1(config-qos-cmap# set dscp-code traceroute CS6 </pre>
ステップ 5	no shutdown	DSCP マップを有効にします。 <pre> apic1(config-qos-cmap) # no shutdown </pre>

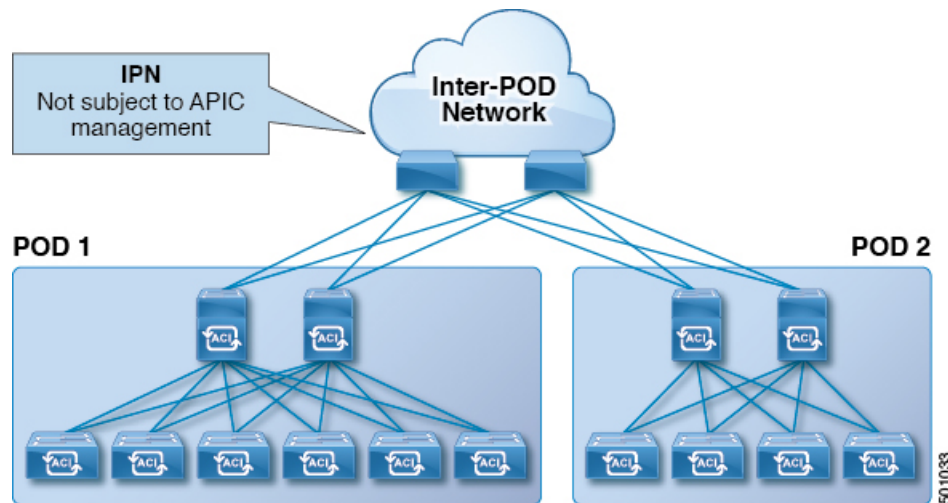
マルチポッド ファブリックでの QoS 優先度の設定項目の保持

このトピックでは、マルチポッド トポロジの QoS 優先度の設定項目を保証する方法について説明します。ポッド内ネットワークのデバイスが APIC 管理に存在しておらず、ネットワークを送信するトラフィックの 802.1P 設定を変更できます。



- (注) ポッド 1 に入りポッド 2 のエングレスを退出する 802.1P トラフィックの QoS 優先度設定を保持しますが、ポッド間でポッド内ネットワーク (IPN) トラフィックの CoS/DSCP 設定を保持しない場合、代わりに CoS 保持を使用できます。詳細については、[802.1P クラスのサービスの設定の保持 \(370 ページ\)](#) を参照してください。

図 27: マルチポッド トポロジ



この図に示すように、マルチポッド トポロジのポッド間のトラフィックは、APIC 管理の下にない可能性がある IPN を通過します。802.1P フレームがポッド 1 のスパインまたはリーフ スイッチから送信されるとき、IPN 内のデバイスでは 802.1P フレームの CoS を保持することはできませんこの状況では、フレームがポッド 2 スパインまたはリーフ スイッチに達すると、ポッド 1 のソースに割り当てられたレベルの代わりに IPN デバイスにより割り当てられた CoS レベルがあります QoS の優先度レベルがこの状況で保持されることを確認するには、DSCP ポリシーを使用します。

マルチポッド トポロジで QoS の優先度設定を保持する DSCP ポリシーを設定し、異なるトラフィック タイプの CoS から DSCP レベルへのマッピングを確定する必要があります。IPN 内のデバイスで設定されているレベルを変更されることを防ぎます。DSCP ポリシーが有効になっている場合、APIC は CoS レベルを設定するマッピングに対応する DSCP レベルに変換します。ポッド 1 からフレームが送信されるとき (DSCP レベルにマッピングされている PCP レベルとともに)、そしてポッド 2 に到達するとき、マッピングされた DSCP レベルは元の PCP CoS レベルに戻ります。

出力マーキングを QoS 入力マーキングの変換

QoS 入力マーキングを出力マーキングに変換する

APIC では、入力 DSCP 値に基づき 802.1P CoS フィールド (サービスのクラス) に変換できます。802.1P CoS 変換は、IP パケットに DSCP が存在し、dot1P がイーサネット フレーム内に存在する場合のみサポートされます。

この機能では、ACI ファブリックがデバイスのトラフィックを分類し、CoS 値のみに基づいてトラフィックを分類できます。これは、入力 dot1P 値に基づいて CoS 値が dot1P をマッピングできるようになります。これは主に、IP ヘッダーを持たないレイヤ 2 のパケットに適用されます。

次の 802.1P CoS 変換の注意事項と制約事項に注意してください。

- ファブリック グローバル QoS ポリシー `dot1p-preserve` オプションを有効にします。
- 802.1 P CoS の変換は、外部の L3 インターフェイスではサポートされていません。
- 802.1 P CoS の変換は、出力フレームが 802.1 q カプセル化された場合にのみサポートされています。

802.1P CoS 変換は、次の設定オプションが有効になっている場合サポートされません。

- 契約は、QoS を含み設定されます。
- 発信インターフェイスが FEX 上にある。
- DSCP ポリシーを使用してマルチポッド QoS を有効にします。
- ダイナミック パケットのプライオリティが有効。
- 内部 EPG エンドポイント分離が実行された状態で EPG が設定される場合。
- EPG がマイクロセグメンテーションの許可が有効になった状態で設定される場合。

NX-OS は、CLI を使用した QoS CoS 設定の変換

次のコマンドを使用してカスタム QoS ポリシーを作成し、EPG とポリシーを関連付けます。

始める前に

カスタム QoS ポリシーで消費されるテナント、アプリケーション、EPG を作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	<code>configure</code> 例 : <code>apicl#configure</code>	グローバル コンフィギュレーション モードを開始します。 (注) 手順 1-5 に記載してあるコマンドを入力し、カスタム QoS ポリシーを作成します。
ステップ 2	<code>tenant tenant-name</code> 例 : <code>apicl(config)#tenant t001</code>	テナントのテナント設定モードを入力します。
ステップ 3	<code>policy-map type qos QoS-policy-name</code> 例 : <code>apicl(config-tenant)#policy-map type qos baz</code>	QoS ポリシーを作成します。

	コマンドまたはアクション	目的
ステップ 4	match dscp AF23 AF31 set-cos 6 例 : <pre>apic1(config-tenant-pmap-qos)#match dscp AF23 AF31 set-cos set-cos 6</pre>	DCSP 値とターゲット QoS 値を設定します。
ステップ 5	exit 例 : <pre>apic1(config-tenant-pmap-qos)#exit</pre>	テナントコンフィギュレーションモードに戻ります。
ステップ 6	application app-name 例 : <pre>apic1(config-tenant)#application ap2</pre>	アプリケーションプロファイルを作成します。 (注) 手順 6-9 に記載されているコマンドを入力し、EPG とカスタム QoS ポリシーを関連付けます。
ステップ 7	epg epg-name 例 : <pre>apic1(config-tenant-app)# epg ep2</pre>	アプリケーションプロファイルに EPG を作成します。
ステップ 8	service-policy policy-name 例 : <pre>apic1(config-tenant-app-epg)#service-policy baz</pre>	EPG をポリシーに関連付けます。
ステップ 9	exit 例 : <pre>apic1(config-tenant-app-epg)#exit</pre>	テナントコンフィギュレーションモードに戻ります。
ステップ 10	external-l2epg epg-name 例 : <pre>apic1(config-tenant)#external-l2 epg myout:12</pre>	外部レイヤ 2 EPG を作成します。 (注) 手順 10-12- に記載されているコマンドを入力し、外部 L2 EPG とカスタム QoS ポリシーを関連付けます。
ステップ 11	service-policy policy-name 例 : <pre>apic1(config-tenant-l2ext-epg)#service-policy baz</pre>	EPG をポリシーに関連付けます。
ステップ 12	exit	テナントコンフィギュレーションモードに戻ります。



第 9 章

管理インターフェイスの設定

- ・ [アウトオブバンド管理アクセスの設定（377 ページ）](#)
- ・ [インバンド管理アクセスの設定（380 ページ）](#)

アウトオブバンド管理アクセスの設定

コントローラ、リーフ スイッチ、またはスパイン スイッチのアウトオブバンド（OOB）管理アクセスを設定するには、次の手順を実行する必要があります。

- ・ 管理インターフェイスで OOB 管理 IP アドレスおよびゲートウェイを設定します
- ・ 必要な外部サブネットからのアクセスを許可します
- ・ 管理ポートで必要なプロトコルを許可します

始める前に

APIC アウトオブバンド管理接続のリンクは、1 Gbps である必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apic1# configure	コンフィギュレーションモードに入ります。
ステップ 2	{controller apic-number-or-range switch node-id[-node-id-or-range]} 例： apic1(config)# controller 1-3	設定するコントローラまたはスイッチを指定します。ハイフンまたはカンマを使用してコントローラまたはスイッチの範囲を入力できます。

	コマンドまたはアクション	目的
ステップ 3	interface mgmt0 例 : <pre>apic1(config-controller)# interface mgmt0</pre>	mgmt0 インターフェイスではアウトオブバンド管理を行え、IPv4 アドレスでデバイスを管理できます。
ステップ 4	ip address addr/mask gateway addr 例 : <pre>apic1(config-controller-if)# ip address-range 172.23.48.16/21 gateway 172.23.48.1</pre>	OOB 管理の IP アドレスおよびゲートウェイを設定します。If you specified more than one controller or switch, the command becomes ip address-range and IP addresses are assigned sequentially beginning with the address specified in this command. (注) APIC 管理インターフェイスは IPv6 アドレスをサポートしておらず、このインターフェイスを介して外部 IPv6 サーバに接続することはできません。
ステップ 5	exit 例 : <pre>apic1(config-controller-if)# exit</pre>	
ステップ 6	exit 例 : <pre>apic1(config-controller)# exit</pre>	
ステップ 7	tenant mgmt 例 : <pre>apic1(config)# tenant mgmt</pre>	System Management ポリシーは、 mgmt と呼ばれる特別なテナント下に設定されます。
ステップ 8	external-l3 epɡ default oob-mgmt 例 : <pre>apic1(config-tenant)# external-l3 epɡ default oob-mgmt</pre>	アウトオブバンド管理 EPG のコンフィギュレーションモードを開始します。
ステップ 9	match ip addr/mask 例 : <pre>apic1(config-tenant-l3ext-epɡ)# match ip 192.0.20.0/24</pre>	外部管理サブネットに対するアウトオブバンド管理インターフェイスのアクセス制御を提供します。
ステップ 10	exit 例 : <pre>apic1(config-tenant-l3ext-epɡ)# exit</pre>	

	コマンドまたはアクション	目的
ステップ 11	access-list oob-default 例 : <pre>apicl(config-tenant)# access-list oob-default</pre>	OOB デフォルト ポリシーのアクセス リスト フィルタを設定します。
ステップ 12	match tcp dest 443 例 : <pre>apicl(config-tenant-acl)# match tcp dest 443</pre>	HTTPS トラフィック (TCP/443) の管理インターフェイスへのアクセスを許可します。
ステップ 13	match tcp dest 22 例 : <pre>apicl(config-tenant-acl)# match tcp dest 22</pre>	SSH トラフィック (TCP/22) の管理インターフェイスへのアクセスを許可します。

例

次に、3つのAPICコントローラにアウトオブバンド管理アクセスを設定する例を示します。この例では、3つのコントローラに一連のIPアドレスが割り当てられ、コントローラ1には172.23.48.16/21、コントローラ2には172.23.48.17/21、コントローラ3には172.23.48.18/21が割り当てられます。

```
apicl# configure
apicl(config)# controller 1-3
apicl(config-controller)# interface mgmt0
apicl(config-controller-if)# ip address-range 172.23.48.16/21 gateway 172.23.48.1
apicl(config-controller-if)# exit
apicl(config-controller)# exit
apicl(config)# tenant mgmt
apicl(config-tenant)# external-l3 epg default oob-mgmt
apicl(config-tenant-l3ext-epg)# match ip 192.0.20.0/24
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# access-list oob-default
apicl(config-tenant-acl)# match tcp dest 443
apicl(config-tenant-acl)# match tcp dest 22
```

次に、リーフまたはスパインスイッチにアウトオブバンド管理アクセスを設定する例を示します。

```
apicl# configure
apicl(config)# switch 101
apicl(config-switch)# interface mgmt0
apicl(config-switch-if)# ip address 172.23.48.101/21 gateway 172.23.48.1
```

インバンド管理アクセスの設定

外部ネットワークからスイッチへのインバンド管理アクセスの設定

リーフ スイッチまたはスパイン スイッチのインバンド (IB) 管理アクセスを設定するには、次の手順を実行する必要があります。

- インバンド管理インターフェイスでインバンド管理 IP アドレスおよびゲートウェイを設定します
- 外部インバンド接続用の VLAN ドメインを作成または指定します
- VLAN ドメインに外部管理ステーション インターフェイスを追加します
- 管理ポートで必要なプロトコルを許可します

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	コンフィギュレーション モードに入ります。
ステップ 2	switch switch-id-or-range 例 : apic1(config)# switch 101	設定するスイッチを指定します。ハイフンまたはカンマを使用してスイッチの範囲を入力できます。
ステップ 3	interface inband-mgmt0 例 : apic1(config-switch)# interface inband-mgmt0	inband-mgmt0 インターフェイスはインバンド管理を提供します。
ステップ 4	ip address addr/mask gateway addr 例 : apic1(config-switch-if)# ip address 10.13.1.1/24 gateway 10.13.1.254	インバンド管理の IP アドレスおよびゲートウェイを設定します。If you specified more than one switch, the command becomes ip address-range and IP addresses are assigned sequentially beginning with the address specified in this command.
ステップ 5	exit 例 : apic1(config-switch-if)# exit	
ステップ 6	exit 例 :	

	コマンドまたはアクション	目的
	apicl(config-switch)# exit	

例

次に、外部ネットワークの管理ステーションからスイッチのインバンド管理を設定する例を示します。

```
apicl# configure
apicl(config)# switch 101
apicl(config-switch)# interface inband-mgmt0
apicl(config-switch-if)# ip address 10.13.1.1/24 gateway 10.13.1.254
apicl(config-switch-if)# exit
apicl(config-switch)# exit
```

次のタスク

- 管理ステーションへのインバンド（IB）管理の接続を設定します。
- インバウンド管理ポート上で必要なプロトコル（HTTPS および SSH）を許可します。

外部ネットワークからコントローラへのインバンド管理アクセスの設定

コントローラのインバンド（IB）管理アクセスを設定するには、次の手順を実行する必要があります。

- インバンド管理インターフェイスでインバンド管理 IP アドレスおよびゲートウェイを設定します
- 外部インバンド接続の VLAN ドメインを作成します
- コントローラに接続されているポートで VLAN を許可します

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apicl# configure	コンフィギュレーションモードに入ります。
ステップ 2	controller controller-id-or-range 例： apicl(config)# controller 1-3	設定するコントローラを指定します。 ハイフンまたはカンマを使用してコントローラの範囲を入力できます。

	コマンドまたはアクション	目的
ステップ 3	interface inband-mgmt0 例 : <pre>apic1(config-controller)# interface inband-mgmt0</pre>	inband-mgmt0 インターフェイスはインバンド管理を提供します。
ステップ 4	ip address addr/mask gateway addr 例 : <pre>apic1(config-controller-if)# ip address-range 10.13.1.1/24 gateway 10.13.1.254</pre>	インバンド管理の IP アドレスおよびゲートウェイを設定します。If you specified more than one controller or switch, the command becomes ip address-range and IP addresses are assigned sequentially beginning with the address specified in this command.
ステップ 5	vlan vlan-id 例 : <pre>apic1(config-controller-if)# vlan 10</pre>	コントローラに接続されたポートでイネーブルになっているコントローラ VLAN を割り当てます。複数のコントローラでは、すべてのコントローラで同じ VLAN を使用する必要があります。
ステップ 6	exit 例 : <pre>apic1(config-controller-if)# exit</pre>	
ステップ 7	exit 例 : <pre>apic1(config-controller)# exit</pre>	
ステップ 8	vlan-domain domain-name 例 : <pre>apic1(config)# vlan-domain apic-inband</pre>	VLAN ドメインのコンフィギュレーション モードを作成および開始します。
ステップ 9	vlan vlan-id 例 : <pre>apic1(config-vlan)# vlan 10</pre>	コントローラ VLAN を VLAN ドメインに割り当てます。
ステップ 10	exit 例 : <pre>apic1(config-vlan)# exit</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 11	leaf node-id 例 : <pre>apic1(config)# leaf 102</pre>	コントローラが接続したリーフスイッチを指定します。

	コマンドまたはアクション	目的
ステップ 12	interface slot/port 例 : apic1(config-leaf) # interface eth 1/1	コントローラを接続したポートを指定します。
ステップ 13	vlan-domain member apic-inband 例 : apic1(config-leaf-if) # vlan-domain member apic-inband	インバンド管理へのコントローラの接続を設定します。
ステップ 14	exit 例 : apic1(config-leaf-if) # exit	
ステップ 15	exit 例 : apic1(config-leaf) # exit	

例

次に、外部ネットワークの管理ステーションからコントローラのインバンド管理を設定する例を示します。APIC コントローラ 1 はリーフ 101 のポート イーサネット 1/1 に接続され、VLAN 10 はコントローラのインバンド接続に使用されます。

```
apic1# configure
apic1(config)# controller 1-3
apic1(config-controller)# interface inband-mgmt0
apic1(config-controller-if)# ip address-range 10.13.1.1/24 gateway 10.13.1.254
apic1(config-controller-if)# vlan 10
apic1(config-controller-if)# exit
apic1(config-controller)# exit

# CREATE A VLAN DOMAIN FOR THE APIC INBAND VLAN
apic1(config)# vlan-domain apic-inband
apic1(config-vlan)# vlan 10
apic1(config-vlan)# exit

# ALLOW THE VLAN ON THE PORT CONNECTED TO THE CONTROLLER
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/1
apic1(config-leaf-if)# vlan-domain member apic-inband
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

次のタスク

- 管理ステーションへのインバンド（IB）管理の接続を設定します。

- インバンド管理ポート上で必要なプロトコル（HTTPS および SSH）を許可します。

管理ステーションへのインバンド管理接続の設定

管理ステーションへのインバンド管理（IB）接続の設定をするには、次の手順を実行する必要があります。

- 外部インバンド接続用の VLAN ドメインを作成または指定します
- VLAN ドメインに外部管理ステーション インターフェイスを追加します

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： <code>apic1# configure</code>	コンフィギュレーションモードに入ります。
ステップ 2	vlan-domain domain-name 例： <code>apic1(config)# vlan-domain external-inband</code>	VLAN ドメインのコンフィギュレーション モードを作成および開始します。
ステップ 3	vlan vlan-id 例： <code>apic1(config-vlan)# vlan 11</code>	VLAN をドメインに割り当てます。
ステップ 4	exit 例： <code>apic1(config-vlan)# exit</code>	グローバル コンフィギュレーション モードに戻ります。
ステップ 5	leaf node-id 例： <code>apic1(config)# leaf 102</code>	管理ステーションが接続されているリーフ スイッチを指定します。
ステップ 6	interface slot/port 例： <code>apic1(config-leaf)# interface eth 1/2</code>	管理ステーションが接続されているポートを指定します。
ステップ 7	vlan-domain member external-inband 例： <code>apic1(config-leaf-if)# vlan-domain member external-inband</code>	インバンド管理に外部レイヤ 2 接続を設定します。

	コマンドまたはアクション	目的
ステップ 8	switchport trunk allowed vlan <i>vlan-id</i> inband-mgmt <i>gateway-ip/mask</i> 例 : apicl(config-leaf-if)# switchport trunk allowed vlan 11 inband-mgmt 179.10.1.254/24	インバンド管理に外部レイヤ 2 接続を設定します。指定した IP アドレスは、外部管理ステーションで使用するゲートウェイ アドレスです。また、ゲートウェイ機能は ACI ファブリックで提供されます。
ステップ 9	exit 例 : apicl(config-leaf-if)# exit	
ステップ 10	exit 例 : apicl(config-leaf)# exit	

例

この例では、管理ステーションへのインバンド管理接続を設定する方法を示します。

```
# CREATE A VLAN DOMAIN FOR EXTERNAL CONNECTIVITY TO INBAND MANAGEMENT
apicl# configure
apicl(config)# vlan-domain external-inband
apicl(config-vlan)# vlan 11
apicl(config-vlan)# exit

# CONFIGURE LAYER 2 CONNECTIVITY FROM THE MANAGEMENT STATION INTERFACE TO INBAND MANAGEMENT
apicl(config)# leaf 102
apicl(config-leaf)# interface eth 1/2
apicl(config-leaf-if)# vlan-domain member external-inband
apicl(config-leaf-if)# switchport trunk allowed vlan 11 inband-mgmt 179.10.1.254/24
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
```

次のタスク

- ・インバンド管理ポート上で必要なプロトコル（HTTPS および SSH）を許可します。

HTTPS/SSH ポートを開くインバンド管理コントラクトの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	コンフィギュレーション モードに入ります。
ステップ 2	tenant mgmt 例 : apicl (config)# tenant mgmt	System Management ポリシーは、 mgmt と呼ばれる特別なテナント下に設定されます。
ステップ 3	access-list inband-default 例 : apicl (config-tenant)# access-list inband-default	インバンドデフォルト ポリシーのアクセス リスト フィルタを設定します。
ステップ 4	match tcp dest 443 例 : apicl (config-tenant-acl)# match tcp dest 443	HTTPS トラフィック (TCP/443) の管理インターフェイスへのアクセスを許可します。
ステップ 5	match tcp dest 22 例 : apicl (config-tenant-acl)# match tcp dest 22	SSH トラフィック (TCP/22) の管理インターフェイスへのアクセスを許可します。

例

次に、インバンド管理ポートへの HTTPS および SSH アクセスを許可する例を示します。

```

apicl# configure
apicl (config)# tenant mgmt
apicl (config-tenant)# access-list inband-default
apicl (config-tenant-acl)# match tcp dest 443
apicl (config-tenant-acl)# match tcp dest 22
apicl (config-tenant-acl)# exit
apicl (config-tenant)# exit

```



第 10 章

セキュリティの設定

- セキュリティの設定について (387 ページ)
- AAA の設定 (388 ページ)
- セキュリティ サーバの設定 (392 ページ)
- パスワード ポリシーの設定 (399 ページ)
- ユーザの設定 (402 ページ)
- 公開キー インフラストラクチャの設定 (406 ページ)
- 通信ポリシーの設定 (412 ページ)
- AES 暗号化の設定 (417 ページ)
- ファブリック セキュア モードの設定 (418 ページ)
- COOP 認証の設定 (419 ページ)
- FIPS の設定 (421 ページ)
- コントロール プレーン ポリシングの設定 (422 ページ)
- ファースト ホップ セキュリティの設定 (427 ページ)
- 802.1X の設定 (435 ページ)

セキュリティの設定について

アクセス コントロールは、ネットワーク サーバへのアクセスを許可するユーザと、アクセスしたユーザに使用を許可するサービスを制御する手段です。認証、許可、アカウントिंग (AAA) ネットワーク セキュリティ サービスは、APIC 上でアクセス コントロールを設定するための基本的なフレームワークを提供します。

AAA 設定の概要

AAA を使用した APIC のセキュリティを設定するには、このプロセスの通りにします。

1. To use a separate security server, configure security protocol parameters using the **radius-server**, **ldap-server**, or **tacacs-server** configuration commands.
2. Define the method lists for authentication by using an **aaa authentication** command.
3. 必要に応じて、方式リストを特定のインターフェイスまたは回線に適用します。

4. (Optional) Configure authorization using the **aaa authentication** command.

ローカルパスワードによるログイン認証

Use the **aaa authentication login** command with the *method* argument to specify that APIC will use the local username database for authentication. たとえば、他の方式リストが定義されていない場合にログイン時のユーザ認証方式としてローカルユーザ名データベースを指定するには、次のコマンドを入力します。

```
apic1# configure
apic1(config)# aaa authentication login default
apic1(config-default)# realm local
```

ローカルのユーザ名データベースにユーザを追加する方法の詳細については、「ローカルで認証されたユーザの設定」の項を参照してください。

リモートサーバによるログイン認証

ログイン認証方式として RADIUS または TACACS+ または LDAP を指定するには、**server radius** 方式または **server tacacs** 方式または **server ldap** 方式を指定して **aaa authentication login** コマンドを使用します。たとえば、他の方式リストが定義されていない場合にログイン時のユーザ認証方式として RADIUS を指定するには、次のコマンドを入力します。

```
apic1# configure
apic1(config)# aaa authentication login default
apic1(config-default)# realm radius
```

ログイン認証方式として RADIUS を使用するには、RADIUS セキュリティ サーバとの通信をイネーブルにしておく必要があります。TACACS+ または LDAP も同様です。リモートセキュリティ サーバとの通信の確立に関する詳細については、該当する章を参照してください。

- 「RADIUS サーバの設定」
- 「TACACS+ サーバの設定」
- 「LDAP サーバの設定」

AAA の設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	aaa authentication login console 例 : apicl(config)# aaa authentication login console	ユーザがコンソールから APIC にアクセスする際のコンソールコンフィギュレーションモードを開始します。
ステップ 3	[no] realm {ldap local radius tacacs} 例 : apicl(config-console)# realm radius	認証方式を指定します。
ステップ 4	[no] group group-name 例 : apicl(config-console)# group radiusGroup5	認証サーバ グループを指定します。
ステップ 5	exit 例 : apicl(config-console)# exit	グローバル コンフィギュレーションモードに戻ります。
ステップ 6	aaa authentication login default 例 : apicl(config)# aaa authentication login default	デフォルトのログイン認証の設定モードを開始します。
ステップ 7	[no] realm {ldap local radius tacacs} 例 : apicl(config-default)# realm radius	認証方式を指定します。
ステップ 8	[no] group group-name 例 : apicl(config-default)# group radiusGroup	認証サーバ グループを指定します。
ステップ 9	exit 例 : apicl(config-default)# exit	グローバル コンフィギュレーションモードに戻ります。
ステップ 10	aaa authentication login domain {domain-name fallback} 例 : apicl(config)# aaa authentication login domain cisco	デフォルトのログイン認証の設定モードを開始します。ログイン ドメインは、ユーザの認証ドメインを指定します。
ステップ 11	[no] realm {ldap local none radius tacacs}	認証方式を指定します。

	コマンドまたはアクション	目的
	例 : <pre>apic1(config-domain)# realm radius</pre>	
ステップ 12	[no] group group-name 例 : <pre>apic1(config-domain)# group radiusGroup</pre>	認証サーバ グループを指定します。
ステップ 13	exit 例 : <pre>apic1(config-domain)# exit</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 14	aaa banner text 例 : <pre>apic1(config)# aaa banner 'Welcome to APIC'</pre>	ユーザログインの前に表示する情報のバナーを指定します。バナーは、単一引用符で囲む必要があります。
ステップ 15	aaa group {ldap radius tacacs} group-name 例 : <pre>apic1(config)# aaa group radius radiusGroup</pre>	認証サーバグループを作成または設定します。
ステップ 16	[no] server {ip-address hostname} priority priority-number 例 : <pre>apic1(config-radius)# server 192.0.20.71 priority 2</pre>	認証サーバをサーバグループに追加し、サーバグループ内の優先度を指定します。0～17の優先度を指定できます。
ステップ 17	exit 例 : <pre>apic1(config-radius)# exit</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 18	aaa scvmm-certificate certificate-name 例 : <pre>apic1(config)# aaa scvmm-certificate myScvmmCert</pre>	SCVMM の証明書を指定します。 『Cisco ACI Virtualization Guide』を参照してください。
ステップ 19	aaa user default-role {assign-default-role no-login} 例 : <pre>apic1(config)# aaa user default-role assign-default-role</pre>	APIC にログインしようとするユーザ ロールを持たないリモートユーザの対処法を指定します。次のいずれかのオプションを選択できます。 • assign-default-role—ユーザー ロールを持たないリモート ユーザー

	コマンドまたはアクション	目的
		は、デフォルトロールに割り当てられます。 • no-login—ユーザー ロールを持たないリモートユーザーはログインできません。
ステップ 20	show aaa authentication 例 : apic1(config)# show aaa authentication	AAA 方式の設定を表示します。
ステップ 21	show aaa groups 例 : apic1(config)# show aaa groups	AAA サーバ グループの設定を表示します。

例

この例では、AAA を設定する方法を示します。

```

apic1# configure terminal
apic1(config)# aaa authentication login console
apic1(config-console)# realm local
apic1(config-console)# exit
apic1(config)# aaa authentication login default
apic1(config-default)# realm radius
apic1(config-default)# group radiusGroup5
apic1(config-default)# exit
apic1(config)# aaa authentication login domain cisco
apic1(config-domain)# realm none
apic1(config-domain)# exit
apic1(config)# aaa banner 'Welcome to APIC'
apic1(config)# aaa group radius radiusGroup
apic1(config-radius)# server 192.0.20.71 priority 2
apic1(config-radius)# exit
apic1(config)# aaa user default-role assign-default-role
apic1(config)# show aaa authentication
Default : radius
Console : local

apic1(config)# show aaa groups
Total number of Groups : 1

RadiusGroups : radiusGroup5
TacacsGroups :
LdapGroups   :
```

セキュリティ サーバの設定

RADIUS サーバの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>apic1# configure</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] radius-server retries count 例 : <pre>apic1(config)# radius-server retries 1</pre>	APIC からサーバに対して、各 RADIUS 要求を送信する回数の上限を指定します。指定できる範囲は 0 ～ 5 です。 グローバル コンフィギュレーション モードでは、特定の RADIUS ホスト設定で上書きされない限り、このコマンドはすべての RADIUS サーバに適用されます。
ステップ 3	[no] radius-server timeout seconds 例 : <pre>apic1(config)# radius-server timeout 5</pre>	APIC が RADIUS 要求を再送信する前に応答を待機する秒数を指定します。 グローバル コンフィギュレーション モードでは、特定の RADIUS ホスト設定で上書きされない限り、このコマンドはすべての RADIUS サーバに適用されます。
ステップ 4	[no] radius-server host {ip-address hostname} 例 : <pre>apic1(config)# radius-server host 192.0.20.71</pre>	RADIUS サーバの IP アドレスまたはホスト名を指定します。
ステップ 5	(任意) [no] retries count 例 : <pre>apic1(config-host)# retries 2</pre>	この RADIUS サーバにおいて、APIC からサーバに対して、各 RADIUS 要求を送信する回数の上限を指定します。指定できる範囲は 0 ～ 5 です。 再試行回数が設定されていない場合、グローバル値が使用されます。

	コマンドまたはアクション	目的
ステップ 6	(任意) [no] timeout seconds 例 : apicl(config-host) # timeout 3	この RADIUS サーバにおいて、APIC が RADIUS 要求を再送信する前に応答を待機する秒数を指定します。 タイムアウト値が設定されていない場合は、グローバル値が使用されます。
ステップ 7	(任意) [no] descr text 例 : apicl(config-host) # descr "My primary RADIUS server"	この RADIUS サーバに関する説明の情報を提供します。テキストには 128 文字以内の英数字を使用できます。テキストでスペースを含める場合は、単一または二重の引用符で囲む必要があります。
ステップ 8	[no] key key-value 例 : apicl(config-host) # key myRaDiUSpassWoRd	APIC とこの RADIUS サーバの間で認証用に使われる共有秘密テキスト文字列を指定します。キーの文字数は最大 32 文字です。
ステップ 9	[no] port port-number 例 : apicl(config-host) # port 1812	認証用にのみ使用されるこの RADIUS サーバの UDP ポートを指定します。
ステップ 10	[no] protocol {chap mschap pap} 例 : apicl(config-host) # protocol pap	認証用の RADIUS サーバプロトコルを指定します。
ステップ 11	exit 例 : apicl(config-host) #	グローバル コンフィギュレーションモードに戻ります。
ステップ 12	show radius-server 例 : apicl(config) # show radius-server	(任意) RADIUS サーバ情報を表示します。

例

次に、1 つの RADIUS サーバとグローバルに RADIUS の設定を行う例を示します。

```

apicl# configure
apicl(config)# radius-server retries 1
apicl(config)# radius-server timeout 5
apicl(config)# radius-server host 192.0.20.71
apicl(config-host)# retries 2
apicl(config-host)# timeout 3

```

```

apic1(config-host)# descr "My primary RADIUS server"
apic1(config-host)# key myRaDiUSpassWoRd
apic1(config-host)# port 1812
apic1(config-host)# protocol pap
apic1(config-host)# exit
apic1(config)# show radius-server
timeout : 5
retries : 1

```

Total number of servers : 1

```

Hostname : 192.0.20.71
Port      : 1812
Protocol  : pap
Timeout   : 3
Retries   : 2
User      : test
Descr     : My primary RADIUS server

```

TACACS+ サーバの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] tacacs-server retries count 例 : apic1(config)# tacacs-server retries 1	APIC からサーバに対して、各 TACACS+ 要求を送信する回数の上限を指定します。指定できる範囲は 0 ～ 5 です。 グローバル コンフィギュレーション モードでは、特定の TACACS+ ホスト設定で上書きされない限り、このコマンドはすべての TACACS+ サーバに適用されます。
ステップ 3	[no] tacacs-server timeout seconds 例 : apic1(config)# tacacs-server timeout 5	APIC が TACACS+ 要求を再送信する前に応答を待機する秒数を指定します。 グローバル コンフィギュレーション モードでは、特定の TACACS+ ホスト設定で上書きされない限り、このコマンドはすべての TACACS+ サーバに適用されます。

	コマンドまたはアクション	目的
ステップ 4	[no] tacacs-server host {ip-address hostname} 例 : <pre>apicl(config)# tacacs-server host 192.0.20.71</pre>	TACACS+ サーバの IP アドレスまたはホスト名を指定します。
ステップ 5	(任意) [no] retries count 例 : <pre>apicl(config-host)# retries 2</pre>	この TACACS+ サーバにおいて、APIC からサーバに対して、各 TACACS+ 要求を送信する回数の上限を指定します。指定できる範囲は 0 ～ 5 です。 再試行回数が設定されていない場合、グローバル値が使用されます。
ステップ 6	[no] key 例 : <pre>apicl(config-host)# key Enter key: myTacAcSpaSSWoRd Enter key again: myTacAcSpaSSWoRd</pre>	APIC とこの TACACS+ サーバの間で認証用に使われる共有秘密テキスト文字列を指定します。キーの文字数は最大 32 文字です。セキュリティを向上させるキー値の入力は、インタラクティブです。
ステップ 7	[no] port port-number 例 : <pre>apicl(config-host)# port 49</pre>	TACACS+ アカウンティング メッセージ用に使用されるこの TACACS+ サーバの UDP ポートを指定します。
ステップ 8	[no] protocol {chap mschap pap} 例 : <pre>apicl(config-host)# protocol pap</pre>	認証用の TACACS+ サーバプロトコルを指定します。
ステップ 9	exit 例 : <pre>apicl(config-host)#</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 10	show tacacs-server 例 : <pre>apicl(config)# show tacacs-server</pre>	(任意) TACACS+ サーバ情報を表示します。

例

次に、1 つの TACACS+ サーバとグローバルに TACACS+ の設定を行う例を示します。

```
apicl# configure
apicl(config)# tacacs-server retries 1
apicl(config)# tacacs-server timeout 5
apicl(config)# tacacs-server host 192.0.20.72
```

```

apic1(config-host)# retries 2
apic1(config-host)# timeout 3
apic1(config-host)# key myTaCaCspassWoRd
apic1(config-host)# port 49
apic1(config-host)# protocol pap
apic1(config-host)# exit
apic1(config)# show tacacs-server
timeout : 5
retries : 1

Total number of servers : 1

Hostname : 192.0.20.72
Port      : 1812
Protocol  : pap
Timeout   : 3
Retries   : 2
User      : test

```

LDAP サーバの設定

Some **ldap-server** commands can be entered in either the global configuration mode or in the configuration mode for a specific LDAP host. グローバル コンフィギュレーション モードでは、特定の LDAP ホスト設定で上書きされない限り、コマンドはすべての LDAP サーバに適用されます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : switch# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] ldap-server host {ip-address hostname} 例 : apic1(config)# ldap-server host 192.0.20.73	LDAP サーバの IP アドレスまたはホスト名を指定し、そのサーバのコンフィギュレーションモードを開始します。
ステップ 3	[no] ldap-server attribute attribute-name 例 : apic1(config-host)# ldap-server attribute memberOf	CiscoAVPair として使用する LDAP エンドポイント属性を指定します。 グローバル コンフィギュレーション モードでは、特定の LDAP ホスト設定で上書きされない限り、このコマンドはすべての LDAP サーバに適用されます。
ステップ 4	[no] ldap-server basedn 例 :	サーバが認可要求を受信した場合に検索を開始する LDAP 階層の位置を指定します。最大 127 文字の文字列を設定

	コマンドまたはアクション	目的
	<code>apicl(config-host)# ldap-server basedn DC=sampldesign,DC=com</code>	できます。文字列でスペースは使用できませんが、他の特殊文字は使用できます。 グローバル コンフィギュレーション モードでは、特定の LDAP ホスト設定で上書きされない限り、このコマンドはすべての LDAP サーバに適用されます。
ステップ 5	[no] ldap-server binddn 例 : <code>apicl(config-host)# ldap-server binddn CN=ucsbind,OU=CiscoUsers,DC=sampldesign,DC=com</code>	ベース DN のすべてのオブジェクトに対する読み取り権限と検索権限を持つ、LDAP データベース アカунツの識別名 (DN) を指定します。最大 127 文字の文字列を設定できます。文字列でスペースは使用できませんが、他の特殊文字は使用できます。
ステップ 6	[no] ldap-server retries count 例 : <code>apicl(config-host)# ldap-server retries 1</code>	APIC からサーバに対して、各 LDAP 要求を送信する回数の上限を指定します。指定できる範囲は 0 ～ 5 です。 グローバル コンフィギュレーション モードでは、特定の LDAP ホスト設定で上書きされない限り、このコマンドはすべての LDAP サーバに適用されます。
ステップ 7	[no] ldap-server timeout seconds 例 : <code>apicl(config-host)# ldap-server timeout 30</code>	APIC が LDAP 要求を再送信する前に応答を待機する秒数を指定します。 グローバル コンフィギュレーション モードでは、特定の LDAP ホスト設定で上書きされない限り、このコマンドはすべての LDAP サーバに適用されます。
ステップ 8	[no] ldap-server filter filter-expression 例 : <code>apicl(config-host)# ldap-server filter sAMAccountName=\$userid</code>	LDAP 検索の結果をフィルタ処理するフィルタを指定します。フィルタには、最大 63 文字を指定できます。 グローバル コンフィギュレーション モードでは、特定の LDAP ホスト設定で上書きされない限り、このコマンドはすべての LDAP サーバに適用されます。

	コマンドまたはアクション	目的
ステップ 9	[no] key <i>key-value</i> 例 : <pre>apic1(config-host)# key Enter key: myLdAppassWoRd Enter key again: myLdAppassWoRd</pre>	APIC とこの LDAP サーバの間で認証用に使用される共有秘密テキスト文字列を指定します。キーの文字数は最大 32 文字です。
ステップ 10	[no] port <i>port-number</i> 例 : <pre>apic1(config-host)# port 389</pre>	認証用の LDAP サーバのポートを指定します。
ステップ 11	(任意) [no] retries <i>count</i> 例 : <pre>apic1(config-host)# retries 2</pre>	この LDAP サーバにおいて、APIC からサーバに対して、各 LDAP 要求を送信する回数の上限を指定します。指定できる範囲は 0 ～ 5 です。 再試行回数が設定されていない場合、グローバル値が使用されます。
ステップ 12	[no] enable-ssl 例 : <pre>apic1(config-host)# enable-ssl</pre>	LDAP プロバイダーとの SSL 接続を有効にします。
ステップ 13	[no] ssl-validation-level [permissive strict] 例 : <pre>apic1(config-host)# ssl-validation-level permissive</pre>	LDAP サーバの SSL 証明書検証レベルを設定します。
ステップ 14	(任意) [no] timeout <i>seconds</i> 例 : <pre>apic1(config-host)# timeout 3</pre>	この LDAP サーバにおいて、APIC が LDAP 要求を再送信する前に応答を待機する秒数を指定します。 タイムアウト値が設定されていない場合は、グローバル値が使用されます。
ステップ 15	exit 例 : <pre>apic1(config-host)# exit</pre>	グローバル コンフィギュレーション モードに戻ります。
ステップ 16	show ldap-server 例 : <pre>apic1(config)# show ldap-server</pre>	

例

次に、LDAP サーバの設定をグローバルおよび 1 つの LDAP サーバに対して行う例を示します。

```
apicl# configure
apicl(config)# ldap-server retries 1
apicl(config)# ldap-server timeout 30
apicl(config)# ldap-server host 192.0.20.73
apicl(config-host)# retries 2
apicl(config-host)# timeout 3
apicl(config-host)# filter sAMAccountName=$userid
apicl(config-host)# key myLdAppassWoRd
apicl(config-host)# ssl-validation-level permissive
apicl(config-host)# enable-ssl
apicl(config-host)# port 389
apicl(config-host)# exit
apicl(config)# show ldap-server
timeout : 30
retries : 1
filter  : sAMAccountName=$userid

Total number of servers : 1

Hostname : 192.0.20.73
Port     : 389
Timeout  : 3
Retries  : 2
SSL      : yes
SSL Level : permissive
User     : test
```

パスワードポリシーの設定

このトピックのパスワードポリシー設定では、ローカル認証されたすべての APIC ユーザのパスワード履歴やパスワード変更間隔プロパティを設定します。ローカル認証されたユーザのそれぞれに異なるパスワードポリシーを指定することはできません。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] password change-count count 例 : apicl(config)# password change-count 5	変更間隔内で許可されるパスワード変更の回数を設定します。指定できる範囲は 0 ～ 10 回です。

	コマンドまたはアクション	目的
ステップ 3	[no] password change-during-interval {enable disable} 例 : <pre>apic1(config)# password change-during-interval enable</pre>	ローカル認証されたユーザが変更間隔の間に実行できるパスワード変更の回数の制限を有効または無効にします。
ステップ 4	[no] password change-interval hours 例 : <pre>apic1(config)# password change-interval 300</pre>	When the change-during-interval is enabled, restricts the number of password changes a locally authenticated user can make within a given number of hours. 範囲は 1 ～ 745 時間です。
ステップ 5	[no] password no-change-interval hours 例 : <pre>apic1(config)# password no-change-interval 60</pre>	ユーザがパスワードを再度変更できない最小期間を設定します。範囲は 1 ～ 745 時間です。
ステップ 6	password expiration-warn-time 例 : <pre>apic1(config)# password expiration-warn-time 5</pre>	パスワードの有効期限の前に警告を表示する警告期間を設定します。範囲は 0 ～ 30 日です。
ステップ 7	[no] password history-count count 例 : <pre>apic1(config)# password history-count 10</pre>	パスワード履歴のカウントにより、ローカル認証されたユーザが何度も同じパスワードを再利用しないようにすることができます。このプロパティが設定されている場合、APIC はローカル認証されたユーザが以前に使用したパスワードを最大 15 個まで保存します。パスワードは最近のものから時系列の逆順で格納され、履歴カウントがしきい値に達した場合に、最も古いパスワードだけを再利用可能にします。 あるパスワードが再利用可能になるまでに、ユーザはパスワード履歴カウントで設定された数だけパスワードを作成して使用する必要があります。たとえば、パスワード履歴カウントを 8 に設定した場合、ローカル認証されたユーザは、9 番目のパスワードが期限切れになるまで、最初のパスワードを再利用できません。 デフォルトでは、パスワード履歴は 0 に設定されます。この値によって履歴カウントが無効化されるため、ユーザはいつ

	コマンドまたはアクション	目的
		でも以前のパスワードを使用できます。 If necessary, you can clear a user's password history using the clear-pwd-history command in the username configuration mode for that user.
ステップ 8	[no] password pwd-strength-check 例 : <pre>apic1(config)# password pwd-strength-check</pre>	すべてのユーザに強力なパスワードの使用を強制します。

例

次に、ローカル認証されたユーザのグローバルパスワード設定を設定する例を示します。

```
apic1# configure
apic1(config)# password change-count 5
apic1(config)# password change-during-interval enable
apic1(config)# password change-interval 300
apic1(config)# password no-change-interval 60
apic1(config)# password expiration-warn-time 5
apic1(config)# password history-count 10
apic1(config)# password pwd-strength-check
```

この例では、ローカル認証されたユーザがパスワード変更後 48 時間以内にパスワードを変更できないようにする方法を示しています。

```
apic1# configure
apic1(config)# password change-during-interval disable
apic1(config)# password no-change-interval 48
```

この例では、ローカル認証されたユーザがパスワード変更後 24 時間以内にパスワードを 1 回だけ変更できるようにする方法を示しています。

```
apic1# configure
apic1(config)# password change-count 1
apic1(config)# password change-during-interval enable
apic1(config)# password change-interval 24
```

ユーザの設定

ローカルで認証されたユーザの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	username {name admin} 例 : apic1(config)# username user5	ローカル認証されたユーザアカウントの作成または既存のユーザを設定します。最大 28 文字までの名前を指定できます。
ステップ 3	[no] first-name first 例 : apic1(config-username)# first-name George	このユーザの名を設定します。
ステップ 4	[no] last-name last 例 : apic1(config-username)# last-name Washington	このユーザの姓を設定します。
ステップ 5	[no] email email-address 例 : apic1(config-username)# email gwashington@exampleCorp.com	このユーザの電子メールアドレスを設定します。
ステップ 6	[no] phone phone-number 例 : apic1(config-username)# phone 14085551212	このユーザの電話番号を設定します。
ステップ 7	[no] account-status {active inactive status} 例 : apic1(config-username)# account-status active	このユーザアカウントをアクティブ化または非アクティブ化します。

	コマンドまたはアクション	目的
ステップ 8	clear-pwd-history 例 : <pre>apicl(config-username) # clear-pwd-history</pre>	ユーザのパスワード履歴リストをクリアし、このユーザが以前のパスワードを再使用できるようにします。
ステップ 9	[no] expires 例 : <pre>apicl(config-username) # expires</pre>	Enables expiration of this user account at the date and time configured by the expiration command.
ステップ 10	expiration date-time 例 : <pre>apicl(config-username) # expiration 2017-12-31T23:59+08:00</pre>	このユーザアカウントの有効期限の日時を設定します。形式は UTC 日付形式 (YYYY-MM-DDThh:mmTZD) です。You must also enable expiration by configuring the expires command.
ステップ 11	password password 例 : <pre>apicl(config-username) # password c1\!\$c0123</pre>	ユーザ パスワードを設定します。 (注) このコマンドでは「\$」または「!」などの特殊文字をバックスラッシュでエスケープし（「\!\$」）、 Bash によって誤って解釈されないようにする必要があります。エスケープのバックスラッシュはこのコマンドでパスワードを設定するときに必要なだけであり、ユーザはログイン時にバックスラッシュを入力しません。
ステップ 12	[no] pwd-lifetime days 例 : <pre>apicl(config-username) # pwd-lifetime 90</pre>	ユーザパスワードの有効期間を設定します。範囲は 0 ～ 3650 日です。
ステップ 13	[no] domain {all common mgmt domain-name} 例 : <pre>apicl(config-username) # domain mySecDomain</pre>	このユーザが属する AAA ドメインを指定または作成します。
ステップ 14	[no] role role 例 : <pre>apicl(config-domain) # role tenant-admin</pre>	ユーザドメインの特権ビットマスクを設定する AAA ドメイン ロールを作成します。

	コマンドまたはアクション	目的
ステップ 15	[no] priv-type {readPriv writePriv} 例 : apic1(config-role) # priv-type writePriv	ユーザドメインの特権ビットマスクを設定する AAA ドメイン ロールを作成します。
ステップ 16	exit 例 : apic1(config-role) # exit	ドメインコンフィギュレーションモードに戻ります。
ステップ 17	exit 例 : apic1(config-domain) # exit	ユーザ名コンフィギュレーションモードに戻ります。
ステップ 18	show username name 例 : apic1(config-username) # show username user5	このユーザに関する構成の詳細情報を表示します。

例

次に、ローカル ユーザの設定方法を示します。

```

apic1# configure terminal
apic1(config) # username user5
apic1(config-username) # first-name George
apic1(config-username) # last-name Washington
apic1(config-username) # email gwashington@exampleCorp.com
apic1(config-username) # phone 14085551212
apic1(config-username) # account-status active
apic1(config-username) # domain mySecDomain
apic1(config-username) # clear-pwd-history
apic1(config-username) # expires
apic1(config-username) # expiration 2017-12-31T23:59+08:00
apic1(config-username) # password c1$c0123
apic1(config-username) # pwd-lifetime 90
apic1(config-username) # domain mySecDomain
apic1(config-domain) # role tenant-admin
apic1(config-role) # priv-type writePriv
apic1(config-role) # exit
apic1(config-domain) # exit
apic1(config-username) # show username user5
UserName                : user5
First-Name              : George
Last-Name               : Washington
Email                   : gwashington@exampleCorp.com
Account Status          : active
Password strength check : yes

```

次のタスク

ローカル ユーザの SSH キーまたは証明書を設定するには、「証明書および SSH キーの設定」を参照してください。

ローカル ユーザの証明書および SSH キーの設定

この項では、ローカルユーザがパスワードを要求されることなく、ログインできるように証明書または SSH キーを設定する方法について説明します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	username {name admin} 例 : apic1 (config) # username user5	ローカル認証されたユーザアカウントの作成または既存のユーザを設定します。最大 28 文字までの名前を指定できます。
ステップ 3	[no] certificate certificate-name 例 : apic1 (config-username) # certificate myCertificate	証明書コンフィギュレーション モードを開始します。
ステップ 4	data certificate-data 例 : apic1 (config-certificate) # data -----BEGIN CERTIFICATE-----MIIC4j.....	PEM エンコードされた証明書を設定します。
ステップ 5	exit 例 : apic1 (config-certificate) # exit	ユーザ名コンフィギュレーション モードに戻ります。
ステップ 6	[no] ssh-key ssh-key-name 例 : apic1 (config-username) # ssh-key mySSHkey	SSH キーを設定すると、パスワードを要求されることなく、SSH クライアントを使用してログインできます。
ステップ 7	data key-data 例 : apic1 (config-ssh-key) # data AAAAB3NzaC1yc2EAA.....	SSH キーを設定します。キーの文字数は最大 64 文字です。

	コマンドまたはアクション	目的
ステップ 8	exit 例 : apicl(config-ssh-key)# exit	ユーザ名コンフィギュレーション モードに戻ります。

例

この例では、ローカル ユーザの SSH キーと証明書を設定する方法を示します。

```

apicl# configure terminal
apicl(config)# username user5
apicl(config-username)# certificate myCertificate
apicl(config-certificate)# data -----BEGIN CERTIFICATE-----MIIC4j.....
apicl(config-certificate)# exit
apicl(config-username)# ssh-key mySSHkey
apicl(config-ssh-key)# data AAAAB3NzaClyc2EAA...
apicl(config-ssh-key)# exit

```

公開キー インフラストラクチャの設定

認証局との信頼チェーンの設定

証明機関（CA）は証明書要求を管理して、ホスト、ネットワーク デバイス、ユーザなどの参加エンティティに証明書を発行します。APIC では、信頼できる CA の自己署名ルート証明書（または下位 CA の証明書チェーン）をローカルに保存しています。信頼できる CA について保存された情報をトラストポイントと呼び、CA 自体もトラストポイント CA と呼びます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] crypto ca trustpoint-name 例 : apicl(config)# crypto ca myCA	選択したトラストポイントの認証局（CA）のコンフィギュレーション モードを開始します。
ステップ 3	[no] cert-chain pem-data 例 : apicl(config-ca)# cert-chain -----BEGIN CERTIFICATE----- MIIC4jCCAoygAw.....	PEM フォーマットの証明書チェーンを保存します。トラストポイントから信頼されたルート証明機関に信頼のチェーン全体を入力します。

例

この例では、CA を設定する方法を示します。

```
apicl# configure

apicl(config)# crypto ca myCA
apicl(config-ca)# cert-chain -----BEGIN CERTIFICATE----- MIIC4jCCAoygAw.....
```

キーとキーリングの設定

RSA キー ペアを作成し、RSA キー ペアと APIC が登録しようとしているトラストポイント CA を関連付けることによって、APIC のアイデンティティ証明書を入手することができます。RSA キーは APIC によって暗号キーリングに保存されます。

APIC ソフトウェアでは、設定変更が可能なキーのサイズ（またはモジュラス）で RSA キー ペアを作成できます。デフォルトのキーのサイズは 512 です。また、RSA キー ペアのラベルも設定できます。デフォルトのキー ラベルは、デバイスの完全修飾ドメイン名（FQDN）です。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] crypto keyring {default keyring-name} 例： apicl(config)# crypto keyring myKeyring	キーリングを作成または設定して SSL 証明書を保持します。
ステップ 3	regen 例： apicl(config-keyring)# regen	RSA キー ペアの再生を強制します。
ステップ 4	[no] cert certificate-data 例： apicl(config-keyring)# cert "-----BEGIN CERTIFICATE----- MIIC4jCCAoygAw....."	公開キーと署名された情報を含んだ証明書をインポートします。証明書データは引用符で囲む必要があります。
ステップ 5	[no] tp certificate-name 例： apicl(config-keyring)# tp myCertificate	デバイス アイデンティティの信頼できるソースからのサードパーティ証明書を設定します。

	コマンドまたはアクション	目的
ステップ 6	[no] key key-data 例 : <pre>apicl(config-keyring)# key XXXXXXXXXXXXXXXXXXXXXXXXXXXX</pre>	証明書の秘密キーを作成します。
ステップ 7	[no] modulus {mod512 mod1024 mod1536 mod2048} 例 : <pre>apicl(config-keyring)# modulus mod1024</pre>	暗号キーの長さを設定します。
ステップ 8	exit 例 : <pre>apicl(config-keyring)# exit</pre>	グローバル コンフィギュレーション モードに戻ります。

例

この例では、キーリングを設定する方法を示します。

```
apicl# configure
apicl(config)# crypto keyring myKeyring
apicl(config-keyring)# cert "-----BEGIN CERTIFICATE----- MIIC4jCCAoygAw.....
apicl(config-keyring)# tp myCertificate
apicl(config-keyring)# key XXXXXXXXXXXXXXXXXXXXXXXX
apicl(config-keyring)# modulus mod1024
apicl(config-keyring)# exit
```

証明書署名要求の生成

証明書署名要求（CSR）は、デジタルID証明書を申請するために申請者がCAに送信するメッセージです。CSRを作成する前に、申請者は、最初に秘密キーを守るキーペアを生成します。CSRには、申請者が生成する公開キーなどの申請者を特定する情報が含まれています。対応する秘密キーはCSRに含まれていませんが、要求全体に対するデジタル署名を行うために使用されます。

始める前に

証明書署名要求（CSR）を生成する前に、トラストポイントの認証局（CA）を設定し、キーペアを生成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] crypto keyring {default keyring-name} 例 : apicl(config)# crypto keyring default	キーリングを作成または設定して SSL 証明書を保持します。
ステップ 3	csr 例 : apicl(config-keyring)# csr	このキーリングに対する証明書署名要求を作成します。
ステップ 4	subj-name name 例 : apicl(config-csr)# subj-name www.exampleCorp.com	要求側デバイスの完全修飾ドメイン名または識別名を設定します。名前には最大 64 文字を使用できます。
ステップ 5	[no] cert certificate-data 例 : apicl(config-csr)# cert "-----BEGIN CERTIFICATE----- MIIC4jCCAoygAw....."	公開キーと署名された情報を含んだ証明書をインポートします。証明書データは引用符で囲む必要があります。
ステップ 6	password 例 : apicl(config-csr)# password Enter password: c1\$c0123 Enter password again: c1\$c0123	新しいパスワードを設定します。
ステップ 7	org-name 例 : apicl(config-csr)# org-name ExampleCorp	組織の完全な正式名称を設定します。
ステップ 8	org-unit-name 例 : apicl(config-csr)# org-unit-name Sales	組織内の部門または部署の名前を設定します。
ステップ 9	email 例 : apicl(config-csr)# email admin@exampleCorp.com	組織の担当者の電子メールアドレスを設定します。

	コマンドまたはアクション	目的
ステップ 10	locality <i>city-name</i> 例 : apic1(config-csr)# locality SanJose	組織の市町村を設定します。
ステップ 11	state <i>state</i> 例 : apic1(config-csr)# state CA	組織が存在する州や県を設定します。
ステップ 12	country <i>country-code</i> 例 : apic1(config-csr)# country US	組織の所在地の国を表す 2 文字の ISO コードを設定します。
ステップ 13	exit 例 : apic1(config-csr)# exit	キーリング コンフィギュレーション モードに戻ります。

例

次に、証明書署名要求（CSR）を生成する例を示します。

```

apic1# configure
apic1(config)# crypto keyring default
apic1(config-keyring)# csr
apic1(config-csr)# subj-name www.exampleCorp.com
apic1(config-csr)# cert "-----BEGIN CERTIFICATE----- MIIC4jCCAoygAw.....
apic1(config-csr)# pwd c1$c0123
apic1(config-csr)# org-name ExampleCorp
apic1(config-csr)# org-unit-name Sales
apic1(config-csr)# email admin@exampleCorp.com
apic1(config-csr)# locality SanJose
apic1(config-csr)# state CA
apic1(config-csr)# country US
apic1(config-csr)# exit

```

次のタスク

CSR を CA に送信します。

ウェブトークンの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apicl# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] crypto webtoken 例 : <code>apicl(config)# crypto webtoken</code>	
ステップ 3	[no] max-validity-period hours 例 : <code>apicl(config-webtoken)# max-validity-period 10</code>	webtoken の最長有効期間を設定します。範囲は 4 ～ 24 時間です。
ステップ 4	[no] session-record-flags csv-list 例 : <code>apicl(config-webtoken)# session-record-flags login,refresh</code>	セッション レコードの更新を有効または無効にします。The session record flags are specified as a comma-separated value list of one or more of the following flags: login , logout , and refresh .
ステップ 5	[no] ui-idle-timeout-seconds seconds 例 : <code>apicl(config-webtoken)# ui-idle-timeout-seconds 120</code>	ログインの更新を要求するまでの最大の GUI アイドル継続時間を設定します。値の範囲は 60 ～ 65525 秒です。
ステップ 6	[no] webtoken-timeout-seconds seconds 例 : <code>apicl(config-webtoken)# webtoken-timeout-seconds 1200</code>	webtoken のタイムアウト間隔を設定します。値の範囲は 600 ～ 9600 秒です。
ステップ 7	exit 例 : <code>apicl(config-webtoken)# exit</code>	

例

次に、webtoken を設定する例を示します。

```
apicl# configure
apicl(config)# crypto webtoken
apicl(config-webtoken)# max-validity-period 10
```

```

apicl(config-webtoken)# session-record-flags login,refresh
apicl(config-webtoken)# ui-idle-timeout-seconds 120
apicl(config-webtoken)# webtoken-timeout-seconds 1200

```

通信ポリシーの設定

HTTP ポリシーの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] comm-policy {default policy-name} 例 : apicl(config)# comm-policy myCommPolicy	通信ポリシー コンフィギュレーション モードを開始します。
ステップ 3	http 例 : apicl(config-comm-policy)# http	HTTP ポリシー コンフィギュレーション モードを開始します。
ステップ 4	[no] admin-state-enable 例 : apicl(config-http)# admin-state-enable	HTTP 通信サービスをイネーブルにします。
ステップ 5	[no] allow-origin url 例 : apicl(config-http)# allow-origin www.example.com	Access-Control-Allow-Origin ヘッダーで返す URL を指定します。
ステップ 6	[no] port port-number 例 : apicl(config-http)# port 8080	HTTP 通信サービスに使用するポートを設定します。
ステップ 7	[no] redirect 例 : apicl(config-http)# no redirect	HTTP リダイレクションをイネーブルにします。
ステップ 8	[no] request-status-count count 例 :	追跡する HTTP 要求の最大数を設定します。指定できる範囲は 0 ～ 10240 です。

	コマンドまたはアクション	目的
	apicl(config-http)# request-status-count 512	
ステップ 9	exit 例 : apicl(config-http)# exit	通信ポリシー コンフィギュレーション モードに戻ります。

例

次に、HTTP サービスを設定する例を示します。

```
apicl# configure
apicl(config)# comm-policy myCommPolicy
apicl(config-comm-policy)# http
apicl(config-http)# admin-state-enable
apicl(config-http)# allow-origin www.example.com
apicl(config-http)# port 8080
apicl(config-http)# no redirect
apicl(config-http)# request-status-count 512
apicl(config-http)# exit
```

HTTPS ポリシーの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] comm-policy {default <i>policy-name</i>} 例 : apicl(config)# comm-policy myCommPolicy	通信ポリシー コンフィギュレーション モードを開始します。
ステップ 3	https 例 : apicl(config-comm-policy)# https	HTTPS ポリシー コンフィギュレーション モードを開始します。
ステップ 4	[no] admin-state-enable 例 : apicl(config-https)# admin-state-enable	HTTPS 通信サービスをイネーブルにし ます。

	コマンドまたはアクション	目的
ステップ 5	[no] port <i>port-number</i> 例 : <code>apic1(config-https) # port 443</code>	HTTPS 通信サービスに使用するポートを設定します。
ステップ 6	[no] request-status-count <i>count</i> 例 : <code>apic1(config-https) # request-status-count 512</code>	追跡する HTTPS 要求の最大数を設定します。指定できる範囲は 0 ～ 10240 です。
ステップ 7	[no] ssl-protocols {<i>TLSv1</i> <i>TLSv1.1</i> <i>TLSv1.2</i>} 例 : <code>apic1(config-https) # ssl-protocols TLSv1.1,TLSv1.2</code>	カンマ区切りのリストでサポートされる SSL プロトコルを指定します。The options are TLSv1 , TLSv1.1 , and TLSv1.2 .
ステップ 8	[no] use-keyring <i>keyring-name</i> 例 : <code>apic1(config-https) # use-keyring myKeyRing</code>	HTTPS サーバの SSL 証明書に使用するキーリングを指定します。
ステップ 9	exit 例 : <code>apic1(config-https) # exit</code>	通信ポリシー コンフィギュレーションモードに戻ります。

例

次に、HTTPS サービスを設定する例を示します。

```
apic1# configure
apic1(config)# comm-policy myCommPolicy
apic1(config-comm-policy)# https
apic1(config-https)# admin-state-enable
apic1(config-https)# port 443
apic1(config-https)# request-status-count 512
apic1(config-https)# ssl-protocols TLSv1.1,TLSv1.2
apic1(config-https)# use-keyring myKeyRing
apic1(config-https)# exit
```

SSH ポリシーの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] comm-policy {default <i>policy-name</i>} 例 : apic1 (config) # comm-policy myCommPolicy	通信ポリシー コンフィギュレーション モードを開始します。
ステップ 3	ssh-service 例 : apic1 (comm-policy) # ssh-service	SSH ポリシー コンフィギュレーション モードを開始します。
ステップ 4	[no] admin-state-enable 例 : apic1 (config-ssh-service) # admin-state-enable	HTTP 通信サービスをイネーブルにします。
ステップ 5	[no] port <i>port-number</i> 例 : apic1 (config-ssh-service) # port 22	SSH 通信サービスに使用するポートを設定します。
ステップ 6	exit 例 : apic1 (config-ssh-service) # exit	通信ポリシー コンフィギュレーション モードに戻ります。

例

次に、SSH サービスを設定する例を示します。

```
apic1# configure
apic1 (config) # comm-policy myCommPolicy
apic1 (config-comm-policy) # ssh-service
apic1 (config-ssh-service) # admin-state-enable
apic1 (config-ssh-service) # port 22
apic1 (config-ssh-service) # exit
```

Telnet ポリシーの設定

始める前に

Telnet 通信を可能にするには、Telnet トラフィック（通常、TCP および UDP ポート 23 上）を許可するアウトオブバンド コントラクトを設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] comm-policy {default <i>policy-name</i>} 例： apic1(config)# comm-policy myCommPolicy	通信ポリシー コンフィギュレーション モードを開始します。
ステップ 3	telnet 例： apic1(config-comm-policy)# telnet	Telnet ポリシー コンフィギュレーション モードを開始します。
ステップ 4	[no] admin-state-enable 例： apic1(config-telnet)# admin-state-enable	Telnet 通信サービスをイネーブルにします。
ステップ 5	[no] port <i>port-number</i> 例： apic1(config-telnet)# port 23	Telnet 通信サービスに使用するポートを設定します。
ステップ 6	exit 例： apic1(config-telnet)# exit	通信ポリシー コンフィギュレーション モードに戻ります。

例

次に、Telnet サービスを設定する例を示します。

```
apic1# configure
apic1(config)# comm-policy myCommPolicy
apic1(config-comm-policy)# telnet
apic1(config-telnet)# admin-state-enable
apic1(config-telnet)# port 23
apic1(config-telnet)# exit
```


AES 暗号化の設定

Cisco APIC リリース 1.1(2) 以降、APIC 設定ファイルのセキュアなプロパティは、AES-256 暗号化を有効にすることで暗号化できます。AES 暗号化はグローバル設定オプションです。すべてのセキュア プロパティは AES 構成設定に従っています。テナント設定などの ACI ファブリック設定のサブセットを AES 暗号化を使用してエクスポートするが、ファブリック設定の残りの部分は暗号化しないということはありません。セキュアなプロパティのリストについては、『*Cisco Application Centric Infrastructure Fundamentals*』の「Appendix K: Secure Properties」を参照してください。

APIC は AES-256 キーを生成する 16 ～ 32 文字パスフレーズを使用します。APIC GUI は、AES パスフレーズのハッシュを表示します。このハッシュは、同じパスフレーズが 2 つの ACI ファブリックに使用されているかどうかを確認するために使用することができます。このハッシュをクライアント コンピュータにコピーして、別の ACI ファブリックのパスフレーズ ハッシュと比較できます。これにより、それらのハッシュが同じパスフレーズを使用して生成されたかどうかを確認できます。ハッシュを使用して、元のパスフレーズまたは AES-256 キーを再構築することはできません。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーション モードに入ります。
ステップ 2	crypto aes 例 : <code>apic1(config)# crypto aes</code>	AES コンフィギュレーション モードを開始します。
ステップ 3	(任意) clear-encryption-key 例 : <code>apic1(config-aes)# clear-encryption-key</code>	既存の AES 暗号化キーを削除します。
ステップ 4	passphrase 例 : <code>apic1(config-aes)# passphrase</code> Enter passphrase: "This is my passphrase" Enter passphrase again: "This is my passphrase"	AES 暗号化のパスフレーズを指定します。パスフレーズは、16 ～ 32 文字で、引用符で囲む必要があります。セキュリティを向上させる、パスフレーズの入力、は、インタラクティブです。
ステップ 5	[no] encryption 例 : <code>apic1(config-aes)# encryption</code>	AES 暗号化をイネーブル（またはディセーブル）にします。

例

この例では、AES 暗号化を有効にする方法およびパスフレーズを設定する方法を示します。

```
apic1# configure
apic1(config)# crypto aes
apic1(config-aes)# clear-encryption-key
apic1(config-aes)# passphrase "This is my passphrase"
apic1(config-aes)# encryption
```

ファブリック セキュア モードの設定

ファブリック セキュア モードでは、ファブリック機器に物理的にアクセスできるユーザが管理者による手動の認証を受けずにファブリックにスイッチまたは APIC コントローラを追加するのを防止できます。Cisco APIC リリース 1.2(1x) 以降、ファームウェアでは、ファブリックのスイッチとコントローラに有効なシスコデジタル署名付き証明書と関連付けられた有効なシリアル番号があることを確認します。この検証は、このリリースへのアップグレード時、またはファブリックの最初のインストール時に行われます。この機能のデフォルト設定は、非制限モードです。既存のファブリックは、リリース 1.2(1) へのアップグレード後も継続して実行されます。ファブリック全体のアクセス権限を持つ管理者は、ストリクトモードを有効にする必要があります。

非制限モード（デフォルト）は次のように動作します。

- 1 つ以上のスイッチに無効な証明書がある場合でも既存のファブリックは正常に動作することができます。
- シリアル番号に基づく認証は適用されません。
- 自動検出されたコントローラとスイッチはシリアル番号の認証を適用せずにファブリックに参加することができます。

ストリクトモードは次のように動作します。

- 有効なシスコ シリアル番号および SSL 証明書を持つスイッチのみが許可されます。
- シリアル番号に基づく認証が適用されます。
- コントローラとスイッチがファブリックに参加するには管理者が手動で承認する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apicl# configure</code>	コンフィギュレーション モードに入ります。
ステップ 2	system fabric-security-mode {permissive strict} 例 : <code>apicl(config)# system fabric-security-mode strict</code>	ファブリック セキュリティ モードを指定します。
ステップ 3	system controller-id controller-id {approve reject} 例 : <code>apicl(config)# system controller-id FCH1750V025 approve</code>	ストリクト モードで、コントローラのファブリックへの参加を承認または拒否します。

例

次に、ファブリック セキュリティ モードをストリクトに変更する例を示します。

```
apicl# configure
apicl(config)# system fabric-security-mode strict
```

次に、ストリクトモードが設定されているときにコントローラのファブリックへの参加を承認する例を示します。

```
apicl# configure
apicl(config)# system controller-id FCH1750V025 approve
```

COOP 認証の設定

COOP 認証について

マッピング情報（ロケーションおよび ID）をスパイン プロキシに伝達するために、Council of Oracles Protocol（COOP）を使用します。リーフ スイッチは、ZeroMQ（Zero Message Queue または ZMQ）を使用してエンドポイントアドレス情報をスパインに転送します。スパイン ノードで実行している COOP によって、すべてのスパイン ノードが一貫性のあるエンドポイントアドレスとロケーション情報のコピーを維持することができ、さらに、ロケーションマッピン

データベースに対するエンドポイント ID の分散ハッシュ テーブル (DHT) レポジトリを維持することができます。

ユーザは COOP 認証なしで任意の COOP メッセージを送信できますが、これはファブリック ノードによって実行されます。Cisco APIC リリース 2.0 は MD5 TCP オプションを追加して ZMQ TCP 転送に認証と整合性の保護を提供します。次の 2 つの認証モードがサポートされています。

- **Compatible** : COOP はメッセージの転送に MD5 認証接続と非認証 ZMQ 接続の両方を許可します。COOP データ パス通信は、セキュアな接続を介した転送を優先します。
- **Strict** : COOP は MD5 認証済みの ZMQ 接続のみを許可します。

COOP の認証タイプの設定を変更すると、次のように影響します。

- 互換モードからストリクトモードに設定を変更すると、すべての非認証 ZMQ 接続が切断されます。
- ストリクトモードから互換モードに設定を変更すると、COOP はすぐに認証済みの ZMQ 接続と非認証の ZMQ 接続の両方を許可するようになります。

COOP 認証の設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	coop-fabric 例 : <code>apic1(config)# coop-fabric</code>	COOP ファブリック コンフィギュレーション モードを開始します。
ステップ 3	authentication type {compatible strict} 例 : <code>apic1(config-coop-fabric)# authentication type compatible</code>	次のいずれかの COOP 認証タイプを選択します。 • compatible : COOP は MD5 認証接続と非人稱 ZMQ 接続を許可します。 • strict : MD5 認証 ZMQ 接続のみを許可します。

例

次に、互換モードで COOP 認証を設定する例を示します。

```
apicl# configure
apicl(config)# coop-fabric
apicl(config-coop-fabric# authentication type compatible
```

FIPS の設定

連邦情報処理標準（FIPS）について

連邦情報処理標準（FIPS）発行 140-2、暗号化モジュールのセキュリティ要件では、暗号化モジュールの米国政府要件が詳述されています。FIPS 140-2 では、暗号モジュールがハードウェア、ソフトウェア、ファームウェア、または何らかの組み合わせのセットで、暗号機能またはプロセスを実装し、暗号アルゴリズムおよび任意のキー生成機能を含み、明確に定義された暗号境界の内部に位置しなければならないと定義しています。

FIPS は特定の暗号アルゴリズムがセキュアであることを条件とするほか、ある暗号モジュールが FIPS 準拠であると称する場合は、どのアルゴリズムを使用すべきかも指定しています。

注意事項と制約事項

次の注意事項と制約事項に従ってください。

- FIPS を有効にすると、Cisco APIC 全体に適用されます。
- Cisco APIC ソフトウェアのダウン グレードを実行しているときに FIPS を最初に無効にする必要があります。
- パスワードは最小限 8 文字の長さで作成してください。
- Telnet をディセーブルにします。ユーザのログインは SSH だけで行ってください。
- SSH サーバの RSA1 キー ペアすべてを削除してください。
- RADIUS/TACACS+ によるリモート認証をディセーブルにしてください。ローカルと LDAP ユーザのみを認証できます。
- セキュア シェル (SSH) および SNMP がサポートされます。
- SNMP v1 および v2 をディセーブルにしてください。SNMPv3 に対して設定された、スイッチ上の既存ユーザ- アカウントのいずれについても、認証およびプライバシー用 AES3 は SHA でのみ設定されていなければなりません。
- リリース 2.3(1x) で始まる、FIPS は、スイッチのレベルで設定できます。

- リリース 3.1(1x) から始まる FIPS を有効にすると、NTP が FIPS モードの動作は、HMAC SHA1 による認証と認証なしで FIPS モード NTP をサポートしています。

NX-OS スタイル CLI を使用した Cisco APIC の FIPS 設定

FIPS を有効にすると、Cisco APIC 全体に適用されます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： <code>apicl# configure</code>	コンフィギュレーション モードに入ります。
ステップ 2	fips mode enable 例： <code>apicl(config)# fips mode enable</code>	FIP を有効にします。 no fips mode enable コマンドにより FIPS が無効になります。 設定を完了するには再起動する必要があります。モードを変更すると、設定を完了するため必ず再起動する必要があります。

コントロールプレーン ポリシングの設定

CoPP の概要

コントロールプレーン ポリシング (CoPP) はコントロールプレーンを保護し、それをデータプレーンから分離することによって、ネットワークの安定性、到達可能性、およびパケット配信を保証します。

この機能により、コントロールプロセッサに到達可能な各プロトコルに対して、パラメータの仕様でポリサーを使用したレート制限が可能になります。ポリシングは、ルータまたはレイヤ 3 スイッチの IP アドレスのいずれかを宛先とするすべてのトラフィックに適用されます。ネットワークデバイスへの一般的な攻撃ベクトルは、過剰なトラフィックがデバイスインターフェイスに転送されるサービス妨害 (DoS) 攻撃です。

Cisco ACI リーフ/スパイン NX-OS は、DoS 攻撃がパフォーマンスに影響しないようにするために CoPP を提供します。このような攻撃は誤って、または悪意を持って実行される場合があります。通常は、ACI リーフ/スパイン CPU のスーパーバイザ モジュールまたは CPU 自体に宛てられた大量のトラフィックが含まれます。

ACI リーフ/スパイン スイッチのスーパーバイザ モジュールは、管理対象のトラフィックを次の 2 つの機能コンポーネント (プレーン) に分類します。

- **データ プレーン**：すべてのデータ トラフィックを処理します。NX-OS デバイスの基本的な機能は、インターフェイス間でパケットを転送することです。スイッチ自身に向けられたものでないパケットは、中継パケットと呼ばれます。データプレーンで処理されるのはこれらのパケットです。
- **コントロール プレーン**：ルーティング プロトコルのすべての制御トラフィックを処理します。ボーダー ゲートウェイ プロトコル (BGP) や Open Shortest Path First (OSPF) プロトコルなどのルーティングプロトコルは、デバイス間で制御パケットを送信します。これらのパケットはルータのアドレスを宛先とし、コントロール プレーン パケットと呼ばれます。

ACI リーフ/スパイン スーパーバイザ モジュールはコントロール プレーンを所有しており、ネットワークの動作に重要です。スーパーバイザモジュールの動作が途絶したり、スーパーバイザモジュールが攻撃されたりすると、重大なネットワークの停止につながります。たとえばスーパーバイザに過剰なトラフィックが加わると、スーパーバイザ モジュールが過負荷になり、Cisco ACI ファブリック全体のパフォーマンスが低下する可能性があります。またたとえば、ACI リーフ/スパイン スーパーバイザ モジュールに対する DoS 攻撃は、コントロール プレーンに対して非常に高速に IP トラフィック ストリームを生成することがあります。これにより、コントロールプレーンは、これらのパケットを処理するために大量の時間を費やしてしまい、本来のトラフィックを処理できなくなります。

次に、DoS 攻撃の例を示します。

- インターネット制御メッセージプロトコル (ICMP) エコー要求
- IP フラグメント
- TCP SYN フラッドイング

これらの攻撃によりデバイスのパフォーマンスが影響を受け、次のようなマイナスの結果をもたらします。

- サービス品質の低下（音声、ビデオ、または重要なアプリケーション トラフィックの低下など）
- ルート プロセッサまたはスイッチ プロセッサの高い CPU 使用率
- ルーティングプロトコルのアップデートまたはキープアライブの消失によるルートフラップ
- メモリやバッファなどのプロセッサ リソースの枯渇
- 着信パケットの無差別のドロップ



(注) ACI リーフ/スパインは、デフォルト設定で CoPP によって保護されているデフォルトです。この機能では、顧客のニーズに基づいてノードのグループにパラメータを調整できます。

コントロールプレーン保護

コントロールプレーンを保護するために、ACI リーフ/スパインで実行している Cisco NX-OS でコントロールプレーンに向かうさまざまなパケットを異なるクラスに分離します。クラスの識別が終わると、Cisco NX-OS デバイスはパケットをポリシングします。これにより、スーパーバイザ モジュールに過剰な負担がかからないようになります。

コントロールプレーンのパケット タイプ:

コントロールプレーンには、次のような異なるタイプのパケットが到達します。

- **受信パケット**：ルータの宛先アドレスを持つパケット。宛先アドレスには、レイヤ2 アドレス（ルータ MAC アドレスなど）やレイヤ3 アドレス（ルータ インターフェイスの IP アドレスなど）があります。これらのパケットには、ルータアップデートとキープアライブ メッセージも含まれます。ルータが使用するマルチキャスト アドレス宛てに送信されるマルチキャスト パケットも、このカテゴリに入ります。
- **例外パケット**：スーパーバイザ モジュールによる特殊な処理を必要とするパケット。たとえば、宛先アドレスが Forwarding Information Base (FIB; 転送情報ベース) に存在せず、結果としてミスとなった場合は、スーパーバイザ モジュールが送信側に到達不能パケットを返します。他には、IP オプションがセットされたパケットもあります。
- **リダイレクト パケット**：スーパーバイザ モジュールにリダイレクトされるパケット。ダイナミック ホスト コンフィギュレーション プロトコル (DHCP) スヌーピングやダイナミック アドレス解決プロトコル (ARP) インスペクションなどの機能は、パケットをスーパーバイザ モジュールにリダイレクトします。
- **収集パケット**：宛先 IP アドレスのレイヤ2 MAC アドレスが FIB に存在していない場合は、スーパーバイザ モジュールがパケットを受信し、ARP 要求をそのホストに送信します。

これらのさまざまなパケットはすべて、コントロールプレーンへの悪意ある攻撃に利用され、Cisco ACI ファブリックに過剰な負荷をかける可能性があります。CoPP は、これらのパケットを異なるクラスに分類し、これらのパケットを ACI リーフ/スパイン スーパーバイザが受信する速度を個別に制御するメカニズムを提供します。

CoPP の分類：

効果的に保護するために、ACI リーフ/スパイン NX-OS はスーパーバイザ モジュールに到達するパケットを分類して、パケットタイプに基づいた異なるレート制御ポリシーを適用できるようにします。たとえば、Hello メッセージなどのプロトコル パケットには厳格さを緩め、IP オプションがセットされているためにスーパーバイザ モジュールに送信されるパケットには厳格さを強めることが考えられます。

レート制御メカニズム：

パケットの分類が終わると、ACI リーフ/スパイン NX-OS にはスーパーバイザ モジュールに到達するパケットのレートを制御するメカニズムがあります。

ポリシングには、次のパラメータを設定できます。

- **認定情報速度 (CIR)** : リンク レートをビット レートまたはパーセンテージで指定した、目的の帯域幅。
- **認定バースト (BC)** : 指定した時間枠内に CIR を超過する可能性があるが、スケジューリングには影響を与えないトラフィック バーストのサイズ。

デフォルトのポリシング ポリシー :

Cisco ACI リーフ/スパインブートアップがある場合は、プラットフォーム設定事前定義された CoPP パラメータのさまざまなプロトコルが Cisco によって実行テストに基づいています。

CoPP の注意事項と制約事項

CoPP に関する注意事項と制約事項は次のとおりです。

- 最初に strict デフォルト CoPP ポリシーを使用し、後で、データセンターおよびアプリケーションの要件に基づいて CoPP ポリシーを変更することを推奨します。
- CoPP のカスタマイズは継続的なプロセスです。CoPP を設定するときには、特定の環境で使用されるプロトコルや機能に加えて、サーバ環境に必要なスーパーバイザ機能を考慮する必要があります。これらのプロトコルや機能が変更されたら、CoPP を変更する必要があります。
- CoPP を継続的にモニタすることを推奨します。ドロップが発生した場合は、CoPP がトラフィックを誤ってドロップしたのか、または誤動作や攻撃に応答してドロップしたのかを判定してください。いずれの場合も、状況を分析し、CoPP ポリシーを変更する必要性を評価します。
- CoPP ポリシーによって、ルーティング プロトコルなどのクリティカルなトラフィック、またはデバイスへのインタラクティブなアクセスがフィルタリングされないように注意してください。このトラフィックをフィルタリングすると、Cisco ACI リーフ/スパインへのリモート アクセスが禁止され、コンソール接続が必要となる場合があります。
- APIC UI を使用して、CoPP パラメータを調整することができます。
- プロトコルごとの各インターフェイスはリーフ スイッチでのみサポートされています。
- プロトコルごとの各インターフェイスで FEX ポートはサポートされていません。
- プロトコルごとの各インターフェイスでサポートされているプロトコルは、ARP、ICMP、CDP、LLDP、LACP、BGP、STP、BFD、および OSPF です。
- プロトコルごとの各インターフェイスの最大の TCAM エントリは 256 です。しきい値を超過すると、障害が発生します。

Cisco NX-OS CLI を使用した CoPP の設定

手順

ステップ 1 CoPP リーフ プロファイルを設定します。

例 :

```
# configure copp Leaf Profile
apic1(config)# policy-map type control-plane-leaf leafProfile
apic1(config-pmap-copp-leaf)# profile-type custom
apic1(config-pmap-copp-leaf)# set arpRate 786
# create a policy group to be applied on leaves
apic1(config)# template leaf-policy-group coppForLeaves
apic1(config-leaf-policy-group)# copp-aggr leafProfile
apic1(config-leaf-policy-group)# exit
# apply the leaves policy group on leaves
apic1(config)# leaf-profile applyCopp
apic1(config-leaf-profile)# leaf-group applyCopp
apic1(config-leaf-group)# leaf 101-102
apic1(config-leaf-group)# leaf-policy-group coppForLeaves
```

ステップ 2 CoPP スパイン プロファイルを設定します。

例 :

```
# configure copp Spine Profile
apic1(config)# policy-map type control-plane-spine spineProfile
apic1(config-pmap-copp-spine)# profile-type custom
apic1(config-pmap-copp-spine)# set arpRate 786
# create a policy group to be applied on spines
apic1(config)# template leaf-policy-group coppForSpines
apic1(config-spine-policy-group)# copp-aggr spineProfile
apic1(config-spine-policy-group)# exit
# apply the spine policy group on spines
apic1(config)# spine-profile applyCopp
apic1(config-spine-profile)# spine-group applyCopp
apic1(config-spine-group)# spine 201-202
apic1(config-spine-group)# spine-policy-group coppForSpines
```

NX-OS スタイル CLI を使用するプロトコル CoPP ポリシーごとのインターフェイスごとの設定

手順

ステップ 1 CoPP クラス マップおよびポリシー マップを定義します。

例 :

```
(config)# policy-map type control-plane-if <name>
(config-pmap-copp)# protocol bgp bps <value>
(config-pmap-copp)# protocol ospf bps <value>
```

ステップ2 リーフのインターフェイスに設定を適用します。

例：

```
(config)# leaf 101
(config-leaf)# int eth 1/10
(config-leaf-if)# service-policy type control-plane-if output<name>
```

ファースト ホップ セキュリティの設定

ファースト ホップ セキュリティについて

ファーストホップセキュリティ（FHS）機能では、レイヤ2リンク上でより優れたIPv4とIPv6のリンクセキュリティおよび管理が可能になります。サービス プロバイダ環境で、これらの機能は重複アドレス検出（DAD）とアドレス解像度（AR）などのアドレス割り当てや派生操作が、より緊密に制御可能です。

次のサポートされている FHS 機能はプロトコルをセキュアにして、ファブリック リーフ スイッチにセキュアなエンドポイントデータベースを構築するのに役立ち、MIM 攻撃やIPの盗難などのセキュリティ盗難を軽減するために使用されます。

- ARP Inspection
- ND 検査
- DHCP 検査
- RA ガード
- IPv4 および Ipv6 ソース ガード
- トラスト制御

FHS 機能は、次のセキュリティ対策を提供します。

- **ロールの適用**：信頼できない主催者が、そのロールの有効範囲を超えるメッセージを送信することを防ぎます。
- **バインディングの適用**：アドレスの盗難を防止します。
- **DoS 攻撃の軽減対策**：悪意あるエンドポイントを防ぎ、データベースが操作サービスを提供することを停止するポイントにエンドポイント データベースを成長させます。
- **プロキシ サービス**：アドレス解決の効率を高めるため一部のプロキシ サービスを提供します。

FHS機能は、テナントブリッジドメイン（BD）ごとに有効になっています。ブリッジドメインとして、単一または複数のリーフ スイッチで展開可能で、FHS 脅威の制御と軽減のメカニズムは単一のスイッチと複数のスイッチのシナリオにも対応できます。

ACI FHS の導入

ほとんどの FHS 機能はツーステップ傾向で設定されています。最初に機能の動作を説明するポリシーを定義し、次にこのポリシーを「ドメイン」に適用します（テナントブリッジドメインまたはテナントエンドポイントグループになる）。異なる動作を定義する別のポリシーは、さまざまな交差ドメインに適用できます。特定のポリシーを使用する決定は、ポリシーを適用するもっとも明確なドメインで行われます。

ポリシーのオプションは、[Tenant_name]>[Networking]>[Protocol Policies]>[First Hop Security] タブの下にある Cisco APIC GUI から定義できます。

注意事項と制約事項

次の注意事項と制約事項に従ってください。

- リリース 3.1 (1) より、仮想エンドポイント（AV のみ）で FHS はサポートされています。
- EPG が VXLAN カプセル化で展開されるとき、FHS 機能はサポートされていません。
- **[ダウン]** 状態の FHS バインディング表データベースでセキュリティ保護されたエンドポイントエントリは、タイムアウトから **18 時間** 後に消去されます。エントリが学習する前面パネルポートがリンクダウンする場合、エントリは **[ダウン]** 状態に移動します。この **18 時間** ウィンドウの中で、エンドポイントが別のロケーションに移動し別のポートで確認される場合、エンドポイントが他のポートから到達可能な限り移行され、エントリはグレースフルに **[ダウン]** 状態から **[REACHABLE/STALE]** に移行します。
- IP 発信元ガードが有効な時、IP 送信元アドレスとして Ipv6 リンクローカルアドレスを使用して供給される Ipv6 トラフィックは、IP 送信元ガード施行を受けません（例：送信元 MAC の施行 <=> IP 調査機能によりセキュリティ保護された送信元 IP バインディング）。バインディングチェック障害に関係なく、デフォルトでこのトラフィックが許可されます。
- L3Out インターフェイスでは、FHS はサポートされていません。
- TOR に基づいて N9K-M12PQ では FHS はサポートされていません。
- ACI マルチサイトの FHS はサイトのローカル機能であるため、APIC クラスタからサイトでのみ有効にできます。また、ACI マルチサイトの FHS は、BD や EPG がサイトローカルであり、サイト上でストレッチしない場合にのみ動作します。ストレッチ BD または EPG の FHS セキュリティを有効にすることはできません。
- レイヤ 2 専用ブリッジドメインでは、FHS はサポートされていません。

NX-OS CLI を使用した FHS の設定

始める前に

- テナントとブリッジドメインが設定されています。

手順

ステップ1 configure

コンフィギュレーション モードに入ります。

例：

```
apicl# configure
```

ステップ2 FHS ポリシーを設定します。

例：

```
apicl(config)# tenant coke
apicl(config-tenant)# first-hop-security
apicl(config-tenant-fhs)# security-policy poll
apicl(config-tenant-fhs-secpol)#
apicl(config-tenant-fhs-secpol)# ip-inspection-admin-status enabled-both
apicl(config-tenant-fhs-secpol)# source-guard-admin-status enabled-both
apicl(config-tenant-fhs-secpol)# router-advertisement-guard-admin-status enabled
apicl(config-tenant-fhs-secpol)# router-advertisement-guard
apicl(config-tenant-fhs-raguard)#
apicl(config-tenant-fhs-raguard)# managed-config-check
apicl(config-tenant-fhs-raguard)# managed-config-flag
apicl(config-tenant-fhs-raguard)# other-config-check
apicl(config-tenant-fhs-raguard)# other-config-flag
apicl(config-tenant-fhs-raguard)# maximum-router-preference low
apicl(config-tenant-fhs-raguard)# minimum-hop-limit 10
apicl(config-tenant-fhs-raguard)# maximum-hop-limit 100
apicl(config-tenant-fhs-raguard)# exit
apicl(config-tenant-fhs-secpol)# exit
apicl(config-tenant-fhs)# trust-control tcpoll
apicl(config-tenant-fhs-trustctrl)# arp
apicl(config-tenant-fhs-trustctrl)# dhcpv4-server
apicl(config-tenant-fhs-trustctrl)# dhcpv6-server
apicl(config-tenant-fhs-trustctrl)# ipv6-router
apicl(config-tenant-fhs-trustctrl)# router-advertisement
apicl(config-tenant-fhs-trustctrl)# neighbor-discovery
apicl(config-tenant-fhs-trustctrl)# exit
apicl(config-tenant-fhs)# exit
apicl(config-tenant)# bridge-domain bd1
apicl(config-tenant-bd)# first-hop-security security-policy poll
apicl(config-tenant-bd)# exit
apicl(config-tenant)# application ap1
apicl(config-tenant-app)# epg epg1
apicl(config-tenant-app-epg)# first-hop-security trust-control tcpoll
```

ステップ3 FHS の設定例を示します。

例：

```
leaf4# show fhs bt all
```

Legend:

TR	: trusted-access	UNRES	: unresolved	Age
	: Age since creation			
UNTR	: untrusted-access	UNDTR	: undetermined-trust	CRTNG
:	creating			
UNKNW	: unknown	TENTV	: tentative	INV
:	invalid			
NDP	: Neighbor Discovery Protocol	STA	: static-authenticated	REACH

```

: reachable
  INCMP      : incomplete                VERIFY : verify                INTF
: Interface
  TimeLeft   : Remaining time since last refresh  LM      : lla-mac-match        DHCP
: dhcp-assigned

EPG-Mode:
  U : unknown    M : mac    V : vlan    I : ip

BD-VNID          BD-Vlan          BD-Name
15630220         3                t0:bd200

```

	Origin	IP	MAC	INTF	EPG(sclass) (mode)	Trust-lvl
	State	Age	TimeLeft			
ARP		192.0.200.12	D0:72:DC:A0:3D:4F	eth1/1	ep300(49154) (V)	LM,TR
STALE		00:04:49	18:08:13			
ARP		172.29.205.232	D0:72:DC:A0:3D:4F	eth1/1	ep300(49154) (V)	LM,TR
STALE		00:03:55	18:08:21			
ARP		192.0.200.21	D0:72:DC:A0:3D:4F	eth1/1	ep300(49154) (V)	LM,TR
REACH		00:03:36	00:00:02			
LOCAL		192.0.200.1	00:22:BD:F8:19:FF	vlan3	LOCAL(16387) (I)	STA
REACH		04:49:41	N/A			
LOCAL		fe80::200	00:22:BD:F8:19:FF	vlan3	LOCAL(16387) (I)	STA
REACH		04:49:40	N/A			
LOCAL		2001:0:0:200::1	00:22:BD:F8:19:FF	vlan3	LOCAL(16387) (I)	STA
REACH		04:49:39	N/A			

ステップ 4 さまざまなタイプと理由の例とともに違反を表示します。

例：

```
leaf4# show fhs violations all
```

```

Violation-Type:
  POL : policy      THR : address-theft-remote
  ROLE : role       TH  : address-theft
  INT  : internal

Violation-Reason:
  IP-MAC-TH      : ip-mac-theft                OCFG_CHK  : ra-other-cfg-check-fail
  ANC-COL        : anchor-collision
  PRF-LVL-CHK    : ra-rtr-pref-level-check-fail  INT-ERR    : internal-error
  TRUST-CHK      : trust-check-fail
  SRV-ROL-CHK    : srv-role-check-fail          ST-EP-COL  : static-ep-collision
  LCL-EP-COL     : local-ep-collision
  MAC-TH         : mac-theft                    EP-LIM     : ep-limit-reached
  MCFG-CHK       : ra-managed-cfg-check-fail
  HOP-LMT-CHK    : ra-hoplimit-check-fail       MOV-COL    : competing-move-collision
  RTR-ROL-CHK    : rtr-role-check-fail
  IP-TH          : ip-theft

EPG-Mode:
  U : unknown    M : mac    V : vlan    I : ip

BD-VNID          BD-Vlan          BD-Name
15630220         3                t0:bd200

```

	Type	Last-Reason	Proto	IP	MAC	Port
	EPG(sclass) (mode)	Count				
THR	IP-TH		ARP	192.0.200.21	D0:72:DC:A0:3D:4F	tunnel5

```
epg300(49154) (V) | 21 |
```

```
-----
```

Table Count: 1

ステップ5 FHS 設定の表示:

例:

```
swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security binding-table
```

Pod/Node	Type	Family	IP Address	MAC Address	Interface	Level
State						
-----	-----	-----	-----	-----	-----	-----
1/102	local reach	ipv4	192.0.200.1	00:22:BD:F8:19:FF	vlan3	static-
authenticated		able				
1/102	local reach	ipv6	fe80::200	00:22:BD:F8:19:FF	vlan3	static-
authenticated		able				
1/102	local reach	ipv6	2001:0:0:200::1	00:22:BD:F8:19:FF	vlan3	static-
authenticated		able				
1/101	arp	ipv4	192.0.200.23	D0:72:DC:A0:02:61	eth1/2	
lla-mac-match		stale				
,untrusted-						
1/101	local reach	ipv4	192.0.200.1	00:22:BD:F8:19:FF	vlan3	access static-
authenticated		able				
1/101	nd	ipv6	fe80::d272:dcff:fea0	D0:72:DC:A0:02:61	eth1/2	
lla-mac-match		reach				
			:261			
,untrusted-						
1/101	nd	ipv6	2001:0:0:200::20	D0:72:DC:A0:02:61	eth1/2	access
lla-mac-match		stale				
,untrusted-						
1/101	nd	ipv6	2001::200:d272:dcff:	D0:72:DC:A0:02:61	eth1/2	access
lla-mac-match		stale				
			fea0:261			
,untrusted-						
1/101	local reach	ipv6	fe80::200	00:22:BD:F8:19:FF	vlan3	access static-
authenticated		able				
1/101	local reach	ipv6	2001:0:0:200::1	00:22:BD:F8:19:FF	vlan3	static-
authenticated		able				
1/103	local reach	ipv4	192.0.200.1	00:22:BD:F8:19:FF	vlan4	static-
authenticated		able				
1/103	local reach	ipv6	fe80::200	00:22:BD:F8:19:FF	vlan4	static-

```

authenticated    able
1/103    local    ipv6    2001:0:0:200::1    00:22:BD:F8:19:FF    vlan4    static-
reach

authenticated    able
1/104    arp      ipv4    192.0.200.10    F8:72:EA:AD:C4:7C    eth1/1
lla-mac-match    stale

,trusted-access
1/104    arp      ipv4    172.29.207.222    D0:72:DC:A0:3D:4C    eth1/1
lla-mac-match    stale

,trusted-access
1/104    local    ipv4    192.0.200.1    00:22:BD:F8:19:FF    vlan4    static-
reach

authenticated    able
1/104    nd      ipv6    fe80::fa72:eaff:fead    F8:72:EA:AD:C4:7C    eth1/1
lla-mac-match    stale
:c47c

,trusted-access
1/104    nd      ipv6    2001:0:0:200::10    F8:72:EA:AD:C4:7C    eth1/1
lla-mac-match    stale

,trusted-access
1/104    local    ipv6    fe80::200    00:22:BD:F8:19:FF    vlan4    static-
reach

authenticated    able
1/104    local    ipv6    2001:0:0:200::1    00:22:BD:F8:19:FF    vlan4    static-
reach

authenticated    able

```

Pod/Node	Type	IP Address	Creation TS	Last Refresh TS
		Lease Period		
1/102	local	192.0.200.1	2017-07-20T04:22:38.000+00:00	
			2017-07-20T04:22:38.000+00:00	
1/102	local	fe80::200	2017-07-20T04:22:56.000+00:00	
			2017-07-20T04:22:56.000+00:00	
1/102	local	2001:0:0:200::1	2017-07-20T04:22:57.000+00:00	
			2017-07-20T04:22:57.000+00:00	
1/101	arp	192.0.200.23	2017-07-27T10:55:20.000+00:00	
			2017-07-27T16:07:24.000+00:00	
1/101	local	192.0.200.1	2017-07-27T10:48:09.000+00:00	
			2017-07-27T10:48:09.000+00:00	
1/101	nd	fe80::d272:dcff:fea0	2017-07-27T10:52:16.000+00:00	
			2017-07-27T16:04:29.000+00:00	
		:261		
1/101	nd	2001:0:0:200::20	2017-07-27T10:57:32.000+00:00	
			2017-07-27T16:07:24.000+00:00	
1/101	nd	2001::200:d272:dcff:	2017-07-27T11:21:45.000+00:00	
			2017-07-27T16:07:24.000+00:00	
		fea0:261		
1/101	local	fe80::200	2017-07-27T10:48:10.000+00:00	
			2017-07-27T10:48:10.000+00:00	
1/101	local	2001:0:0:200::1	2017-07-27T10:48:11.000+00:00	
			2017-07-27T10:48:11.000+00:00	
1/103	local	192.0.200.1	2017-07-26T22:03:56.000+00:00	
			2017-07-26T22:03:56.000+00:00	


```

1/103      local    fe80::200              2017-07-26T22:03:57.000+00:00
2017-07-26T22:03:57.000+00:00
1/103      local    2001:0:0:200::1        2017-07-26T22:03:58.000+00:00
2017-07-26T22:03:58.000+00:00
1/104      arp      192.0.200.10              2017-07-27T11:21:13.000+00:00
2017-07-27T16:05:48.000+00:00
1/104      arp      172.29.207.222           2017-07-27T11:54:48.000+00:00
2017-07-27T16:06:38.000+00:00
1/104      local    192.0.200.1              2017-07-27T10:49:13.000+00:00
2017-07-27T10:49:13.000+00:00
1/104      nd        fe80::fa72:eaff:fead    2017-07-27T11:21:13.000+00:00
2017-07-27T16:06:43.000+00:00
                                :c47c
1/104      nd        2001:0:0:200::10      2017-07-27T11:21:13.000+00:00
2017-07-27T16:06:19.000+00:00
1/104      local    fe80::200              2017-07-27T10:49:14.000+00:00
2017-07-27T10:49:14.000+00:00
1/104      local    2001:0:0:200::1        2017-07-27T10:49:15.000+00:00
2017-07-27T10:49:15.000+00:00

swtb23-ifc1#

swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security statistics arp
Pod/Node      : 1/101
Request Received : 4
Request Switched : 2
Request Dropped : 2
Reply Received  : 257
Reply Switched  : 257
Reply Dropped   : 0

Pod/Node      : 1/104
Request Received : 6
Request Switched : 6
Request Dropped : 0
Reply Received  : 954
Reply Switched  : 954
Reply Dropped   : 0

swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security statistics dhcpv4
Pod/Node      : 1/102
Discovery Received : 5
Discovery Switched : 5
Discovery Dropped : 0
Offer Received    : 0
Offer Switched    : 0
Offer Dropped     : 0
Request Received   : 0
Request Switched   : 0
Request Dropped    : 0
Ack Received       : 0
Ack Switched       : 0
Ack Dropped        : 0
Nack Received      : 0
Nack Switched      : 0
Nack Dropped       : 0
Decline Received   : 0
Decline Switched   : 0
Decline Dropped    : 0
Release Received   : 0
Release Switched   : 0
Release Dropped    : 0
Information Received : 0

```

```

Information Switched      : 0
Information Dropped       : 0
Lease Query Received      : 0
Lease Query Switched      : 0
Lease Query Dropped       : 0
Lease Active Received     : 0
Lease Active Switched     : 0
Lease Active Dropped      : 0
Lease Unassignment Received : 0
Lease Unassignment Switched : 0
Lease Unassignment Dropped : 0
Lease Unknown Received    : 0
Lease Unknown Switched    : 0
Lease Unknown Dropped     : 0

```

```

swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security statistics
neighbor-discovery

```

```

Pod/Node                  : 1/101
Neighbor Solicitation Received : 125
Neighbor Solicitation Switched : 121
Neighbor Solicitation Dropped : 4
Neighbor Advertisement Received : 519
Neighbor Advertisement Switched : 519
Neighbor Advertisement Drop : 0
Router Solicitation Received : 4
Router Solicitation Switched : 4
Router Solicitation Dropped : 0
Router Adv Received : 0
Router Adv Switched : 0
Router Adv Dropped : 0
Redirect Received : 0
Redirect Switched : 0
Redirect Dropped : 0

```

```

Pod/Node                  : 1/104
Neighbor Solicitation Received : 123
Neighbor Solicitation Switched : 47
Neighbor Solicitation Dropped : 76
Neighbor Advertisement Received : 252
Neighbor Advertisement Switched : 228
Neighbor Advertisement Drop : 24
Router Solicitation Received : 0
Router Solicitation Switched : 0
Router Solicitation Dropped : 0
Router Adv Received : 53
Router Adv Switched : 6
Router Adv Dropped : 47
Redirect Received : 0
Redirect Switched : 0
Redirect Dropped : 0

```

802.1X の設定

802.1X の概要

802.1X では、クライアント サーバベースのアクセス コントロールと認証プロトコルを定義し、許可されていないクライアントが公にアクセス可能なポートを経由して LAN に接続するのを規制します。認証サーバは、Cisco NX-OS デバイスのポートに接続されるクライアントを個々に認証します。

802.1X アクセス コントロールでは、クライアントが認証されるまで、そのクライアントが接続しているポート経由では Extensible Authentication Protocol over LAN (EAPOL) トラフィックしか許可されません。認証に成功すると、通常のトラフィックはポートを通過できるようになります。

RADIUS 分散クライアント/サーバ システムを使用すると、不正アクセスからネットワークを保護できます。Cisco ACI 実装では、RADIUS クライアントは ToR で稼働し、すべてのユーザー認証情報およびネットワーク サービス アクセス情報が格納された中央の RADIUS サーバに認証要求およびアカウントिंग要求を送信します。

ホスト サポート

802.1X 機能は、次のモードでポート上のトラフィックを制限できます。

- **単一ホストモード**：802.1X ポートで1台のエンドポイントデバイスのみからのトラフィックが許可されます。エンドポイント デバイスが認証されると、APIC はポートを許可状態にします。エンドポイント デバイスがログオフすると、OS はポートを無許可状態に戻します。802.1X のセキュリティ違反とは、認証に成功して許可された単一の MAC アドレスとは異なる MAC アドレスをソースとするフレームが検出された場合をいいます。このような場合、このセキュリティ アソシエーション (SA) 違反 (他の MAC アドレスからの EAPOL フレーム) が検出されたインターフェイスはディセーブルにされます。シングルホストモードは、ホストツースイッチ型トポロジで1台のホストが APIC のレイヤ2 ポート (イーサネット アクセス ポート) またはレイヤ3 ポート (ルーテッド ポート) に接続されている場合にだけ適用できます。
- **複数のホストモード**：ポートごとに複数のホストを使用できますが、最初の1つだけが認証されます。最初のホストの許可に成功すると、ポートは許可ステートに移行します。ポートが許可ステートになると、後続のホストがネットワーク アクセスの許可を受ける必要はありません。再認証に失敗したり、または EAPOL ログオフ メッセージを受信して、ポートが無許可ステートになった場合には、接続しているすべてのクライアントはネットワーク アクセスを拒否されます。マルチホストモードでは、SA 違反の発生時にインターフェイスをシャットダウンする機能がディセーブルになります。このモードは、スイッチ ツースイッチ型トポロジおよびホストツースイッチ型トポロジの両方に適用できます
- **マルチ認証モード**：複数のホストとすべてのホストを個別に認証を使用できます。



(注) 各ホストには、同じ EPG/VLAN 情報を必須です。

- **マルチ ドメイン モード** : 別のデータおよび音声ドメイン。IP 電話で使用します。

認証モード

ACI 802.1 x は次の認証モードをサポートしています。

- **EAP** : オーセンティケータは EAP-Request/Identity フレームをサブリカントに送信して識別情報を要求します（通常、オーセンティケータは1つまたは複数の識別情報の要求のあとに、最初の Identity/Request フレームを送信します）。サブリカントはフレームを受信すると、EAP-Response/Identity フレームで応答します。
- **MAB** : フォールバック認証モードとして MAC 認証バイパス (MAB) がサポートされています。MAB により、エンドポイントの MAC アドレスを使用してポートベースのアクセス コントロールが有効になります。MAB が有効なポートは接続するデバイスの MAC アドレスに基づいて、動的に有効または無効にできます。MAB の前に、エンドポイントの ID が不明であり、すべてのトラフィックがブロックされます。スイッチでは、単一のパケットを検査して送信元 MAC アドレスを学習および認証します。MAB が成功するとエンドポイントの ID が判明し、エンドポイントからのすべてのトラフィックが許可されます。スイッチは送信元 MAC アドレス フィルタリングを実行し、MAB の認証されたエンドポイントのみがトラフィックの送信を許可されます。

注意事項と制約事項

802.1X ポートベースの認証には、次の設定に関する注意事項と制約事項があります。

- Cisco ACI が 802.1X 認証をサポートするのは、物理ポート上だけです。
- Cisco ACI は、ポート チャネルまたはサブインターフェイスでは 802.1X 認証をサポートしません。
- Cisco ACI は、ポート チャネルのメンバポートでは 802.1X 認証をサポートしますが、ポート チャネル自体ではサポートしません。
- 802.1X 設定を含むメンバポートと含まないメンバポートはポート チャネルで共存できます。ただし、チャネリングと 802.1X が連携して動作するためには、すべてのメンバポートで 802.1X 設定を同一にする必要があります。
- 802.1X 認証を有効にした場合、サブリカントが認証されてから、イーサネット インターフェイス上のレイヤ 2 またはレイヤ 3 のすべての機能が有効になります。
- 802.1 x はリーフ シャーシでのみサポート EX または FX タイプことです。
- 802.1 x はサポートされているファブリック アクセス ポートだけです。802.1X はポート チャネルまたは仮想ポート チャネルではサポートされていません。

- IPv6 は、dot1x クライアント 3.2(1) リリースではサポートされていません。
- 特に特定のインターフェイス設定（ホストモードおよび認証タイプ）がそのリリースでサポートされていない場合に以前のリリースにダウングレードすると、dot1x 認証タイプはデフォルトでなしになります。ホストモードは希望に応じて単一のホストか複数のホストのどちらかに手動で再設定する必要があります。これで、ユーザーがそのリリースでのみサポートされているモード/認証タイプを設定し、サポートされていないシナリオで実行していないことを確認します。
- マルチ認証では、1 音声クライアントと複数のデータ クライアント（すべて同じデータ vlan/epg に属する）をサポートします。
- 障害 epg/vlan で 802.1x 認証ポリシーのノードでは、必須の設定を示します。
- マルチドメインの方法より 1 の音声およびデータが 1 つのクライアントは、セキュリティが無効な状態で、ポートを配置します。
- 802.1 の次のプラットフォームはサポートされていません x:
 - N9K-C9396PX
 - N9K-M12PQ
 - N9K-C93128TX
 - N9K-M12PQ

コンフィギュレーションの概要

APIC で有効になっている場合にのみ、802.1 x および RADIUS プロセスが開始されます。内部的にこれは、radius エンティティの作成時に 802.1x Inst MO が作成され radius プロセスが作成されたときに、dot1x プロセスが開始されることを意味します。そのインターフェイスに接続しているユーザーを認証するため、Dot1x ベースの認証が各インターフェイスで有効になる必要があります。そうでない場合、動作が変更されません。

RADIUS サーバの設定は、dot1x 設定とは別に行われます。RADIUS の設定は、RADIUS サーバのリストとそれらに到達する方法を定義します。Dot1x 設定には、認証に使用する RADIUS グループ（またはデフォルト グループ）への参照が含まれています。

802.1 x および RADIUS の設定の両方ともが正常に認証が行われる必要があります。設定の順序は重要ではありませんが、RADIUS 設定がない場合は、802.1 x 認証は正常に行われません。

NX-OS スタイル CLI を使用した 802.1 x ノード認証の設定

手順

ステップ 1 Radius 認証グループを設定します。

例：

```

apic1# configure
apic1(config)#
apic1(config)# aaa group server radius myradiusgrp
apic1(config-radius)#server 192.168.0.100 priority 1
apic1(config-radius)#exit

```

ステップ 2 ノード レベル ポート認証ポリシーを設定します。

例 :

```

apic1(config)# policy-map type port-authentication mydot1x
apic1(config-pmap-port-authentication)#radius-provider-group myradiusgrp
apic1(config-pmap-port-authentication)#fail-auth-vlan 2001
apic1(config-pmap-port-authentication)#fail-auth-epg tenant tn1 application ap1 epg
epg256
apic1(config)# exit

```

ステップ 3 ポリシー グループを設定し、グループ内でポート認証ポリシーを指定します。

例 :

```

apic1(config)#template leaf-policy-group lpg2
apic1(config-leaf-policy-group)# port-authentication mydot1x
apic1(config-leaf-policy-group)#exit

```

ステップ 4 リーフ スイッチ プロファイルを設定します。

例 :

```

apic1(config)# leaf-profile mylp2
apic1(config-leaf-profile)#leaf-group mylg2
apic1(config-leaf-group)# leaf-policy-group lpg2
apic1(config-leaf-group)#exit

```

NX-OS スタイル CLI を使用した 802.1x ポート認証の設定

手順

ステップ 1 ポリシー グループを設定します。

例 :

```

apic1# configure
apic1(config)#
apic1(config)# template policy-group mypol
apic1(config-pol-grp-if)# switchport port-authentication mydot1x
apic1(config-port-authentication)# host-mode multi-host
apic1(config-port-authentication)# no shutdown
apic1(config-port-authentication)# exit
apic1(config-pol-grp-if)# exit

```

ステップ 2 リーフ インターフェイス ポリシーを設定します。

例 :

```
apicl(config)#
apicl(config)#leaf-interface-profile myprofile
apicl(config-leaf-if-profile)#leaf-interface-group mygroup
apicl(config-leaf-if-group)# interface ethernet 1/10-12
apicl(config-leaf-if-group)# policy-group mypol
apicl(config-leaf-if-group)# exit
apicl(config-leaf-if-profile)# exit
```

ステップ3 リーフ プロファイルを設定します。

例：

```
apicl(config)#
apicl(config)# leaf-profile myleafprofile
apicl(config-leaf-profile)# leaf-group myleafgrp
apicl(config-leaf-group)# leaf 101
apicl(config-leaf-group)# exit
```

ステップ4 リーフ スイッチ プロファイルにインターフェイス ポリシーを適用します。

例：

```
apicl(config-leaf-profile)# leaf-interface-profile myprofile
apicl(config-leaf-group)# exit
```



第 11 章

エニーキャスト サービスの設定

この章は、次の項で構成されています。

- [エニーキャスト サービスについて \(441 ページ\)](#)
- [NX-OS スタイル CLI を使用したエニーキャスト サービスの設定 \(443 ページ\)](#)

エニーキャスト サービスについて

エニーキャスト サービスは、Cisco ACI ファブリックでサポートされます。Typical use(典型的な事例、一般的な使用)ケースは、multipod ファブリックのポッドの Cisco 適応型セキュリティ アプライアンス (ASA) ファイアウォールをサポートするためにも、他の DNS サーバなどのサービスまたは印刷サービスを有効にするエニーキャストを使用することもできます。ASA で使用例、ポッドごとに、ファイアウォールがインストールされているおよびエニーキャスト サービスとして、ファイアウォールを提供できるように、エニーキャストが有効にします。下方向にファイアウォールの1つのインスタンスでは、インスタンスが使用可能な最も近い、次に、要求がルーティングされるよう、クライアントは影響しません。エニーキャストを有効にする各ポッドで ASA ファイアウォールをインストールし、IP アドレスと使用される MAC アドレスを設定します。

エニーキャストは、(たとえば、N9K-C93180LC-EX) EX で終了する名前と Cisco Nexus 9000 シリーズ スイッチでサポートされている以降です。

エニーキャスト設定できますアプリケーション Epg でまたはレイヤ 7 サービスにレイヤ 4 (付きまたはポリシー ベース リダイレクト (PBR) なし)。

最大 2000年エニーキャスト サービスは、ファブリックあたりサポートされます。

サービス ノードは、ポリシーが適用される場所ポッドでエニーキャスト サービスで使用されます。

APIC は、VRF が展開されている、またはエニーキャスト EPG を許可する契約がリーフ スイッチにエニーキャスト MAC および IP アドレスの設定を展開します。

最初に、各リーフ スイッチは、スパイン スイッチへのプロキシルートとしてエニーキャスト MAC および IP アドレスをインストールします。エニーキャスト サービスから最初のパケットを受信すると、サービスがインストールされているの背後にあるリーフ スイッチで、サービスの接続先の情報がインストールされます。その他のすべてのリーフ スイッチは、スパイン

プロキシを指すように進みます。学習すると、エニー キャスト サービスがされて、ポッドのリーフの背後にある COOP ポッドへのローカル サービスを指すスパイン スイッチで、エントリをインストールします。

エニー キャスト サービスは、1つのポッドで実行されている、スパインは BGP EVPN を介してポッドに存在するエニー キャスト サービスのルート情報を受信します。エニー キャスト サービスがすでにある場合 COOP リモート ポッドのエニー キャスト サービス情報のキャッシュをローカルに存在します。ローカル サービスのインスタンスがダウンすると、リモートポッドを介してこのルートはのみインストールされます。

エニー キャスト サービスは、次の機能とオプションではサポートされていません。

- Multi-Site management
- リモートのリーフ スイッチ
- アクティブ/スタンバイ リレーションシップ内の 2つのファイアウォール (このシナリオでは、エニー キャスト サービスがアクティブで 1つだけのポッドとアクティブなサービスを使用して、すべてのトラフィックが送信される)
- 2つのポート チャンネル (Pc) に導入されるファイアウォール
- 冗長リンクの 1 台の PC に導入されるファイアウォール
- ECMP
- 対称ポリシー ベースのリダイレクト
- ポッドの ID 認識リダイレクション
- IP SLA Monitoring Policies
- リダイレクト ヘルス グループ
- DAD エニー キャスト IPv6 アドレスを使用するの外部デバイスで有効になっています。
- リモート IP アドレス学習の IP アドレスを防ぐために、サービスのインスタンスを通過するエニー キャスト サービス MAC および IP アドレスのリモートの学習がオフです。
- L3Outs の背後にあるエニー キャスト サービス
- エニー キャスト アドレスと、既存の静的なエンドポイントの MAC および IP アドレスを使用します。



(注) 既存の静的なエンドポイントのアドレスを使用して、エニー キャスト MAC および IP アドレスを設定すると設定は、スイッチに、APICからプッシュし、障害を生成できませんが、スイッチでは、ハードウェアにエニー キャスト アドレスをインストールしません。スタティック エンドポイントを削除すると、問題が解決しないかもしれません。スタティックのエンドポイントとエニー キャスト設定の両方を削除し、エニー キャスト アドレスを再設定する必要があります。

NX-OS スタイル CLI を使用したエニーキャスト サービスの設定

次の例では、NX-OS スタイル CLI を使用して、3 つの方法でエニーキャスト サービスを設定する方法を示します。

- EPG の背後
- ポリシーベース リダイレクト (PBR) とレイヤ 4 ～レイヤ 7 サービス グラフの一部
- レイヤ 7 サービス グラフ PBR せずにレイヤ 4 の一部として

始める前に

- テナント、アプリケーションプロファイル、アプリケーション EPG が作成されています。
- ノードグループ ポリシーと L3Out ポリシーがすでに作成されています。
- Interpod Network (IPN) がすでに設定されています。
- マルチポッドが設定されています。
- 各ポッドで、IPN への接続に使用されるスパイン スイッチは、少なくとも 1 個のリーフ スイッチに接続します。
- ASA ファイアウォールは、各ポッドにインストールされます。

手順

ステップ 1 次のコマンドを使用して、EPG サブネットの背後にあるエニーキャスト サービスを設定します。

a) **configure**

コンフィギュレーション モードに入ります。

例 :

```
apic1# configure
```

b) **tenant *tenant-name***

Creates a tenant if it does not exist or enters tenant configuration mode.

例 :

```
apic1(config)# tenant anycast1-it
```

c) **application *app-name***

Creates an application profile if it doesn't exist and enters application profile configuration mode.

例 :

```
apic1(config-tenant)# application AP0
```

d) **epg *epg-name***

EPG configuration mode(設定モード、コンフィギュレーションモード)を開始し、存在しない場合は、EPG を作成します。

例 :

```
apic1(config-tenant-app)# epg epg1
```

e) **endpointip *ip-address*anycast *mac-address***

ネットマスクおよびEPGの背後にあるエニーキャストサービスのMACアドレスでエニーキャストのIPアドレスを設定します。エニーキャストサブネットは、/32 または/128 必須ネットマスク。

例 :

```
apic1(config-tenant-app-epg)# endpoint ip 1.2.3.4/32 anycast 00:11:22:33:44:55
```

ステップ 2 次のコマンドを使用して、PBR をレイヤ 7 サービスにレイヤ 4 のエニーキャストを設定します。

a) **configure**

コンフィギュレーションモードに入ります。

例 :

```
apic1# configure
```

b) **tenant *name***

Creates a tenant if it does not exist or enters tenant configuration mode.

```
apic1(config)# tenant t1
```

c) **svcredir-pol *name***

サービスリダイレクトポリシーモードを開始し、サービスリダイレクトポリシーを作成します。

例 :

```
apic1(config-tenant)# svcredir-pol N1Ext
```

d) **anycast enable**

サービスのリダイレクトポリシーのエニーキャストを有効にします。Use the no form of the command to disable Anycast for the policy.

例 :

```
apic1(svcredir-pol)# anycast enable
```

e) **redir-dest *ip-addr* *mac-addr***

レイヤ 4～レイヤ 7 サービスリダイレクトポリシーのエニーキャスト IP および MAC アドレスを定義します。

例 :

```
apic1(svcredir-pol)# redir-dest 2000::25 00:00:00:00:00:07
```

ステップ 3 レイヤ 7 サービス PBR、なしで、次のコマンドにレイヤ 4 のエニーキャストを設定します。

a) **configure**

コンフィギュレーション モードに入ります。

例 :

```
apicl# configure
```

b) **tenant name**

Creates a tenant if it does not exist or enters tenant configuration mode.

```
apicl(config)# tenant t1
```

c) **l4l7 graph connector-name contract name**

コントラクトに関連付けられたレイヤ 4～レイヤ 7 サービスグラフを作成します。

例 :

```
apicl(config-tenant)# l4l7 graph WebGraph contract default
```

d) **service device-cluster-name**

デバイスのクラスとサービスを定義します。

例 :

```
apicl(config-graph)# service N1
```

e) **connector name [cluster-interface cluster-interface-name]**

コネクタ configuration mode(設定モード、コンフィギュレーションモード)を開始し、デバイスのクラス インターフェイスを定義します。

例 :

```
apicl(config-service)# connector provider
```

f) **subnet-ip IP-addr_with_netmask subnet-ctrl no-default-gateway**

エニーキャストの IP アドレスの定義 (/32 とネットマスクおよびサブネット制御、no デフォルト ゲートウェイ)。To remove it, use the no form of the command.

例 :

```
apicl(config-connector)# subnet-ip 50.50.50.50/32 subnet-ctrl no-default-gateway
```

g) **mac-address mac-address**

エニーキャストの MAC アドレスを定義します。To remove it, use the no form of the command.

例 :

```
apicl(config-subnet-ip)# mac-address 00.00.00.00.00.50
```

例

次の例では、EPG1 の背後にあるエニーキャスト サービスで設定します。

```
apic1# configure
apic1(config)# tenant anycast1-it
apic1(config-tenant)# application AP0
apic1(config-tenant-app)# epq epq-1
apic1(config-tenant-app-epq)# endpoint ip 1.2.3.4/32 anycast 00:11:22:33:44:55
```

次の例では、レイヤ7サービス リダイレクトのポリシーにレイヤ4でエニーキャスト サービスで設定します。

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# svcredirect-pol N1Ext
apic1(svcredirect-pol)# anycast enable
apic1(svcredirect-pol)# redirect-dest 2000::25 00:00:00:00:00:07
```

次の例では、レイヤ7サービス PBR せずにレイヤ4でエニーキャスト サービスで設定します。

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# l4l7 graph WebGraph contract default
apic1(config-graph)# service N1
apic1(config-service)# connector provider
apic1(config-connector)# subnet-ip 50.50.50.50/32 subnet-ctrl no-default-gateway
apic1(config-subnet-ip)# mac-address 00.00.00.00.00.50
```



第 12 章

VMM の設定

For information about configuring virtual machine management using the NX-OS style CLI for APIC, see the 『*Cisco ACI Virtualization Guide*』 .



第 13 章

Layer 4 - Layer 7 サービスの設定

For information about configuring Layer 4 - Layer 7 services using the NX-OS style CLI for APIC, see the 『Cisco APIC レイヤ 4 ～ レイヤ 7 サービス導入ガイド』.



第 14 章

グローバル ポリシーの設定

- グローバル ポリシーについて (451 ページ)
- アウトオブバンド管理 NTP の設定 (452 ページ)
- システム クロックの設定 (454 ページ)
- エラー ディセーブル回復の設定 (455 ページ)
- リンク レベル検出プロトコルの設定 (456 ページ)
- Miscabling プロトコルの設定 (457 ページ)
- エンドポイントのループ防止ポリシーの設定 (459 ページ)
- 不正なエンドポイントの制御ポリシーの設定 (460 ページ)
- IP エージングの設定 (462 ページ)
- ダイナミック ロード バランサの設定 (463 ページ)
- スパニングツリー プロトコルの設定 (464 ページ)
- 『Configuring IS-IS』 (466 ページ)
- BGP ルート リフレクタの設定 (469 ページ)
- ノードのデコミッション (470 ページ)
- 電力管理の設定 (471 ページ)
- スケジューラの設定 (472 ページ)
- システム MTU の設定 (475 ページ)
- PTP について (475 ページ)

グローバル ポリシーについて

APIC ファブリックには多くのファブリック レベル構成があり、その構成はファブリック コンポーネント全体に適用されます (スイッチおよびポート)。状況によっては、低レベルのポリシー (スイッチまたはインターフェイス レベル) が、これらのポリシーを上書きするために存在します。たとえば、MCP ポリシーは、MCP 機能をグローバルに有効にすることができますが、インターフェイス レベルの MCP ポリシーは、個々のインターフェイスの MCP を有効または無効にするために存在します。

アウトオブバンド管理 NTP の設定

ACI ファブリックをアウトオブバンド管理で展開する場合、ファブリックの各ノードは ACI ファブリックの外部から管理されます。アウトオブバンド管理の NTP サーバを設定すると、各ノードは一貫したクロックソースとして同じ NTP サーバに個々に照会することができます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	コンフィギュレーションモードに入ります。
ステップ 2	template ntp-fabric <i>ntp-fabric-template-name</i> 例 : <code>apic1(config)# template ntp-fabric poll</code>	ファブリックの NTP テンプレート (ポリシー) を指定します。
ステップ 3	[no] server dns-name-or-ipaddress [prefer] [use-vrf {inband-mgmt oob-default}] [key key-value] 例 : <code>apic1(config-template-ntp-fabric)# server 192.0.20.123 prefer use-vrf oob-mgmt</code>	アクティブ NTP ポリシーの NTP サーバを設定します。このサーバをアクティブ NTP ポリシーの優先サーバにするには、 prefer キーワードを含めます。NTP 認証が有効になっている場合は、参照キー ID を指定します。To specify the in-band or out-of-band management access VRF, include the use-vrf keyword with the inb-default or oob-default keyword.
ステップ 4	[no] authenticate 例 : <code>apic1(config-template-ntp-fabric)# no authenticate</code>	NTP 認証を有効または無効にします。
ステップ 5	[no] authentication-key key-value 例 : <code>apic1(config-template-ntp-fabric)# authentication-key 12345</code>	認証 NTP 認証を設定します。指定できる範囲は 1 ～ 65535 です。
ステップ 6	[no] trusted-key key-value 例 : <code>apic1(config-template-ntp-fabric)# trusted-key 54321</code>	信頼 NTP 認証を設定します。指定できる範囲は 1 ～ 65535 です。

	コマンドまたはアクション	目的
ステップ 7	exit 例 : apicl(config-template-ntp-fabric) # exit	グローバル コンフィギュレーション モードに戻ります。
ステップ 8	template pod-group <i>pod-group-template-name</i> 例 : apicl(config) # template pod-group allPods	ポッドグループ テンプレート (ポリシー) を設定します。
ステップ 9	inherit ntp-fabric <i>ntp-fabric-template-name</i> 例 : apicl(config-pod-group) # inherit ntp-fabric poll	事前に設定した NTP ファブリック テンプレート (ポリシー) を使用するよう に NTP ファブリック のポッドグループ を設定します。
ステップ 10	exit 例 : apicl(config-template-pod-group) # exit	グローバル コンフィギュレーション モードに戻ります。
ステップ 11	pod-profile <i>pod-profile-name</i> 例 : apicl(config) # pod-profile all	ポッド プロファイルを設定します。
ステップ 12	pods { <i>pod-range-1-255</i> all } 例 : apicl(config-pod-profile) # pods all	一連のポッドを設定します。
ステップ 13	inherit pod-group <i>pod-group-name</i> 例 : apicl(config-pod-profile-pods) # inherit pod-group allPods	事前に設定したポッドグループにポッドプロファイルを関連付けます。
ステップ 14	end 例 : apicl(config-pod-profile-pods) # end	EXEC モードに戻ります。

例

次に、優先アウトオブバンドNTPサーバを設定し、その設定および展開を確認する例を示します。

```
apicl# configure t
apicl(config) # template ntp-fabric poll
```

```

apic1(config-template-ntp-fabric)# server 192.0.20.123 use-vrf oob-default
apic1(config-template-ntp-fabric)# no authenticate
apic1(config-template-ntp-fabric)# authentication-key 12345
apic1(config-template-ntp-fabric)# trusted-key 12345
apic1(config-template-ntp-fabric)# exit
apic1(config)# template pod-group allPods
apic1(config-pod-group)# inherit ntp-fabric poll
apic1(config-pod-group)# exit
apic1(config)# pod-profile all
apic1(config-pod-profile)# pods all
apic1(config-pod-profile-pods)# inherit pod-group allPods
apic1(config-pod-profile-pods)# end
apic1#

```

```

apic1# show ntpq
nodeid      remote      refid      st      t      when      poll      reach      delay      offset      jitter
-----  -  -----  -----  ----  --  -----  -----  -----  -----  -----
1          *  192.0.20.123  .GPS.          u      27        64        377        76.427      0.087      0.067
2          *  192.0.20.123  .GPS.          u       3        64        377        75.932      0.001      0.021
3          *  192.0.20.123  .GPS.          u       3        64        377        75.932      0.001      0.021

```

システムクロックの設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] clock display-format {local utc} 例： apic1(config)# clock display-format local	クロックの日付と時刻の形式をローカルまたは UTC 時間のいずれかに設定します。
ステップ 3	[no] clock show-offset enable 例： apic1(config)# clock show-offset enable	UTCからのオフセットの表示を有効（または無効）にします。この設定は、display-format が local の場合にのみ有効です。
ステップ 4	[no] clock timezone timezone-code 例：	ローカル タイムゾーンを指定します。デフォルトは p0_utc です。

	コマンドまたはアクション	目的
	<code>apic1(config)# clock timezone n420_America-Los_Angeles</code>	
ステップ 5	show clock 例 : <code>apic1(config)# show clock</code>	任意のインターフェイス上で LLDP を初期化する際の遅延時間を指定します。指定できる範囲は 1 ～ 10 秒です。デフォルトは 2 秒です。

例

次に、ロサンゼルス タイムゾーンのローカル時刻にシステムクロックを設定する例を示します。

```
apic1# configure terminal
apic1(config)# clock display-format local
apic1(config)# clock show-offset enable
apic1(config)# clock timezone n420_America-Los_Angeles
apic1(config)# show clock
Time : 20:47:37.038 UTC-08:00 Sun Nov 08 2015
```

エラー ディセーブル回復の設定

エラーディセーブル回復（EDR）ポリシーは、ループ検出ポートをイネーブルにすることができるファブリックレベルのポリシーおよび、管理者が設定することができる間隔が経過した後でディセーブルになる BPDU ポリシーです。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] errdisable recovery interval seconds 例 : <code>apic1(config)# errdisable recovery interval 300</code>	インターフェイスが error-disabled ステートから回復する間隔を指定します。有効な範囲は 30 ～ 65535 秒です
ステップ 3	[no] errdisable recovery cause {bpduguard ep-move mcp-loop} 例 : <code>apic1(config)# errdisable recovery cause mcp-loop</code>	インターフェイスが error-disabled ステートから自動的に回復する条件を指定すると、デバイスはインターフェイスを再びアップします。デフォルトではディセー

	コマンドまたはアクション	目的
		ブルになっています。条件オプションは次のとおりです。 • bpduguard : BPDU ガードエラー ディセーブルから回復するためのイネーブル タイマー。 • ep-move : エンドポイント移動エラー ディセーブルから回復するためのイネーブルタイマー。 • mcp-loop : MCP ループエラー ディセーブルから回復するためのイネーブルタイマー。

例

この例では、MCP ループ エラー ディセーブルから回復するための EDR を設定する方法を示します。

```
apic1# configure terminal
apic1(config)# errdisable recovery interval 300
apic1(config)# errdisable recovery cause mcp-loop
```

リンク レベル検出プロトコルの設定

Link Layer Discovery Protocol (LLDP) は、ネットワーク デバイスがネットワーク上の他のデバイスに自分の情報をアドバタイズできるようにするデバイス ディスカバリ プロトコルです。LLDP によってスイッチ間のレイヤ 2 接続が決まります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] lldp holdtime seconds 例 : apic1(config)# lldp holdtime	LLDP パケットで送信されるホールド時間を指定します。

	コマンドまたはアクション	目的
ステップ 3	[no] lldp holdtime <i>seconds</i> 例 : <code>apic1(config)# lldp holdtime 120</code>	LLDP パケットで送信されるホールド時間を指定します。値の範囲は 10 ～ 255 秒で、デフォルト値は 120 秒です。
ステップ 4	[no] lldp reinit <i>seconds</i> 例 : <code>apic1(config)# lldp reinit 2</code>	任意のインターフェイス上で LLDP を初期化する際の遅延時間を指定します。指定できる範囲は 1 ～ 10 秒です。デフォルトは 2 秒です。
ステップ 5	[no] lldp timer <i>seconds</i> 例 : <code>apic1(config)# lldp timer 30</code>	LLDP アップデートの送信頻度を秒単位で設定します。値の範囲は 5 ～ 254 秒で、デフォルト値は 30 秒です。

例

この例では、LLDP を設定する方法を示します。

```
apic1# configure terminal
apic1(config)# lldp holdtime 120
apic1(config)# lldp reinit 2
apic1(config)# lldp timer 30
```

Miscabling プロトコルの設定

ACI ファブリックは、ACI アクセス ポートに接続されているレイヤ 2 ネットワーク セグメントでループを検出できるループ検出ポリシーを提供します。ACI ファブリックはファブリックレベルのポリシーである Mis-Cabling Protocol (MCP) を実装し、MCP パラメータのプロビジョニング、誤配線が検出された場合のポートの動作の決定を行うことができます。MCP は外部レイヤ 2 ネットワークで実行される STP を補完するように機能し、アクセス ポートが受信するブリッジプロトコルデータユニット (BPDU) パケットを処理します。

ファブリック管理者は、ACI ファブリックによって開始された MCP パケットを識別するために MCP が使用するキーを提供します。管理者は、MCP ポリシーがループを識別する方法、ループに対するアクション (syslog のみ、またはポートを無効にする) を選択できます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	[no] mcp action port-disable 例 : apicl(config)# mcp action port-disable	誤配線が検出された場合にポートを無効状態にするかどうかを指定します。
ステップ 3	[no] mcp enable [key key-value] 例 : apicl(config)# mcp enable key 0123456789abcdef	ファブリック全体での MCP プロトコルのグローバルな有効化または無効化を可能にします。ポリシーを有効にするときはパスワード（キー）が必要ですが、無効にするときは不要です。
ステップ 4	[no] mcp factor number 例 : apicl(config)# mcp factor 64	MCP パケットを送信するときに使用されるループ検出の増倍係数を設定します。指定できる範囲は 1 ～ 255 です。
ステップ 5	[no] mcp init-delay seconds 例 : apicl(config)# mcp init-delay 180	初期遅延時間を指定します。指定できる範囲は 0 ～ 1800 秒です。デフォルト値は 180 秒です。
ステップ 6	[no] mcp transmit-frequency frequency 例 : apicl(config)# mcp transmit-frequency 2	誤配線を検出するために MCP パケットが送信される頻度を設定します。範囲は 100 ミリ秒～300 秒です。デフォルトは 2 秒です。

例

MCP の送信頻度の間隔を 2 秒に設定する例を示します。

```
apicl# configure terminal
apicl(config)# mcp action port-disable
apicl(config)# mcp enable key 0123456789abcdef
apicl(config)# mcp factor 64
apicl(config)# mcp init-delay 180
apicl(config)# mcp transmit-frequency 2
```

この例では、MCP の送信頻度を 2 秒と 300 ミリ秒単位に設定する方法を示します。

```
apicl# configure terminal
apicl(config)# mcp action port-disable
apicl(config)# mcp enable key 0123456789abcdef
apicl(config)# mcp factor 64
apicl(config)# mcp init-delay 180
apicl(config)# mcp transmit-frequency 2 300
```

エンドポイントのループ防止ポリシーの設定

エンドポイントのループ防止ポリシーは、あるファブリックポートから別のファブリックポートまでのエンドポイント（ホスト）の頻繁な移動の検出に使用されるファブリックレベルのポリシーです。ポリシーは、このようなイベントが検出された場合に、実行するアクションの内容を設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] endpoint loop-detect action {bd-learn-disable port-disable} 例： <code>apic1(config)# endpoint loop-detect action port-disable</code>	エンドポイントのループが検出されたときに実行するアクションを指定します。次のオプションがあります。 • bd-learn-disable—ブリッジ ドメインで MAC アドレス ラーニングを無効にします。 • port-disable—ポートを無効にします。
ステップ 3	[no] endpoint loop-detect enable 例： <code>apic1(config)# endpoint loop-detect enable</code>	ファブリック全体のグローバルなエンドポイントのループ防止プロトコルをイネーブルまたはディセーブルにすることができます。
ステップ 4	[no] endpoint loop-detect factor <i>number</i> 例： <code>apic1(config)# endpoint loop-detect factor 64</code>	ループ検出の乗算係数を設定します。指定できる範囲は 1 ～ 255 です。
ステップ 5	[no] endpoint loop-detect interval <i>seconds</i> 例： <code>apic1(config)# endpoint loop-detect interval 60</code>	ループ検出間隔を指定します。指定できる範囲は 30 ～ 300 秒です。

例

この例では、エンドポイントのループ防止ポリシーを設定する方法を示します。

```

apic1# configure terminal
apic1(config)# endpoint loop-detect action port-disable
apic1(config)# endpoint loop-detect enable
apic1(config)# endpoint loop-detect factor 64
apic1(config)# endpoint loop-detect interval 60

```

不正なエンドポイントの制御ポリシーの設定

不正なエンドポイントの制御ポリシーについて

別の ToR ポートで頻繁かつ急速にパケットを挿入し、802.1q を変更することで (したがってエンドポイントのエミュレートが移動します)、不正なエンドポイントはラック上部 (ToR) のスイッチに攻撃して、学習クラスと EPG ポートが変更されることになります。

誤設定により頻繁に IP アドレスと MAC アドレスが変更 (移動する) されることになります。

ファブリックの急速な移動などで、大きなネットワークの不安定状態、高い CPU 使用率、まれなケースでは、大量かつ長期のメッセージおよびトランザクションサービス (MTS) バッファ消費のため、エンドポイント マッパー (EPM) および EPM クライアント (EPMC) がクラッシュすることになります。また、このような頻繁な移動により、EPM および EPMC ログが非常にすばやくロールオーバーされ、無関係なエンドポイントのデバッグを妨害する可能性があります。

不正 EP 制御機能は、この脆弱性にすばやく対処します。

- このような急速 MAC および IP エンドポイントの移動の特定
- 一時的に移動を停止することで、エンドポイントを静的にします (したがって、エンドポイントを隔離します)
- **不正 EP 検出間隔** にエンドポイント スタティックを保持し、不正なエンドポイント経由のトラフィックを廃棄します。この時間が経過すると、認可されていない MAC または IP アドレスが削除されます
- ホスト トラッキング パケットを生成することで、影響を受ける MAC または IP アドレスを再学習するようシステムを有効にします
- 障害が発生すると是正措置が有効になります

不正 EP 制御ポリシーはグローバルに設定され、その他のループ防止方法とは異なり、個別のエンドポイントレベルで機能します (IP および MAC アドレス)。ローカルまたはリモート間の移動を区別せず、インターフェイスのあらゆる種類の変更がエンドポイントが隔離されるかどうか決定する際に移動とみなされます。

NX-OS スタイル CLI を使用した エンドポイント の設定

不正 EP 制御 ポリシーをファブリックに設定して、NX-OS スタイル CLI を使用して認証されていないエンドポイントを検出および削除できます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	endpoint rogue-detect enable 例 : apic1(config)# endpoint rogue-detect enable	グローバル不正エンドポイント制御ポリシーを有効にします。
ステップ 3	endpoint rogue-detect hold-interval hold_interval 例 : apic1(config)# endpoint rogue-detect hold-interval 1800	エンドポイントの不正が発見された後に保留間隔を秒単位で設定することで、ラーニングが防止されるように静的を維持し、不正なエンドポイントによるトラフィックがドロップします。このインターバルが経過すると、エンドポイントは削除されます。有効な値は 300 ～ 3600 秒です。デフォルト値は 1800 です。 (注) サポートされている保留間隔の範囲は 1800 ～ 3600 秒です。
ステップ 4	endpoint rogue-detect interval interval 例 : apic1(config)# endpoint rogue-detect interval 60	不正なエンドポイントを検出するまでの時間を指定する不正の検出間隔を秒で設定します。有効な値は 0 ～ 65535 秒です。デフォルトは 60 です。
ステップ 5	endpoint rogue-detect factor factor 例 : apic1# endpoint rogue-detect factor 6	エンドポイントが承認されたかどうかを確認する際に、乗算係数を指定します。エンドポイントは、複数の時間間隔の間に移動した場合、EP は不正を宣言します。有効値は 2 ～ 65535 です。デフォルト値は 6 です。 (注) サポートされている 不正 EP 検出乗算係数 の範囲は 2 ～ 10 です。

	コマンドまたはアクション	目的
ステップ 6	この例では、不正なエンドポイント制御ポリシーを設定します。 例 : <pre>apicl# cconfigure apicl(config)# endpoint rogue-detect enable apicl(config)# endpoint rogue-detect hold-interval 1800 apicl(config)# endpoint rogue-detect interval 60 apicl(config)# endpoint rogue-detect factor 6</pre>	

IP エージングの設定

概要

IP エージング ポリシーは、エンドポイントの未使用の IP アドレスを追跡しエージングが行われます。トラッキングはブリッジドメインに設定されたエンドポイント保持ポリシーを使用して実行され、ローカルエンドポイント エージング間隔の 75% で、ARP 要求 (IPv4) やネイバー要請 (IPv6) を送信します。IP アドレスから応答を受信しなかった場合、その IP アドレスはエージングアウトします。

このドキュメントでは、IP エージング ポリシーを設定する方法について説明します。

NX-OS スタイル CLI を使用した IP エージング ポリシーの設定

このセクションでは、CLI を使用した IP エージング ポリシーを有効および無効にする方法を説明します。

手順

ステップ 1 IP エージング ポリシーを有効にするには :

例 :

```
ifc1(config)# endpoint ip aging
```

ステップ 2 IP エージング ポリシーを無効にするには :

例 :

```
ifav9-ifc1(config)# no endpoint ip aging
```

次のタスク

エンドポイントの IP アドレスをトラッキングするために使用される間隔を指定するには、エンドポイント保持ポリシーを作成します。

ダイナミック ロード バランサの設定

ダイナミック ロード バランシング (DLB) により、輻輳レベルに従ってトラフィックの割り当てが調整されます。DLB では、使用可能なパス間の輻輳が測定され、輻輳状態が最も少ないパスにフローが配置されるので、データが最適またはほぼ最適に配置されます。

DLB は、フローまたはフローレットの粒度を使用して使用可能なアップリンクにトラフィックを配置するように設定できます。フローレットは、時間の大きなギャップによって適切に区切られるフローからのパケットのバーストです。パケットの2つのバースト間のアイドル間隔が使用可能なパス間の遅延の最大差より大きい場合、2 番目のバースト（またはフローレット）を1つ目とは異なるパスに沿ってパケットのリオーダーなしで送信できます。このアイドル間隔は、フローレット タイマーと呼ばれるタイマーによって測定されます。フローレットにより、パケットリオーダーを引き起こすことなくロードバランシングに対する粒度の高いフローの代替が提供されます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] system dynamic-load-balance mode {dynamic-aggressive dynamic-conservative link-failure-resiliency packet-prioritization}	ロード バランサの動作モードを指定します。使用可能なモードを次に示します。 • dynamic-aggressive : フローレット タイムアウトは比較的小さい値です。この非常に精密なダイナミック ロード バランシングはトラフィックの分配に最適ですが、パケット リオーダーが発生する場合があります。ただし、アプリケーションのパフォーマンスに対する包括的なメリットは、保守的なモードと同等かそれよりも優れています。 • dynamic-conservative : フローレット タイムアウトは、パケットが並び替えられないことを保証する大きな値です。新しいフローレットの機

	コマンドまたはアクション	目的
		会の頻度が少ないので、トレードオフは精度が低いロード バランシングです。 • link-failure-resiliency : 静的ロード バランシングにより、使用可能なリンクにほぼ均等な流量が分配されます。 • packet-prioritization : Dynamic Packet Prioritization (DPP) は、長いフローよりも短いフローを優先します。短いフローは約 15 パケット未満です。短いフローは、長いフローよりも遅延に敏感であるため、DPP はアプリケーション全体のパフォーマンスを向上できます。 <pre>apic1(config)# system dynamic-load-balance mode packet-prioritization</pre>

例

この例では、パケットの優先順位付でロード バランシングを設定する方法を示します。

```
apic1# configure terminal
apic1(config)# system dynamic-load-balance mode packet-prioritization
```

スパニングツリー プロトコルの設定

Multiple Spanning-Tree (MST) は複数の VLAN を同一のスパニングツリー インスタンスにマッピングできるようにして、多数の VLAN をサポートする場合に必要なスパニングツリー インスタンスの数を減らします。



(注) ポート VLAN ごとの機能 (localPort 範囲に同じ VLAN ID を使用してリーフ スイッチで複数の EPG を設定する) で設定されたインターフェイスでは、マルチ スパニング ツリー (MST) はサポートされません。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	spanning-tree mst configuration 例 : <code>apic1(config)# spanning-tree mst configuration</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 3	[no] bpdu-filter 例 : <code>apic1(config-stp)# bpdu-filter</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 4	[no] region <i>region-name</i> 例 : <code>apic1(config-stp)# region region1</code>	スイッチを MST インスタンスに加入させるには、同じ MST コンフィギュレーション情報を使用して矛盾のないようにスイッチを設定する必要があります。同じ MST 設定の相互接続スイッチの集まりによって MST リージョンが構成されます。各リージョンは、最大 65 のスパニングツリー インスタンスまでサポートします。
ステップ 5	[no] instance <i>instance-id</i> vlan <i>vlan-range</i> 例 : <code>apic1(config-stp-region)# instance 2 vlan 1-63</code>	VLAN を MSTI にマップします。VLAN には、一度に 1 つのスパニングツリー インスタンスのみ割り当てることができます。インスタンス ID の範囲は 1 ～ 4094 です。VLAN の範囲を指定するには、ハイフンを使用します。
ステップ 6	revision <i>number</i> 例 : <code>apic1(config-stp-region)# revision 16</code>	設定リビジョン番号を指定します。指定できる範囲は 0 ～ 65535 です。

例

次に、MST スパニングツリー ポリシーを設定する例を示します。

```
apic1# configure terminal
apic1(config)# spanning-tree mst configuration
apic1(config-stp)# bpdu-filter
apic1(config-stp)# region region1
apic1(config-stp-region)# instance 2 vlan 1-63
```

```
apic1(config-stp-region)# revision 16
```

『Configuring IS-IS』

Intermediate System-to-Intermediate System (IS-IS) はネットワーク トポロジの変化を検出し、ネットワーク上の他のノードへのループフリー ルートを計算できる、ダイナミック リンクステート ルーティング プロトコルです。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	template isis-fabric <i>isis-fabric-template-name</i> 例 : <code>apic1(config)# template isis-fabric polIsIs</code>	Intermediate system-to-intermediate system (IS-IS) configuration mode(設定モード、コンフィギュレーション モード)を開始し、IS-IS ファブリック テンプレート (policy) を作成します。
ステップ 3	[no] lsp-fast-flood 例 : <code>apic1(config-template-isis-fabric)# lsp-fast-flood</code>	高速フラッディング機能をイネーブルにします。これにより、ネットワーク内に新しいリンクステート パケット (LSP) が生成され、新しいLSPによって最短パス優先 (SPF) がトリガーされるときのコンバージェンス時間が改善されます。 ネットワーク全体でコンバージェンス時間を短縮するために、ルータが SPF 計算を実行する前に、LSP の高速フラッディングをイネーブルにしておくことをお勧めします。
ステップ 4	[no] lsp-gen-interval level-1 <i>lsp-max-wait</i> [<i>lsp-initial-wait</i> <i>lsp-second-wait</i>] 例 : <code>apic1(config-template-isis-fabric)# lsp-gen-interval level-1 500 500 500</code>	LSP 発生に関する IS-IS スロットルを設定します。パラメータは次のとおりです。 • <i>lsp-max-wait</i> : トリガーから LSP 発生までの最大待ち時間。 • <i>lsp-initial-wait</i> : トリガーから LSP 発生までの初期待ち時間。

	コマンドまたはアクション	目的
		• <i>lsp-second-wait</i> : バックオフ時の LSP スロットルに使用する第 2 待ち時間。 <i>lsp-max-wait</i> パラメータは必須です。他の 2 つのパラメータはオプションですが、同時に表示する必要があります。それぞれの範囲は 50 ～ 120000 ミリ秒です。
ステップ 5	[no] lsp-mtu <i>mtu</i> 例 : <pre>apicl(config-template-isis-fabric)# lsp-mtu 2048</pre>	IS-IS hello パケットの最大伝送ユニット (MTU) を設定します。指定できる範囲は 256 ～ 4352 です。 IS-IS hello パケットは、隣接関係の検出および維持のために使用されます。デフォルトでは、hello パケットは最大伝送ユニット (MTU) の最大限のサイズまでパディングされ、大きいフレームでの伝送の問題または隣接インターフェイスでの MTU の不一致に起因するエラーを早期に検出できます。ただし、IS-IS 隣接関係の形成はリンクでの MTU 不一致が原因で失敗し、MTU サイズの調整が必要になることがあります。
ステップ 6	[no] spf-interval level-1 <i>spf-max-wait</i> [<i>spf-initial-wait</i> <i>spf-second-wait</i>] 例 : <pre>apicl(config-template-isis-fabric)# spf-interval level-1 500 500 500</pre>	LSA 到着までのインターバルを設定します。パラメータは次のとおりです。 • <i>spf-max-wait</i> : トリガーから SPF 計算までの最大待機時間。 • <i>spf-initial-wait</i> : トリガーから SPF 計算までの最初の待機時間。 • <i>spf-second-wait</i> : バックオフ時の SPF 計算に使用する第 2 待機時間。 <i>spf-max-wait</i> パラメータは必須です。他の 2 つのパラメータはオプションですが、同時に表示する必要があります。それぞれの範囲は 50 ～ 120000 ミリ秒です。

	コマンドまたはアクション	目的
ステップ 7	exit 例 : <code>apic1(config-template-isis-fabric)# exit</code>	グローバル コンフィギュレーション モードに戻ります。
ステップ 8	template pod-group <i>pod-group-template-name</i> 例 : <code>apic1(config)# template pod-group allPods</code>	ポッド グループ テンプレート (policy) を作成します。
ステップ 9	inherit pod-group <i>pod-group-name</i> 例 : <code>apic1(config-pod-group)# inherit isis-fabric polIsIs</code>	Configures the template pod-group to use the previously configured isis-fabric template (policy).
ステップ 10	exit 例 : <code>apic1(config-pod-group)# exit</code>	グローバル コンフィギュレーション モードに戻ります。
ステップ 11	pod-profile <i>pod-profile-name</i> 例 : <code>apic1(config)# pod-profile all</code>	ポッド プロファイルを設定します。
ステップ 12	pods { <i>pod-range-1-255</i> all } 例 : <code>apic1(config-pod-profile)# pods all</code>	一連のポッドを設定します。
ステップ 13	inherit pod-group <i>pod-group-name</i> 例 : <code>apic1(config-pod-profile-pods)# inherit pod-group allPods</code>	Configures the pod-profile to use the previously configured pod group.
ステップ 14	end 例 : <code>apic1(config-pod-profile-pods)# end</code>	EXEC モードに戻ります。

例

次に、IS-IS を設定する例を示します。

```
aapic1# configure
apic1(config)# template isis-fabric polIsIs
apic1(config-template-isis-fabric)# lsp-fast-flood
```

```

apicl(config-template-isis-fabric)# lsp-gen-interval level-1 500 500 500
apicl(config-template-isis-fabric)# lsp-mtu 2048
apicl(config-template-isis-fabric)# spf-interval level-1 500 500 500
apicl(config-template-isis-fabric)# exit
apicl(config)# template pod-group allPods
apicl(config-pod-group)# inherit isis-fabric pollIsIs
apicl(config-pod-group)# exit
apicl(config)# pod-profile all
apicl(config-pod-profile)# pods all
apicl(config-pod-profile-pods)# inherit pod-group allPods
apicl(config-pod-profile-pods)# end
apicl#

```

BGP ルート リフレクタの設定

ACIファブリックのルートリフレクタは、マルチプロトコルBorder Gateway Protocol（MP-BGP）を使用してファブリック内に外部ルートを配布します。ACIファブリックでルートリフレクタをイネーブルにするには、ファブリックの管理者がルートリフレクタになるスパインスイッチを選択して、自律システム（AS）番号を提供する必要があります。冗長性のために、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台のプライマリリフレクタと1台のセカンダリリフレクタ）。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	bgp-fabric 例： apicl(config)# bgp-fabric	ファブリックの BGP 設定モードを開始します。
ステップ 3	asn <i>asn-value</i> 例： apicl(config-bgp-fabric)# asn 123456789	自律システムを一意に識別する、BGP 自律システム番号（ASN）を設定します。ASN は 1 ～ 4294967295 の間です。 ネットワーク全体でコンバージェンス時間を短縮するために、ルータが SPF 計算を実行する前に、LSP の高速フラッディングをイネーブルにしておくことをお勧めします。
ステップ 4	[no] route-reflector spine リスト 例： apicl(config-bgp-fabric)# route-reflector spine spine1,spine2	ルートリフレクタとして最大2つのスパインノードを設定します。冗長性のために、プライマリおよびセカンダリ

	コマンドまたはアクション	目的
		ルートリフレクタを設定する必要があります。

例

この例では、BGP ルートリフレクタとして spine1 および spine2 を設定する方法を示します。

```
apic1# configure
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 123456789
apic1(config-bgp-fabric)# route-reflector spine spine1,spine2
apic1(config-bgp-fabric)# exit
apic1(config)#
```

ノードのデコミッション

次の 2 レベルの稼働停止がサポートされています。

- 標準：ノードの無効化と同様です。After being decommissioned, the node cannot rejoin the fabric until the **no decommission** command is executed.
- 完全：ノードを稼働停止すると、ノードに関連するすべてのファブリック設定がクリアされます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] decommission {controller switch} node-id [remove-from-controller] 例： apic1(config)# decommission switch 104 remove-from-controller	指定されたノードを稼働停止します。コントローラ ノード ID の番号は 1 ～ 100 の範囲にあり、スイッチ ノード ID の番号は 101 ～ 4000 の範囲にあることに注意してください。

例

次に、ノード 104（スイッチ）と、標準レベルで稼働停止された再稼働ノード 5（コントローラ）の完全な稼働停止を実行する例を示します。

```
apic1# configure
apic1(config)# decommission switch 104 remove-from-controller
apic1(config)# no decommission controller 5
```

電力管理の設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>apic1# configure</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] power redundancy-policy <i>policy-name</i> 例 : <pre>apic1(config)# power redundancy-policy myPowerPolicy</pre>	電源の冗長性ポリシーを作成または設定します。
ステップ 3	[no] description <i>text</i> 例 : <pre>apic1(config-power)# description 'This is my power redundancy policy'</pre>	この電源の冗長性ポリシーの説明を追加します。テキストにスペースが含まれている場合は、単一引用符で囲む必要があります。
ステップ 4	[no] redundancy-mode {combined ps-redundant redundant} 例 : <pre>apic1(config-power)# redundancy-mode ps-redundant</pre>	電源の冗長性モードを指定します。 • combined—このモードは電源の冗長性を提供しません。使用可能な電力は、すべての電源装置の電力の合計です。 • ps-redundant—このモードは、アクティブな電源がダウンしたときに備えて予備の電源装置を1台追加したものです。最大の電力を供給できる電源がスタンバイ モードで動作します。残りの1台または2台の電源装置がアクティブになります。使用可能な電力は、アクティブな電源ユニットによって供給される電力の合計です。 • redundant—このモードは、電源の冗長性と入力電源回路網の冗長性を組み合わせたものです。つまり、

	コマンドまたはアクション	目的
		シャーンは予備の電源装置を1台備えており、各電源装置の半分が1つの電源回路網に接続され、残りの半分がもう1つの電源回路網に接続されます。使用可能な電力は、電源装置モードと入力電源モードのうち使用可能電力が小さいほうです。

例

次に、ps-redundant モードの電源の冗長性ポリシーを設定する例を示します。

```
apic1# configure
apic1(config)# power redundancy-policy myPowerPolicy
apic1(config-pod)# isis fabric
apic1(config-power)# description 'This is my power redundancy policy'
apic1(config-power)# redundancy-mode ps-redundant
```

スケジュールの設定

スケジュールにより、設定のインポート/エクスポートまたはテクニカル サポートの収集などの操作を1つ以上の指定した時間帯に発生させることができます。

スケジュールには、一連のタイムウィンドウ（オカレンス）が含まれます。これらのウィンドウは、1回だけ発生させるか、または毎週指定した日時に繰り返し発生させることができます。期間や実行するタスクの最大数などのウィンドウで定義されているオプションにより、スケジュール設定されたタスクの実行時期が決定されます。たとえば、最大時間長またはタスク数に達したため特定のメンテナンス時間帯に変更を展開できない場合、この展開は次のメンテナンス時間に持ち越されます。

各スケジュールは、APIC が1つまたは複数のメンテナンス時間帯に入っているかどうか、定期的に確認します。入っている場合、スケジュールはメンテナンスポリシーで指定された制限に対し適切な展開を実行します。

スケジュールには、スケジュールに関連付けられたメンテナンス時間を決定する1つ以上のオカレンスが含まれています。オカレンスは次のいずれかになります。

- 絶対（1回）時間帯：絶対時間帯は、1回しか発生しないスケジュールを定義します。これらの時間帯は、その時間帯の最大時間長まで、または時間帯の中で実行可能なタスクの最大数に達するまで継続されます。
- 繰り返し時間帯：繰り返し時間帯は、繰り返しのスケジュールを定義します。この時間帯は、タスクの最大数に達するまで、または時間帯に指定された日の終わりに達するまで継続します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apicl# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] scheduler <i>schedule-name</i> 例 : <code>apicl(config)# scheduler controller schedule myScheduler</code>	新しいスケジューラを作成するか、既存のスケジューラを設定します。
ステップ 3	[no] description <i>text</i> 例 : <code>apicl(config-scheduler)# description 'This is my scheduler'</code>	このスケジューラの説明を追加します。テキストにスペースが含まれている場合は、単一引用符で囲む必要があります。
ステップ 4	[no] absolute window <i>window-name</i> 例 : <code>apicl(config-scheduler)# absolute window myAbsoluteWindow</code>	絶対（1 回）の時間帯スケジュールを作成します。
ステップ 5	[no] max concurrent nodes <i>count</i> 例 : <code>apicl(config-scheduler-absolute)# max concurrent nodes 300</code>	同時に処理できるノード（タスク）の最大数を設定します。指定できる範囲は 0 ～ 65535 です。ノード数を制限しない場合は 0 に設定します。
ステップ 6	[no] max running time <i>time</i> 例 : <code>apicl(config-scheduler-absolute)# max running time 00:01:30:00</code>	dd:hh:mm:ss の形式でタスクの最大実行時間を設定します。指定できる範囲は 0 ～ 65535 です。時間の制限がない場合は 0 に設定します。
ステップ 7	[no] time start <i>time</i> 例 : <code>apicl(config-scheduler-absolute)# time start 2016:jan:01:12:01</code>	[[[yyyy:]mmm:]dd:]HH:MM 形式で開始時刻を設定します。
ステップ 8	exit 例 : <code>apicl(config-scheduler-absolute)# exit</code>	スケジューラ コンフィギュレーション モードに戻ります。
ステップ 9	[no] recurring window <i>window-name</i> 例 : <code>apicl(config-scheduler)# recurring window myRecurringWindow</code>	繰り返し時間帯のスケジュールを作成します。

	コマンドまたはアクション	目的
ステップ 10	[no] max concurrent nodes count 例 : <pre>apic1(config-scheduler-recurring)# max concurrent nodes 300</pre>	同時に処理できるノード（タスク）の最大数を設定します。指定できる範囲は 0 ～ 65535 です。ノード数を制限しない場合は 0 に設定します。
ステップ 11	[no] max running time time 例 : <pre>apic1(config-scheduler-recurring)# max running time 00:01:30:00</pre>	dd:hh:mm:ss の形式でタスクの最大実行時間を設定します。指定できる範囲は 0 ～ 65535 です。時間の制限がない場合は 0 に設定します。
ステップ 12	[no] time start {daily HH:MM weekly (See usage) HH:MM} 例 : <pre>apic1(config-scheduler-recurring)# time start weekly wednesday 12:30</pre>	期間（毎日または毎週）と開始時刻を設定します。If weekly is selected, choose from these options: • monday • tuesday • wednesday • thursday • friday • saturday • sunday • even-day • odd-day • every-day

例

次に、毎週水曜日に実行するよう繰り返しスケジューラを設定する例を示します。

```
apic1# configure
apic1(config)# scheduler controller schedule myScheduler
apic1(config-scheduler)# description 'This is my scheduler'
apic1(config-scheduler)# recurring window myRecurringWindow
apic1(config-scheduler-recurring)# max concurrent nodes 300
apic1(config-scheduler-recurring)# max running time 00:01:30:00
apic1(config-scheduler-recurring)# time start weekly wednesday 12:30
```

システム MTU の設定

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] system jumbomtu size 例 : <code>apic1(config)# system jumbomtu 9000</code>	ホスト側ポートの最大伝送ユニット (MTU) を設定します。Cisco APIC リリース 3.1 (2) まで、最大値の範囲は 576 に 9000 バイトです。リリース 3.1 (2) 以降、最大 MTU 値は 9216 です。デフォルトでは、9000 から変更されていません。

例

次に、システムの MTU サイズを設定する例を示します。

```
apic1# configure terminal
apic1(config)# system jumbomtu 9000
```

PTP について

Precision Time Protocol (PTP) は、IEEE 1588 で定義された、ネットワークに分散したノードの時刻同期プロトコルです。PTP を使用すると、分散したクロックを、イーサネット ネットワークを経由して、1 マイクロ秒以下の精度で同期させることができます。PTP の精度は ACI ファブリック スパイน์およびリーフでサポートされているハードウェアから提供されます。これにより、プロトコルが正確にネットワーク全体のメッセージ遅延と変動を補正できます。

PTP は、システムのリアルタイム PTP クロックが相互に同期する方法を指定する分散プロトコルです。これらのクロックは、グランドマスタークロック (階層の最上部にあるクロック) を持つマスター/スレーブ同期階層に編成され、システム全体の時間基準を決定します。同期は、タイミング情報を使用して階層のマスターの時刻にクロックを調整するメンバーと、PTP タイミングメッセージを交換することによって実現されます。PTP は、PTP ドメインと呼ばれる論理範囲内で動作します。

PTPプロセスは、マスター/スレーブ階層の確立とクロックの同期の2つのフェーズで構成されます。PTPドメイン内では、オーディナリクロックまたは境界クロックの各ポートが、次のプロセスに従ってステートを決定します。

- 受信したすべての（マスタ状態のポートによって発行された）アナウンスメッセージの内容を検査します。
- 外部マスタのデータセット（アナウンスメッセージ内）とローカルクロックで、優先順位、クロッククラス、精度などを比較します。
- 自身の状態がマスタまたはスレーブのいずれであるかを決定します。

マスター/スレーブ階層が確立されると、クロックは次のように同期されます。

- マスターはスレーブに同期メッセージを送信し、送信された時刻を記録します。
- スレーブは同期メッセージを受信し、受信した時刻を記録します。すべての同期メッセージには、フォローアップメッセージがあります。したがって、同期メッセージの数は、フォローアップメッセージの数と同じである必要があります。
- スレーブはマスターに遅延要求メッセージを送信し、送信された時刻を記録します。
- マスターは遅延要求メッセージを受信し、受信した時刻を記録します。
- マスターはスレーブに遅延応答メッセージを送信します。遅延要求メッセージの数は、遅延応答メッセージの数と同じである必要があります。
- スレーブは、これらのタイムスタンプを使用して、クロックをマスターの時刻に調整します。

ACIファブリックでは、PTP機能がAPIC内でグローバルに有効になっている場合、ソフトウェアがサポートされているスパインおよびリーフの特定のインターフェイスで自動的にPTPを有効にします。この自動設定は、サポートされているすべてのノードで最適にPTPが有効になっていることを保証します。外部グラントマスタクロックがない場合、スパインスイッチのいずれかがグラントマスタとして選択されます。PTPスレーブとして動作するように、マスタスパインには他のスパインおよびリーフスイッチと比較して異なるPTP優先順位が与えられます。この方法により、ファブリック内のすべてのリーフスイッチがマスタスパインのPTPクロックと同期します。

外部グラントマスタクロックがスパインに接続している場合、スパインは外部GMに同期し、リーフノードに対してマスタとして動作します。

PTP デフォルト設定

次の表に、PTPパラメータのデフォルト設定を示します。

パラメータ	デフォルト
PTP デバイス タイプ	境界クロック
PTP クロック タイプ	ツーステップクロック

パラメータ	デフォルト
PTP ドメイン	0
クロックをアドバタイズする場合、PTP プライオリティ 1 値	255
クロックをアドバタイズする場合、PTP プライオリティ 2 値	255
PTP アナウンス間隔	1 ログ秒
PTP アナウンス タイムアウト	3 アナウンス間隔
PTP 遅延要求間隔	0 ログ秒
PTP 同期間隔	-2 ログ秒
PTP VLAN	1



- (注) PTPは境界クロックモードのみで動作します。シスコでは、スイッチに接続された、同期を必要とするクロックが含まれるサーバを使用して、グランドマスター クロック（10 MHz）アップストリームを配置することを推奨します。

PTP の検証

コマンド	目的
show ptp brief	PTP のステータスを表示します。
show ptp clock	ローカルクロックのプロパティ（クロック ID など）を表示します。
show ptp clock foreign-masters record interface ethernet slot/port	PTP プロセスが認識している外部マスターの状態を表示します。外部マスターごとに、出力に、クロック ID、基本的なクロックプロパティ、およびクロックがグランドマスターとして使用されているかどうかが表示されます。
show ptp corrections	最後の数個の PTP 修正を表示します。
show ptp counters [all interface Ethernet slot/port]	すべてのインターフェイスまたは指定されたインターフェイスの PTP パケットカウンタを表示します。
show ptp parent	PTP の親のプロパティを表示します。

注意事項と制約事項

次の注意事項と制約事項に従ってください。

- 遅延には、Precision Time Protocol (PTP) を使用して同期するファブリック内のすべてのノードが必要です。
- Cisco ACI リリース 3.0(1x) 以降では、遅延測定および PTP は次でのみサポートされます。
 - N9K-X9732C-EX、
 - N9K-X9736C-EX
 - N9K-C93180YC-FX
 - N9K-C93108TC-FX
 - N9K-C93108TC-EX
 - N9K-C93180YC-EX
 - N9K-X9736C-FX
- 遅延測定は、TOR ベースで EX または FX 経由で入力、出力、送信を行うパケットでのみサポートされています。
- ファブリック内のすべてのスパイン ノードは、PTP をサポートするためライン カードに基づいて EX または FX を有する必要があります。
- PTP および遅延機能は、任意の NS Tor またはスパインではサポートされていません。ファブリック内の非 EX/FX Tor のプレゼンスについて、すべてのスパインに外部 GM 接続を行い、PTP 時間がサポートされているすべての TOR で同期することを推奨します。
- 外部グランドマスタ (GM) クロックは、単一ポッドで PTP が必須ではありません。ACI ファブリックに接続されている外部 GM がいない場合は、スパイン ノードの 1 個が GM として機能します。このスパインには、254 の PTP priority1 値があります。ファブリック内のその他すべてのスパインおよびリーフ スイッチは、このマスタ スパイン クロックにクロックを同期します。外部 GM がスパインに後で接続される場合、ファブリック全体に対して GM として動作するために 254 未満の優先順位値が必要です。
- 外部グランドマスタ クロックは、マルチポッド シナリオで PTP が必須です。さらに、外部 GM はグランドマスタ クロックが異なる POD のスパインに対するマスタになるなど、IPN に接続される必要があります。IPN に接続されているスパインは境界クロックとして機能し、ポッド内のすべてのノードがこのスパインのクロックを同期します。
- PTP は境界クロック モードのみで動作します。エンドツーエンド トランスペアレント クロック モードとピアツーピア トランスペアレント クロック モードはサポートされません。
- PTP はユーザ データグラム プロトコル (UDP) 上の転送をサポートします。イーサネット上の転送はサポートされません。

- ACIPTPはマルチキャスト通信のみをサポートし、ユニキャストモードはサポートされていません。

NX-OS CLI を使用した PTP の設定

手順

ステップ1 PTP を有効にします。

例：

```
Enable ptp:
=====
apic# configure terminal
apic(config)# ptp
Disable ptp:
=====
apic# configure terminal
apic(config)# no ptp
```

ステップ2 ACI スイッチで PTP を確認するには：

例：

```
leaf1# show ptp brief
PTP port status
-----
Port          State
-----
Eth1/49       Slave

leaf1#
leaf1#
leaf1# show ptp clock
PTP Device Type: Boundary clock
Clock Identity : 0c:75:bd:ff:fe:03:1d:10
Clock Domain: 0
Number of PTP ports: 1
Priority1 : 255
Priority2 : 255
Clock Quality:
    Class : 248
    Accuracy : 254
    Offset (log variance) : 65535
Offset From Master : 32
Mean Path Delay : 128
Steps removed : 1
Local clock time:Thu Jul 27 19:43:42 2017

leaf1#
leaf1# show ptp clock foreign-masters record interface ethernet 1/49

P1=Priority1, P2=Priority2, C=Class, A=Accuracy,
OSLV=Offset-Scaled-Log-Variance, SR=Steps-Removed
GM=Is grandmaster
```

```
-----
Interface          Clock-ID          P1    P2    C    A    OSLV    SR
```

```

-----
Eth1/49      d4:6d:50:ff:fe:e6:4d:3f      254  255    248    254  65535  0      GM

leaf1#
leaf1#
leaf1# show ptp corrections

PTP past corrections
-----
Slave Port          SUP Time          Correction(ns)    MeanPath Delay(ns)
-----
Eth1/49             Thu Jul 27 19:44:11 2017 364281          36                152
Eth1/49             Thu Jul 27 19:44:11 2017 114565          16                132
Eth1/49             Thu Jul 27 19:44:10 2017 862912           8                132
Eth1/49             Thu Jul 27 19:44:10 2017 610823           8                132
Eth1/49             Thu Jul 27 19:44:10 2017 359557          16                132
Eth1/49             Thu Jul 27 19:44:10 2017 109937           8                132
Eth1/49             Thu Jul 27 19:44:09 2017 858113          16                132
Eth1/49             Thu Jul 27 19:44:09 2017 606536          16                132
Eth1/49             Thu Jul 27 19:44:09 2017 354837         -16               132
Eth1/49             Thu Jul 27 19:44:09 2017 104226          24                148
Eth1/49             Thu Jul 27 19:44:08 2017 853263          24                148
Eth1/49             Thu Jul 27 19:44:08 2017 601780          16                148
Eth1/49             Thu Jul 27 19:44:08 2017 349639         -4                148
Eth1/49             Thu Jul 27 19:44:08 2017 99970           16                144
Eth1/49             Thu Jul 27 19:44:07 2017 848507           0                144
Eth1/49             Thu Jul 27 19:44:07 2017 596143          24                144
Eth1/49             Thu Jul 27 19:44:07 2017 344808           4                144
Eth1/49             Thu Jul 27 19:44:07 2017 93156          -16               140
Eth1/49             Thu Jul 27 19:44:06 2017 843263          24                140
Eth1/49             Thu Jul 27 19:44:06 2017 590189           8                140
leaf1#
leaf1#
leaf1# show ptp counters all

PTP Packet Counters of Interface Eth1/49:
-----
Packet Type          TX          RX
-----
Announce              56          5424
Sync                  441         43322
FollowUp              441         43321
Delay Request         7002         0
Delay Response         0          7002
PDelay Request         0           0
PDelay Response         0           0
PDelay Followup         0           0
Management             0           0
-----

leaf1#
leaf1#
leaf1# show ptp parent

PTP PARENT PROPERTIES

Parent Clock:
Parent Clock Identity: d4:6d:50:ff:fe:e6:4d:3f
Parent Port Number: 258
Observed Parent Offset (log variance): N/A
Observed Parent Clock Phase Change Rate: N/A

Grandmaster Clock:

```



```

Grandmaster Clock Identity: d4:6d:50:ff:fe:e6:4d:3f
Grandmaster Clock Quality:
    Class: 248
    Accuracy: 254
    Offset (log variance): 65535
    Priority1: 254
    Priority2: 255

```

```
leaf1#
```

ステップ3 トラブルシューティングの手順を確認するには：

例：

```
apic1# show troubleshoot eptoe session eptoe latency
```

```
Source --> Destination
Last Collection(30 seconds)
```

Average (microsec)	Standard Deviation (microsec)	Packet Count
18	24	1086

```
Cumulative
```

Average (microsec)	Max (microsec)	Total Packet Count
18	202	6117438



第 15 章

Cisco Tetration Analytics の設定

- [概要 \(483 ページ\)](#)
- [NX-OS スタイル CLI を使用した Cisco Tetration Analytics の設定 \(483 ページ\)](#)

概要

この記事は、Cisco APIC を使用しているとき、Cisco Tetration Analytics を設定する方法の例を示します。次の情報は Cisco Tetration Analytics を設定する際に適用されます。

- インバンド管理 IP アドレスは、Cisco Tetration Analytics エージェントがアクティブになっている各リーフで設定する必要があります。
- 分析ポリシーを定義し、Cisco Tetration Analytics サーバの宛先 IP アドレスを指定します。
- スイッチ プロファイルを作成し、前の手順で作成したポリシー グループを含みます。

NX-OS スタイル CLI を使用した Cisco Tetration Analytics の設定

手順

ステップ 1 **configure terminal**

グローバル コンフィギュレーション モードを開始します。

例 :

```
apic1# configure terminal
```

ステップ 2 **analytics cluster cluster_name**

分析ポリシーを作成します。

例 :

```
apic1(config)# analytics cluster cluster1
```

ステップ 3 **flow-exporter** *server_name*

外部分析情報を設定します。

例：

```
apic1(config-analytics)# flow-exporter server1
```

ステップ 4 **destination** *ip_address*

宛先ポートを設定します。

例：

```
apic1(config-analytics-cluster-exporter)# destination 192.0.2.1
```

ステップ 5 **exit**

コマンドモードを終了します。

例：

```
# apic1(config-analytics-cluster-exporter)# exit
```

ステップ 6 **exit**

コマンドモードを終了します。

例：

```
apic1(config-analytics)# exit
```

ステップ 7 **fabric-internal**

ファブリック内部設定モードを開始します。

例：

```
apic1(config)# fabric-internal
```

ステップ 8 **template leaf-policy-group** *leaf_group_name*

リーフ ポリシー グループを定義します。

例：

```
apic1(config-fabric-internal)# template leaf-policy-group lpg1
```

ステップ 9 **inherit analytics-policy cluster** *cluster_name* **server** *server_name*

リーフ ポリシー グループに分析ポリシーを関連付けます。

例：

```
apic1(config-leaf-policy-group)# inherit analytics-policy cluster cluster1 server server1
```

ステップ 10 **exit**

コマンドモードを終了します。

例：

```
apic1(config-leaf-policy-group)# exit
```

ステップ 11 `leaf-profile leaf_profile_name`

リーフ プロファイルを定義します。

例 :

```
apicl(config-fabric-internal)# leaf-profile lp1
```

ステップ 12 `leaf-group leaf_group_name`

リーフ グループを定義します。

例 :

```
apicl(config-leaf-profile)# leaf-group lg1
```

ステップ 13 `leaf-policy-group leaf_policy_group_name`

リーフ グループにリーフ ポリシー グループに関連付けます。

例 :

```
apicl(config-leaf-group)# leaf-policy-group lpg1
```

ステップ 14 `leaf leaf_group_number`

ノードをリーフ グループに追加します。

例 :

```
apicl(config-leaf-group)# leaf 101
```

ステップ 15 `show analytics`

分析を表示します。

(注) 宛先ポートは設定可能です。UDP ポート 5640 は常にリーフ スイッチを使用し、UDP ポート 5641 はスパイン スイッチに常に使用します。

DSCP は設定できません。VA (音声を確認する) は常に使用されます。

例 :

```
apicl# show analytics
Cluster          : cluster1
Config Server Name : server1
Destination IP    : 192.0.2.1
Destination Port  : unspecified
DSCP              : VA
```

ステップ 16 `show running-config analytics`

実行コンフィギュレーションの分析を表示します。

例 :

```
apicl# show running-config analytics
# Command: show running-config analytics
# Time: Wed May 25 21:14:43 2016
analytics cluster cluster1
  flow-exporter server1
    destination 192.0.2.1
    destination-port unspecified
    dscp VA
```

```
        ip-filter-action deny
      exit
    exit
  apic1#
```



第 16 章

NetFlow の設定

- [NetFlow について \(487 ページ\)](#)
- [NX OS スタイル CLI を使用した仮想マシン ネットワーキングの NetFlow エクスポートポリシーの設定 \(488 ページ\)](#)
- [NX OS スタイル CLI を使用したノード制御ポリシーを介した NetFlow およびTetration Analytics 機能の優先順位設定 \(488 ページ\)](#)
- [NX-OS スタイル CLI を使用した NetFlow ノードポリシー の設定 \(489 ページ\)](#)
- [NX-OS スタイル CLI を使用した NetFlow インフラ セクタの設定 \(490 ページ\)](#)
- [NX-OS スタイル CLI を使用した NetFlow オーバーライドの設定 \(492 ページ\)](#)
- [NX-OS スタイルの CLI を使用した NetFlow テナント階層の設定 \(493 ページ\)](#)
- [VMware VDS の NX-OS スタイル CLI を使用して VMM ドメインで NetFlow エクスポートポリシーを利用する \(496 ページ\)](#)
- [VMware 用 NX OS スタイル CLI を使用したエンドポイント グループ上の NetFlow の有効化または無効化 \(497 ページ\)](#)

NetFlow について

NetFlowテクノロジーは、ネットワークトラフィックアカウンティング、従量制のネットワーク課金、ネットワークプランニング、そしてサービス拒絶に対する監視機能、ネットワーク監視、社外マーケティング、およびサービスプロバイダと企業顧客向け両方のデータマイニングなど、主要な一連のアプリケーションの計測基盤を効果的にします。Cisco は NetFlow エクスポートデータの収集、データ量削減、ポストプロセッシングを行う一連の NetFlow アプリケーションを提供し、エンドユーザーアプリケーションが NetFlow データへ簡単にアクセスできるようにします。この機能により、同じレベルを介したトラフィックのモニタリングを実行する、NetFlow がデータセンターを通過するトラフィックのモニタリングを有効にすると、Cisco Application Centric Infrastructure (Cisco ACI) ファブリック。

ハードウェアがレコードからコレクタに直接エクスポートする代わりに、レコードはスーパーバイザエンジンで処理され、必要な形式で標準の NetFlow コレクタにエクスポートされます。

仮想マシン ネットワーキングの NetFlow の設定についてを参照してください、『Cisco ACI Virtualization Guide』。



(注) NetFlow は EX スイッチでのみサポートされています。参照してください、 *Cisco Nexus 9000* シリーズ *ACI* モードのスイッチの *Cisco NX-OS* リリース ノート マニュアル、サポートされているリストについては、インストールされているリリースにスイッチ EX。

NX OS スタイル CLI を使用した仮想マシン ネットワーキングの NetFlow エクスポート ポリシーの設定

次の手順の例では、NX OS スタイル CLI を使用して、仮想マシン ネットワーキングの NetFlow エクスポート ポリシーを設定します。

手順

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# config
```

ステップ 2 エクスポート ポリシーを設定します。

例：

```
apic1(config)# flow vm-exporter vmExporter1 destination address 2.2.2.2 transport udp 1234
apic1(config-flow-vm-exporter)# source address 4.4.4.4
apic1(config-flow-vm-exporter)# exit
apic1(config)# exit
```

NX OS スタイル CLI を使用したノード制御ポリシーを介した NetFlow および Tetration Analytics 機能の優先順位設定

次の手順の例では、NX OS スタイル CLI を使用して、ノード コントロール ポリシーを介した NetFlow と Tetration Analytics 機能の優先順位を設定します。

手順

ステップ 1 コンフィギュレーション モードを開始します。

例 :

```
apicl# config
```

ステップ 2 ノード制御ポリシーを作成します。

例 :

```
apicl(config)# node-control policy poll
```

ステップ 3 優先順位機能として NetFlow を設定します。

例 :

```
apicl(config-node)# feature netflow
```

ステップ 4 ノードの制御ポリシー設定を終了します。

例 :

```
apicl(config-node)# end
```

ステップ 5 ノード 101 およびノード 102 にポリシーを展開します。

例 :

```
ifav-isim15-ifc1(config)# fabric-internal
ifav-isim15-ifc1(config-fabric-internal)# template leaf-policy-group lpg1
ifav-isim15-ifc1(config-leaf-policy-group)# inherit node-control-policy poll
ifav-isim15-ifc1(config-leaf-policy-group)# exit
ifav-isim15-ifc1(config-fabric-internal)# leaf-profile leafProfile1
ifav-isim15-ifc1(config-leaf-profile)# leaf-group leafgrp1
ifav-isim15-ifc1(config-leaf-group)# leaf 101
ifav-isim15-ifc1(config-leaf-group)# leaf 102
ifav-isim15-ifc1(config-leaf-group)# leaf-policy-group lpg1
ifav-isim15-ifc1(config-leaf-group)# end
```

NX-OS スタイル CLI を使用した NetFlow ノード ポリシー の設定

次の手順の例では、NX OS スタイル CLI を使用して、NetFlow ノード ポリシーを設定します。

手順

ステップ 1 コンフィギュレーション モードを開始します。

例 :

```
apicl# config
```

ステップ 2 ノード ポリシーを設定します。

例 :

```

apic1(config)# flow node-policy nodePol
apic1(config-flow-node-pol)# flow timeout collection 100
apic1(config-flow-node-pol)# flow timeout template 123
apic1(config-flow-node-pol)# exit

```

NX-OS スタイル CLI を使用した NetFlow インフラ セクタの設定

NX-OS スタイル CLI を使用して、NetFlow インフラ セクタを設定することができます。インフラ セクタは、Netflow モニタを PHY、ポート チャネル、仮想ポート チャネル、ファブリック エクステンダ (FEX)、またはポート チャネルファブリック エクステンダ (FEXPC) インターフェイスに関連付けるために使用されます。

次の例の CLI コマンドでは、NX-OS スタイル CLI を使用して NetFlow インフラ セクタを設定する方法を示します。

手順

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# config
```

ステップ 2 NetFlow エクスポート ポリシーを作成します。

例：

次のコマンドでは、宛先エンドポイントグループは、エクスポートが背後にあるエンドポイントグループです。このエンドポイントグループは、外部のレイヤ 3 エンドポイントグループにもなります。

```

apic1(config)# flow exporter infraExporter1 destination address 1.2.3.4 transpo udp 1234
apic1(config-flow-exporter)# destination epg tenant tn2 application ap2 epg epg2
apic1(config-flow-exporter)# vrf member tenant tn2 vrf vrf2
apic1(config-flow-exporter)# version v9
apic1(config-flow-exporter)# source address 1.1.1.1
apic1(config-flow-exporter)# exit

```

ステップ 3 2 番目の NetFlow エクスポート ポリシーを作成します。

例：

次のコマンドでは、宛先エンドポイントグループはエクスポートが背後に存在するエンドポイントであり、外部レイヤ 3 エンドポイントグループのケースです。

```

apic1(config)# flow exporter infraExporter2
apic1(config-flow-exporter)# transport udp 9990
apic1(config-flow-exporter)# destination address 2001:db5:a0c:1f0::2
apic1(config-flow-exporter)# destination external-l3 epg tenant tn2 vrf v2 epg
accounting-inst

```

```
apic1(config-flow-exporter)# vrf member tenant tn2 vrf vrf2
apic1(config-flow-exporter)# version v5
apic1(config-flow-exporter)# source address 2001:db8:a0b:12f0::1
apic1(config-flow-exporter)# exit
```

ステップ 4 NetFlow レコード ポリシーを作成します。

例 :

```
apic1(config)# flow record infraRecord1
apic1(config-flow-record)# match dst-ip
apic1(config-flow-record)# match dst-ipv4
apic1(config-flow-record)# match dst-ipv6
apic1(config-flow-record)# match dst-mac
apic1(config-flow-record)# match dst-port
apic1(config-flow-record)# match ethertype
apic1(config-flow-record)# match proto
apic1(config-flow-record)# match src-ip
apic1(config-flow-record)# match src-ipv4
apic1(config-flow-record)# match src-ipv6
apic1(config-flow-record)# match src-mac
apic1(config-flow-record)# match src-port
apic1(config-flow-record)# match tos
apic1(config-flow-record)# match vlan
apic1(config-flow-record)# collect count-bytes
apic1(config-flow-record)# collect count-pkts
apic1(config-flow-record)# collect pkt-disp
apic1(config-flow-record)# collect sampler-id
apic1(config-flow-record)# collect src-intf
apic1(config-flow-record)# collect tcp-flags
apic1(config-flow-record)# collect ts-first
apic1(config-flow-record)# collect ts-recent
apic1(config-flow-record)# exit
```

ステップ 5 NetFlow モニタ ポリシーを作成します。

例 :

```
apic1(config)# flow monitor infraMonitor1
apic1(config-flow-monitor)# record infraRecord1
apic1(config-flow-monitor)# exporter infraExporter1
apic1(config-flow-monitor)# exporter infraExporter2
apic1(config-flow-monitor)# exit
```

最大 2 個のエクスポートを接続できます。

ステップ 6 インターフェイス ポリシー グループを作成します (AccPortGrp)。

例 :

```
apic1(config)# template policy-group pgl
apic1(config-pol-grp-if)# ip flow monitor infraMonitor1
apic1(config-pol-grp-if)# ipv6 flow monitor infraMonitor2
apic1(config-pol-grp-if)# exit
```

アドレス ファミリーごとに 1 個のモニタ ポリシーを設定できます (IPv4 および IPv6)。

ステップ 7 ノード プロファイルおよびインフラ セレクタを作成します。

例 :

```
apic1(config)# leaf-profile lp1
apic1(config-leaf-profile)# leaf-group lg1
apic1(config-leaf-group)# leaf 101
```

```

apic1(config-leaf-profile)# exit
apic1(config)# leaf-interface-profile lip1
apic1(config-leaf-if-profile)# exit
apic1(config)# leaf-interface-profile lip1
apic1(config-leaf-if-profile)# leaf-interface-group lig1
apic1(config-leaf-if-group)# interface ethernet 1/5
apic1(config-leaf-if-profile)# policy-group pg1
apic1(config-leaf-if-profile)# exit
apic1(config-leaf-profile)# exit

```

ステップ 8 ポート チャネル ポリシー グループを作成します (AccBndlGrp)。

例：

```

apic1(config)# template port-channel po6
apic1(config-if)# ip flow monitor infraMonitor1
apic1(config-if)# ipv6 flow monitor infraMonitor1
apic1(config-if)# exit
apic1(config-leaf-profile)# leaf-profile lp2
apic1(config-leaf-group)# leaf-group lg2
apic1(config-leaf-profile)# leaf 101
apic1(config-leaf-profile)# exit
apic1(config)# leaf-interface-profile lip2
apic1(config-leaf-if-profile)# exit
apic1(config)# leaf-interface-profile lip2
apic1(config-leaf-if-profile)# leaf-interface-group lig2
apic1(config-leaf-if-group)# interface ethernet 1/6
apic1(config-leaf-if-profile)# channel-group po6
apic1(config-leaf-if-profile)# exit

```

アドレス ファミリごとに 1 個のモニタ ポリシーを設定できます (IPv4 および IPv6)。インターフェイスでは、Vpc こともできます。

NX-OS スタイル CLI を使用した NetFlow オーバーライドの設定

次の procedure NetFlow オーバーライド NX OS スタイル CLI を使用して設定を示します。

手順

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# config
```

ステップ 2 オーバーライドを作成します。

例：

```

apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant tn2 vrf vrf2
apic1(config-leaf)# exit

```

```
apicl(config)# interface ethernet 1/15
apicl(config-if)# ip flow monitor infraMonitor1
apicl(config-if)# ipv6 flow monitor infraMonitor2
apicl(config-if)# exit
apicl(config)# exit
apicl# exit
```

アドレス ファミリ (IPv4 および IPv6) ごとに 1 つのモニタ ポリシーを設定できます。インターフェイスでは、Vpc こともできます。

NX-OS スタイルの CLI を使用した NetFlow テナント階層の設定

次の手順の例では、NetFlow テナント階層を設定する NX OS スタイル CLI を使用します。

手順

ステップ 1 コンフィギュレーション モードを開始します。

例 :

```
apicl# config
```

ステップ 2 テナントとブリッジ ドメインを作成し、VRF に追加します。

例 :

```
apicl(config)# tenant tn2
apicl(config-tenant)# vrf context vrf2
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# bridge-domain bd2
apicl(config-tenant-bridge-domain)# vrf member vrf2
apicl(config-tenant-bridge-domain)# exit
apicl(config-tenant)# bridge-domain bd3
apicl(config-tenant-bridge-domain)# vrf member vrf2
apicl(config-tenant-bridge-domain)# exit
```

ステップ 3 エクスポートが存在する背後にアプリケーション エンドポイント グループを作成します。

例 :

```
apicl(config-tenant)# application ap2
apicl(config-tenant-app)# epgr epg2
apicl(config-tenant-app)# bridge-domain member bd2
apicl(config-tenant-app-bridge-domain)# exit
apicl(config-tenant-app)# exit
```

ステップ 4 エクスポートが存在する背後に 2 番目のアプリケーション エンドポイント グループを作成します。

例 :

```

apic1(config-tenant)# application ap3
apic1(config-tenant-app)# epq epq3
apic1(config-tenant-app)# bridge-domain member bd3
apic1(config-tenant-app-bridge-domain)# exit
apic1(config-tenant-app)# exit

```

ステップ 5 ブリッジドメインに NetFlow モニタ ポリシーを接続します。

例 :

```

apic1(config)# interface bridge-domain bd2
apic1(config-if)# ipv6 flow monitor tnMonitor1
apic1(config-if)# ip flow monitor tnMonitor1
apic1(config-if)# layer2-switched flow monitor tnMonitor1
apic1(config-if)# exit
apic1(config)# interface bridge-domain bd3
apic1(config-if)# ipv6 flow monitor tnMonitor1
apic1(config-if)# ip flow monitor tnMonitor1
apic1(config-if)# exit

```

アドレス ファミリ (IPv4 および IPv6) ごとに 1 つのモニタ ポリシーを設定できます。インターフェイスでは Vpc にすることもできます。

ステップ 6 Netflow エクスポート ポリシーを作成します。

例 :

次のコマンドでは、宛先エンドポイントグループは、エクスポートの背後にあるエンドポイントグループです。このエンドポイントグループは、外部のレイヤ 3 エンドポイントグループにすることもできます。

```

apic1(config)# flow exporter tnExporter1
apic1(config-flow-exporter)# transport udp 1234
apic1(config-flow-exporter)# destination address 2.2.2.2
apic1(config-flow-exporter)# destination epq tenant tn2 application ap2 epq epq2
apic1(config-flow-exporter)# vrf member tenant tn2 vrf vrf2
apic1(config-flow-exporter)# version v9
apic1(config-flow-exporter)# source address 1.1.1.1
apic1(config-flow-exporter)# exit

```

ステップ 7 2 番目の Netflow エクスポート ポリシーを作成します。

例 :

次のコマンドでは、宛先エンドポイントグループは、エクスポートの背後にあるエンドポイントグループであり、このケースでは外部レイヤ 3 エンドポイントグループです。

```

apic1(config)# flow exporter tnExporter2
apic1(config-flow-exporter)# transport udp 9990
apic1(config-flow-exporter)# destination address 2001:db5:a0c:1f0::2
apic1(config-flow-exporter)# destination external-l3 epq tenant tn2 vrf v2 epq
accounting-inst
apic1(config-flow-exporter)# vrf member tenant tn2 vrf vrf2
apic1(config-flow-exporter)# version v5
apic1(config-flow-exporter)# source address 2001:db8:a0b:12f0::1
apic1(config-flow-exporter)# exit

```

ステップ 8 NetFlow レコード ポリシーを作成します。

例 :

```
apic1(config)# flow record tnRecord1
apic1(config-flow-record)# match dst-ip
apic1(config-flow-record)# match dst-ipv4
apic1(config-flow-record)# match dst-ipv6
apic1(config-flow-record)# match dst-mac
apic1(config-flow-record)# match dst-port
apic1(config-flow-record)# match ethertype
apic1(config-flow-record)# match proto
apic1(config-flow-record)# match src-ip
apic1(config-flow-record)# match src-ipv4
apic1(config-flow-record)# match src-ipv6
apic1(config-flow-record)# match src-mac
apic1(config-flow-record)# match src-port
apic1(config-flow-record)# match tos
apic1(config-flow-record)# match vlan
apic1(config-flow-record)# collect count-bytes
apic1(config-flow-record)# collect count-pkts
apic1(config-flow-record)# collect pkt-disp
apic1(config-flow-record)# collect sampler-id
apic1(config-flow-record)# collect src-intf
apic1(config-flow-record)# collect tcp-flags
apic1(config-flow-record)# collect ts-first
apic1(config-flow-record)# collect ts-recent
apic1(config-flow-record)# exit
```

ステップ 9 NetFlow モニタ ポリシーを作成します。

例 :

```
apic1(config)# flow monitor tnMonitor1
apic1(config-flow-monitor)# record tnRecord1
apic1(config-flow-monitor)# exporter tnExporter1
apic1(config-flow-monitor)# exporter tnExporter2
apic1(config-flow-monitor)# exit
```

最大 2 つのエクスポートを接続できます。

ステップ 10 VLAN のドメインに VLAN を追加し、リーフ ノードの VRF を設定します。

例 :

```
apic1(config)# vlan-domain dom1
apic1(config-vlan)# vlan 5-100
apic1(config-vlan)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant tn2 vrf vrf2
apic1(config-leaf-vrf)# exit
```

ステップ 11 ブリッジドメインを展開するためインターフェイスでエンドポイントグループを展開します。

例 :

```
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant tn2 application ap2 epg
epg2
apic1(config-leaf-if)# exit
```

ステップ 12 インターフェイスで別のエンドポイント グループを展開します。

例 :

```
apic1(config-leaf)# interface ethernet 1/11
apic1(config-leaf-if)# vlan-domain member dom1
```

```
apic1(config-leaf-if)# switchport trunk allowed vlan 11 tenant tn2 application ap3 epg
epg3
apic1(config-leaf-if)# exit
```

ステップ 13 モニタ ポリシーをサブインターフェイスに接続します。

例 :

```
apic1(config-leaf)# interface ethernet 1/20
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/20.20
apic1(config-leaf-if)# vrf member tenant tn2 vrf vrf2
apic1(config-leaf-if)# ipv6 address 20::1/64 preferred
apic1(config-leaf-if)# ipv6 flow monitor tnMonitor1
apic1(config-leaf-if)# ip flow monitor tnMonitor2
apic1(config-leaf-if)# exit
```

ステップ 14 スイッチ仮想インターフェイス (SVI) にモニタ ポリシーを接続します。

例 :

```
apic1(config-leaf)# interface vlan 30
apic1(config-leaf-if)# vrf member tenant tn2 vrf vrf2
apic1(config-leaf-if)# ipv6 address 64::1/64 preferred
apic1(config-leaf-if)# ip flow monitor tnMonitor1
apic1(config-leaf-if)# ip6 flow monitor tnMonitor1
apic1(config-leaf-if)# exit
```

ステップ 15 SVI をレイヤ 2 インターフェイスに関連付けます。

例 :

```
apic1(config-leaf)# interface ethernet 1/30
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 30 tenant tn2 external-svi
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# exit
```

VMware VDS の NX-OS スタイル CLI を使用して VMM ドメインで NetFlow エクスポート ポリシーを利用する

次の手順では、VMM ドメインで NetFlow エクスポート ポリシーを消費するために、NX OS スタイル CLI を使用します。

手順

ステップ 1 コンフィギュレーション モードを開始します。

例 :

```
apic1# config
```


ステップ2 NetFlow エクスポート ポリシーを消費します。

例：

```
apicl(config)# vmware-domain mininet
apicl(config-vmware)# configure-dvs
apicl(config-vmware-dvs)# flow exporter vmExporter1
apicl(config-vmware-dvs-flow-exporter)# active-flow-timeout 62
apicl(config-vmware-dvs-flow-exporter)# idle-flow-timeout 16
apicl(config-vmware-dvs-flow-exporter)# sampling-rate 1
apicl(config-vmware-dvs-flow-exporter)# exit
apicl(config-vmware-dvs)# exit
apicl(config-vmware)# exit
apicl(config)# exit
```

VMware 用 NX OS スタイル CLI を使用したエンドポイントグループ上の NetFlow の有効化または無効化

NX-OS スタイルの CLI を使用して、エンドポイント グループでの NetFlow を有効または無効にするには、次の手順を実行します。

手順

ステップ1 NetFlow の有効化：

例：

```
apicl# config
apicl(config)# tenant tn1
apicl(config-tenant)# application appl
apicl(config-tenant-app)# epg epg1
apicl(config-tenant-app-epg)# vmware-domain member mininet
apicl(config-tenant-app-epg-domain)# flow monitor enable
apicl(config-tenant-app-epg-domain)# exit
apicl(config-tenant-app-epg)# exit
apicl(config-tenant-app)# exit
apicl(config-tenant)# exit
apicl(config)# exit
```

ステップ2 （任意） NetFlow を使用しない場合は、この機能を無効にします。

例：

```
apicl(config-tenant-app-epg-domain)# no flow monitor enable
```




第 17 章

ファームウェアの管理

- [ファームウェアの管理 \(499 ページ\)](#)
- [リポジトリ イメージの追加または削除 \(499 ページ\)](#)
- [カタログ ファームウェアの変更 \(500 ページ\)](#)
- [コントローラ ファームウェアのアップグレード \(501 ページ\)](#)
- [スイッチ ファームウェアのアップグレード \(503 ページ\)](#)

ファームウェアの管理

各ファームウェア イメージには、サポートされるタイプおよびスイッチ モデルを識別する互換性カタログが含まれます。APIC は、ファームウェア イメージ、スイッチ タイプ、およびそのファームウェア イメージを使用できるモデルのカタログを保持しています。デフォルトの設定では、互換性カタログに適合しない場合、ファームウェアの更新が拒否されます。

APIC には、互換性カタログ、コントローラのファームウェア イメージおよびスイッチ イメージのイメージ リポジトリがあります。管理者は、外部 HTTP サーバまたは SCP サーバから新しいファームウェア イメージを APIC イメージ リポジトリにダウンロードできます。



(注) スイッチをアップグレードするには、その前に、APIC のアップグレードが完了し、ヘルス ステータスが [Fully Fit] になっている必要があります。

リポジトリ イメージの追加または削除

手順

	コマンドまたはアクション	目的
ステップ 1	firmware repository add <i>absolute-image-path</i> 例 :	リポジトリにファームウェア イメージを追加します。

	コマンドまたはアクション	目的
	<code>apicl# firmware repository add /home/admin/aci-catalog-dk9.1.2.1b.bin</code>	
ステップ 2	firmware repository delete image 例 : <code>apicl# firmware repository delete aci-catalog-dk9.1.2.1a.bin</code>	リポジトリからファームウェア イメージを削除します。

例

```
apicl# firmware repository add /home/admin/aci-catalog-dk9.1.2.1b.bin
apicl# firmware repository delete aci-catalog-dk9.1.2.1a.bin
```

カタログ ファームウェアの変更

この手順は、リポジトリからカタログファームウェアバージョンを選択する方法を示します。

手順

	コマンドまたはアクション	目的
ステップ 1	show firmware repository [detail] 例 : <code>apicl# show firmware repository</code>	リポジトリにあるファームウェア イメージを表示します。The detail option displays additional information such as MD5 checksum, release date, and download date.
ステップ 2	configure 例 : <code>apicl# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 3	firmware 例 : <code>apicl(config)# firmware</code>	ファームウェア アップグレード コンフィギュレーション モードを開始します。
ステップ 4	(任意) show version 例 : <code>apicl(config-firmware)# show version</code>	現在インストールされているコントローラとスイッチ ファームウェア バージョンが表示されます。
ステップ 5	catalog-version firmware-name 例 : <code>apicl(config-firmware)# catalog-version aci-catalog-dk9.1.2.1b.bin</code>	リポジトリの使用可能なイメージにカタログ バージョンを変更します。

例

この例では、リポジトリからカタログファームウェアバージョンを選択する方法を示します。

```
apicl# show firmware repository
Name                               Type      Version  Size (MB)
-----
aci-catalog-dk9.1.2.1a.bin         catalog   1.2.1a   0.023
aci-catalog-dk9.1.2.1b.bin         catalog   1.2.1b   0.025

apicl# configure
apicl(config)# firmware
apicl(config-firmware)# catalog-version aci-catalog-dk9.1.2.1b.bin
```

コントローラ ファームウェアのアップグレード

ランダムにコントローラのアップグレード。各 APIC コントローラのアップグレードには約 10 分かかります。コントローライメージがアップグレードされると、そのコントローラはクラスタからドロップし、クラスタ内の他の APIC コントローラが動作中に新しいバージョンで再起動します。コントローラが再起動すると、クラスタに再び参加します。その後、クラスタが収束し、次のコントローラのイメージのアップグレードを開始します。

カタログ ファームウェア イメージは、APIC コントローラ イメージがアップグレードされるとアップグレードされます。カタログ ファームウェア イメージを個別にアップグレードする必要はありません。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	firmware 例 : apicl(config)# firmware	ファームウェア アップグレード コンフィギュレーション モードを開始します。
ステップ 3	(任意) show version 例 : apicl(config-firmware)# show version	現在インストールされているコントローラとスイッチ ファームウェア バージョンが表示されます。
ステップ 4	controller-group 例 :	コントローラ アップグレード コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	<code>apic1(config-firmware)# controller-group</code>	
ステップ 5	<code>firmware-version firmware-name</code> 例 : <code>apic1(config-firmware-controller)# firmware-version aci-apic-dk9.1.2.1b.iso</code>	アップグレードの目的のバージョンを指定します。
ステップ 6	<code>[no] time start time</code> 例 : <code>apic1(config-firmware-controller)# time start 2016:jan:01:12:01</code>	[[[yyyy:]mmm:]dd:]HH:MM 形式で開始時刻を設定します。日付はオプションです。 (注) To upgrade the controllers immediately, return to EXEC mode and type the command firmware upgrade controller-group .

例

次に、コントローラをアップグレードする例を示します。

```
apic1# show controller
Fabric Name       : mininet
Operational Size  : 3
Cluster Size     : 3
Time Difference   : 0
Fabric Security Mode : permissive
```

ID	Address	In-Band Address	OOB Address	Version	Flags	Serial Number	Health
1*	10.0.0.1	192.168.11.1	192.168.10.1	1.2(1a)	crva	TEP-1-1	fully-fit
2	10.0.0.2	192.168.11.2	192.168.10.2	1.2(1a)	crva	TEP-1-2	fully-fit
3	10.0.0.3	192.168.11.3	192.168.10.3	1.2(1a)	crva	TEP-1-3	fully-fit

Flags - c:Commissioned | r:Registered | v:Valid Certificate | a:Approved

```
apic1# configure
apic1(config)# firmware
apic1(config-firmware)# show version
```

Role	Id	Name	Version
controller	1	apic1	1.2(1a)
controller	2	apic2	1.2(1a)
controller	3	apic3	1.2(1a)
leaf	101	leaf1	n9000-11.2(1a)
leaf	102	leaf2	n9000-11.2(1a)
leaf	103	leaf2	n9000-11.2(1a)

```

spine      201      spine1      n9000-11.2 (1a)
spine      202      spine2      n9000-11.2 (1a)

apic1(config-firmware)# controller-group
apic1(config-firmware-controller)# firmware-version aci-apic-dk9.1.2.1b.iso
apic1(config-firmware-controller)# time start 2016:jan:01:12:01

```

スイッチ ファームウェアのアップグレード

始める前に

アップグレードをいつ実行するのかを指定するには、スケジューラが存在する必要があります。



(注) スイッチをアップグレードするには、その前に、APIC のアップグレードが完了し、ヘルス ステータスが [Fully Fit] になっている必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	firmware 例 : apic1(config)# firmware	ファームウェア アップグレード コンフィギュレーション モードを開始します。
ステップ 3	[no] switch-group group-name 例 : apic1(config-firmware)# switch-group mySwitchGroup5	スイッチグループを作成（または削除）し、スイッチアップグレード コンフィギュレーション モードを開始します。
ステップ 4	[no] switch node-id-or-name[,node-id-or-name,...] 例 : apic1(config-firmware-switch)# switch leaf1-leaf3,leaf6 apic1(config-firmware-switch)# no switch leaf4,leaf5	アップグレードのため、スイッチグループに対してスイッチまたはスイッチのリストの追加（または削除）を行います。ノード ID（101 など）または名前（spine1 など）を指定できます。カンマを使用すると、複数のスイッチを指定することができます。

	コマンドまたはアクション	目的
ステップ 5	firmware-version <i>firmware-name</i> 例 : <pre>apic1(config-firmware-switch)# firmware-version aci-apic-dk9.11.2.1a.bin</pre>	ターゲットのファームウェア イメージを指定します。
ステップ 6	[no] run-mode {pause-never pause-on-failure} 例 : <pre>apic1(config-firmware-switch)# run-mode pause-on-failure</pre>	現在のノードセットでアップグレードが失敗した場合に次のノードセットに進むかどうかを指定します。
ステップ 7	schedule <i>scheduler-name</i> 例 : <pre>apic1(config-firmware-switch)# schedule myNextSunday</pre>	スケジューラをアップグレードに割り当てます。すでに定義されているスケジューラの名前を入力します。 (注) To upgrade the switch group immediately, return to EXEC mode and type the command firmware upgrade switch-group .
ステップ 8	[no] scheduler pause 例 : <pre>apic1(config-firmware-switch)# scheduler pause apic1(config-firmware-switch)# no scheduler pause</pre>	メンテナンス ポリシー スケジューラを一時停止します。Use the [no] prefix to resume.
ステップ 9	show running-config 例 : <pre>apic1(config-firmware-switch)# show run</pre>	設定を表示します。

例

次に、3 つのリーフ スイッチのファームウェアをアップグレードする例を示します。

```
apic1# configure
apic1(config)# firmware
apic1(config-firmware)# switch-group mySwitchGroup5
apic1(config-firmware-switch)# switch leaf1,leaf3,leaf6
apic1(config-firmware-switch)# no switch leaf4,leaf5
apic1(config-firmware-switch)# firmware-version aci-apic-dk9.1.1.3f.bin
apic1(config-firmware-switch)# run-mode pause-on-failure
apic1(config-firmware-switch)# schedule myNextSunday
apic1(config-firmware-switch)# show run
# Command: show running-config firmware switch-group mySwitchGroup5
# Time: Fri Nov 6 23:55:35 2015
```



```
firmware
  switch-group mySwitchGroup5
    switch 101
    switch 102
    switch 103
    switch 106
    schedule myNextSunday
  exit
exit
```




第 18 章

スナップショットによる設定の管理

- [構成管理とスナップショットについて \(507 ページ\)](#)
- [スナップショットのエクスポート \(507 ページ\)](#)
- [スナップショットのインポート \(509 ページ\)](#)
- [スナップショットを使用したロールバック設定 \(510 ページ\)](#)
- [スナップショット ファイルのリモート パスへのアップロードまたはダウンロード \(512 ページ\)](#)
- [スナップショット ファイルとジョブの管理 \(513 ページ\)](#)

構成管理とスナップショットについて

ローカルのコントローラ管理フォルダとの間で、構成アーカイブ（スナップショット）をエクスポートおよびインポートすることによって、システム構成をバックアップおよび復元することができます。構成の変更を加える前後にスナップショットをエクスポートすることにより、2つのスナップショット間に適用された構成変更をロールバックすることができます。

また、リモート サーバとの間でスナップショット ファイルをアップロードおよびダウンロードすることもできます。

各スナップショットのアクション（エクスポート、インポート、ロールバック、アップロードおよびダウンロード）は、アクションのためのポリシーを作成し、ジョブとしてアクションをトリガーすることによって行われます。エクスポートのアクションは、将来の時刻または定期的に行うようにスケジュールすることもできます。インポート、エクスポートおよびロールバックのジョブは並列実行できません。ジョブがすでに実行中の場合、新しいジョブをトリガーすることはできません。

スナップショットのエクスポート

始める前に

スケジュールに従ってスナップショットをエクスポートする場合、エクスポートポリシーを設定する前にスケジューラを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] snapshot export policy-name 例 : apicl(config)# snapshot export myExportPolicy	スナップショットをエクスポートするポリシーを作成します。
ステップ 3	format {xml json} 例 : apicl(config-export)# format json	エクスポートされたコンフィギュレーション ファイルのデータ形式を指定します。デフォルトは
ステップ 4	(任意) [no] schedule schedule-name 例 : apicl(config-export)# schedule EveryEightHours	スナップショットをエクスポートする既存のスケジューラを指定します。
ステップ 5	(任意) [no] target [infra fabric tenant-name] 例 : apicl(config-export)# target tenantExampleCorp	エクスポートのターゲットを割り当てます。ファブリック、インフラ、特定のテナント、またはなしにすることができます。ターゲットを指定しない場合、すべての設定情報がエクスポートされます。デフォルトではターゲットはありません。
ステップ 6	(任意) [no] remote path remote-path-name 例 : apicl(config-export)# remote path myBackupServer	ファイルが送信される設定済みのリモートパスの名前を指定します。リモートパスを指定しない場合、ファイルはコントローラ内のフォルダにローカルにエクスポートされます。デフォルトではリモートパスはありません。
ステップ 7	end 例 : apicl(config-export)# end	EXEC モードに戻ります。
ステップ 8	必須: trigger snapshot export policy-name 例 : apicl# trigger snapshot export myExportPolicy	スナップショット エクスポート タスクを実行します。エクスポート ポリシーにスケジューラを設定する場合、すぐにエクスポートする場合を除きこの手順は不要です。

例

次に、特定のテナント設定について JSON 形式のスナップショット ファイルの定期的なエクスポートを設定する例を示します。

```
apicl# configure
apicl(config)# snapshot export myExportPolicy
apicl(config-export)# format json
apicl(config-export)# target tenantExampleCorp
apicl(config-export)# schedule EveryEightHours
```

スナップショットのインポート

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] snapshot import <i>policy-name</i> 例 : apicl(config)# snapshot import myImportPolicy	スナップショットをインポートするポリシーを作成します。
ステップ 3	file <i>filename</i> 例 : apicl(config-import)# file ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz	インポートするファイルの名前を指定します。
ステップ 4	action {merge replace} 例 : apicl(config-import)# action replace	インポートされた構成設定を現在の設定とマージするかどうか、またはインポートされた設定で現在の設定を完全に置き換えるかどうかを指定します。
ステップ 5	[no] mode {atomic best-effort} 例 : apicl(config-import)# mode atomic	インポートされた設定を適用するときに設定エラーをインポート プロセスでどのように処理するかを指定します。ベストエフォート型のインポート モードではアーカイブ内の個々の設定エラーをスキップし、アトミック モードでは設定エラーがあるとインポートをキャンセルします。

	コマンドまたはアクション	目的
ステップ 6	(任意) [no] remote path <i>remote-path-name</i> 例 : <pre>apic1(config-import)# remote path myBackupServer</pre>	ファイルをインポートする設定済みリモートパスの名前を指定します。リモートパスを指定しない場合、ファイルはコントローラ内のフォルダからローカルにインポートされます。デフォルトではリモートパスはありません。
ステップ 7	end 例 : <pre>apic1(config-import)# end</pre>	EXEC モードに戻ります。
ステップ 8	必須: trigger snapshot import policy-name 例 : <pre>apic1# trigger snapshot import myImportPolicy</pre>	スナップショット インポート タスクを実行します。

例

次に、スナップショットファイルのインポートの設定および実行を行って現在の設定を置き換える例を示します。

```
apic1# show snapshot files
File      : ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
Created   : 2015-11-21T01:00:21.167+00:00
Root      :
Size      : 22926

apic1# configure
apic1(config)# snapshot import myImportPolicy
apic1(config-import)# file ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
apic1(config-import)# action replace
apic1(config-import)# mode atomic
apic1(config-import)# end
apic1# trigger snapshot import myImportPolicy
```

スナップショットを使用したロールバック設定

ロールバック機能では、1つのスナップショットアーカイブと以降のスナップショットアーカイブの間に行った変更を戻す、「元に戻す」機能を提供します。ロールバックでは、ローカルに保存されたスナップショットファイルのみがサポートされます。実装する前に、必要に応じてプレビュー モードを有効にして、ロールバックを生成および表示できます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] snapshot rollback <i>policy-name</i> 例 : <code>apic1(config)# snapshot rollback myRollbackPolicy</code>	スナップショットを使用してロールバックのポリシーを作成します。
ステップ 3	first-file <i>filename</i> 例 : <code>apic1(config-rollback)# first-file oe2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz</code>	以前のファイルの名前を指定します。
ステップ 4	second-file <i>filename</i> 例 : <code>apic1(config-rollback)# second-file oe2_DailyAutoBackup-2015-11-21T09-00-21.tar.gz</code>	以降のファイルの名前を指定します。
ステップ 5	[no] preview 例 : <code>apic1(config-rollback)# preview</code>	(オプション) ロールバックの変更が生成されプレビューされますが、適用されないように指定します。プレビューモードが有効にすると、設定の変更は行われません。ロールバックの変更をプレビューした後、 no preview コマンドを使用してプレビューモードを終了し、 trigger snapshot rollback コマンドを再入力するときに適用するロールバックを有効にします。
ステップ 6	end 例 : <code>apic1(config-rollback)# end</code>	EXEC モードに戻ります。
ステップ 7	必須: trigger snapshot rollback <i>policy-name</i> 例 : <code>apic1# trigger snapshot rollback myRollbackPolicy</code>	スナップショット ロールバック タスクを実行します。

例

この例では、最初にプレビューすることなくロールバックを設定し、実行する方法を示します。

```
apic1# show snapshot files
File      : ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
Created   : 2015-11-21T01:00:21.167+00:00
Root      :
Size      : 22926

File      : ce2_DailyAutoBackup-2015-11-21T09-00-21.tar.gz
Created   : 2015-11-21T09:00:24.025+00:00
Root      :
Size      : 23588

apic1# configure
apic1(config)# snapshot rollback myRollbackPolicy
apic1(config-rollback)# first-file ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
apic1(config-rollback)# second-file ce2_DailyAutoBackup-2015-11-21T09-00-21.tar.gz
apic1(config-rollback)# end
apic1# trigger snapshot rollback myRollbackPolicy
```

スナップショット ファイルのリモートパスへのアップロードまたはダウンロード

ローカルストレージからリモートパスにスナップショットアーカイブファイルをアップロードできます。また、リモートパスからスナップショットアーカイブファイルをダウンロードすることもできます。

始める前に

ファイルを受信するリモートパスを設定します。[ファイルエクスポートのリモートパスの設定 \(530 ページ\)](#) を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] snapshot {upload download} policy-name remote-path-name 例 :	リモートパスを使用してスナップショット ファイルをアップロードまたはダウンロードするポリシーを作成します。

	コマンドまたはアクション	目的
	<code>apic1(config)# snapshot upload myUpPolicy</code>	
ステップ 3	remote path <i>remote-path-name</i> 例 : <code>apic1(config-upload)# remote path myBackupServer</code>	スナップショット ファイルが送信される設定済みのリモートパスの名前を指定します。
ステップ 4	file <i>filename</i> 例 : <code>apic1(config-upload)# file ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz</code>	送信されるスナップショット ファイルの名前を指定します。
ステップ 5	end 例 : <code>apic1(config-upload)# end</code>	EXEC モードに戻ります。
ステップ 6	trigger snapshot {upload download} <i>policy-name</i> 例 : <code>apic1# trigger snapshot upload myUpPolicy</code>	スナップショットのアップロードまたはダウンロードタスクを実行します。

例

次に、スナップショットファイルのリモートパスへのアップロードを設定および実行する例を示します。

```
apic1# show snapshot files
File      : ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
Created   : 2015-11-21T01:00:21.167+00:00
Root      :
Size      : 22926

apic1# configure
apic1(config)# snapshot upload myUpPolicy
apic1(config-upload)# remote path myBackupServer
apic1(config-upload)# file ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
apic1(config-upload)# end
apic1# trigger snapshot upload myUpPolicy
```

スナップショット ファイルとジョブの管理

次のコマンドは、スナップショット ファイルやジョブを管理するために使用できます。

コマンド	説明
clear snapshot file <i>filename</i>	ローカル ストレージからスナップショット ファイルを削除します。
clear snapshot job <i>job-name</i>	履歴からスナップショット ジョブを削除します。
show snapshot files	ローカルストレージのスナップショット ファイルが表示されます。
show snapshot jobs	最近のスナップショット タスクが表示されます。
show snapshot active jobs	現在アクティブなスナップショット タスクが表示されます。

例

この例では、スナップショットファイルとスナップショットジョブ履歴を表示する方法を示しています。

```

apic1# show snapshot files
File       : ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
Created    : 2015-11-21T01:00:21.167+00:00
Root       :
Size       : 22926

File       : ce2_DailyAutoBackup-2015-11-21T09-00-21.tar.gz
Created    : 2015-11-21T09:00:24.025+00:00
Root       :
Size       : 23588

apic1# show snapshot jobs
Type       : export
Run        : 2015-11-21T01-00-17
State      : success
Details    : Success
File Name  : ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz

Type       : export
Run        : 2015-11-21T09-00-21
State      : success
Details    : Success
File Name  : ce2_DailyAutoBackup-2015-11-21T09-00-21.tar.gz

Type       : rollback
Run        : 2015-11-22T00-25-06
State      : running
Details    :
File Name  : not applicable

apic1# clear snapshot job 2015-11-22T00-25-06

```



第 19 章

モニタリングの設定

- [Syslog の設定 \(515 ページ\)](#)
- [Call Home の設定 \(518 ページ\)](#)
- [TACACS 外部ログの設定 \(526 ページ\)](#)
- [NX-OS スタイルの CLI を使用したオンデマンドの techsupport ファイルの送信 \(528 ページ\)](#)
- [ファイルエクスポートのリモートパスの設定 \(530 ページ\)](#)
- [モニタリングの Show コマンドの使用法 \(531 ページ\)](#)
- [SNMP の設定 \(537 ページ\)](#)
- [CLI を使用した SNMP ポリシーの設定 \(539 ページ\)](#)
- [Smart Callhome の設定 \(540 ページ\)](#)

Syslog の設定

ロギング サーバグループの設定

ACI ファブリックでは、1 つ以上のロギング サーバグループは 1 つ以上のロギング宛先サーバで設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	logging server-group server-group-name 例 : apic1(config)# logging server-group myLoggingGroup	監視用のサーバのグループを設定します。

	コマンドまたはアクション	目的
ステップ 3	[no] description <i>text</i> 例 : <pre>apic1(config-logging)# logging description "This is my logging server group"</pre>	指定します。
ステップ 4	[no] console [severity {alerts critical emergencies}] 例 : <pre>apic1(config-logging)# console severity critical</pre>	コンソールへのロギングをイネーブルにして（スイッチ用のみ）、オプションでロギングの最小の重大度レベルを設定します。
ステップ 5	[no] logfile [severity {alerts critical debugging emergencies errors information notifications warnings}] 例 : <pre>apic1(config-logging)# logfile severity critical</pre>	ログファイルへのロギングをイネーブルにして、オプションでロギングの最小の重大度レベルを設定します。
ステップ 6	[no] server <i>ip</i> アドレスまたはホスト名[facility ローカルレベル]mgmtepg [severity 重大度レベル] [inb {oob}] [port ポート番号] 例 : <pre>apic1(config-logging)# server reach.example.com level local4 mgmtepg inb port 514</pre>	宛先ロギング サーバを追加して、オプションでロギングの最小の重大度レベルを設定します。 • facility—Local facility in the form local<i>n</i> • severity : ロギングの最小の重大度レベル。Can be one of the options shown in the logfile command. • mgmt—Management endpoint group, either inb (inband) or oob (out of band). • port : ログング サーバのサービスポート番号。

例

この例では、syslog 宛先サーバ グループを設定する方法を示します。

```
apic1# configure
apic1(config)# logging server-group myLoggingGroup
apic1(config-logging)# logging description "This is my logging server group"
apic1(config-logging)# console severity critical
apic1(config-logging)# logfile severity critical
apic1(config-logging)# server reach.example.com level local4 mgmtepg inb port 514
```

次のタスク

ロギング宛先としてこのロギング サーバグループを使用して syslog を設定します。

Syslog の設定

システム ログ メッセージを受信してモニタするためには、syslog 宛先（コンソール、ローカル ファイル、または syslog サーバを実行している 1 つ以上のリモート ホスト）を指定する必要があります。また、コンソールに表示されるか、ファイルまたはホストによってキャプチャされるメッセージの重大度の最小値を指定できます。

始める前に

syslog メッセージの送信先サーバを含むロギング サーバグループを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	syslog common 例 : <code>apic1(config)# syslog common</code>	syslog 共通のポリシー コンフィギュレーション モードを開始します。
ステップ 3	[no] logging description text 例 : <code>apic1(config-syslog)# logging description "This is the common logging policy"</code>	ポリシーに関する説明文を追加します。
ステップ 4	[no] logging severity {alerts critical debugging emergencies errors information notifications warnings} 例 : <code>apic1(config-syslog)# logging severity notifications</code>	syslog メッセージを送信する最小の重大度を指定します。
ステップ 5	[no] logging server-group server-group-name 例 : <code>apic1(config-syslog)# logging server-group myLoggingGroup</code>	宛先のロギング サーバ グループを指定します。
ステップ 6	[no] logging audit 例 :	ポリシーに対する監査ログを有効にします。

	コマンドまたはアクション	目的
	<code>apic1(config-syslog)# logging audit</code>	
ステップ 7	[no] logging event 例 : <code>apic1(config-syslog)# logging event</code>	ポリシーに対するイベント ログを有効にします。
ステップ 8	[no] logging fault 例 : <code>apic1(config-syslog)# logging fault</code>	ポリシーに対する障害ログを有効にします。
ステップ 9	[no] logging session 例 : <code>apic1(config-syslog)# logging session</code>	ポリシーに対するセッション ログを有効にします。

例

次に、「通知」以上の重大度のメッセージに syslog を設定する例を示します。障害およびイベントログからの syslog メッセージは、サーバグループ myLoggingGroup のサーバに送信されます。

```
apic1# configure
apic1(config)# syslog common
apic1(config-syslog)# logging description "This is the common logging policy"
apic1(config-syslog)# logging severity notifications
apic1(config-syslog)# logging server-group myLoggingGroup
apic1(config-syslog)# logging audit
apic1(config-syslog)# logging event
```

Call Home の設定

Call Home ポリシーの設定

ACI ファブリックでは、Cisco Call Home 設定を共通のモニタリングポリシーに追加できます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	callhome common 例 : <code>apic1(config)# callhome common</code>	Call Home 共通ポリシー コンフィギュレーション モードを開始します。
ステップ 3	[no] logging audit 例 : <code>apic1(config-callhome)# logging audit</code>	ポリシーに対する監査ログを有効にします。
ステップ 4	[no] logging event 例 : <code>apic1(config-callhome)# logging event</code>	ポリシーに対するイベント ログを有効にします。
ステップ 5	[no] logging fault 例 : <code>apic1(config-callhome)# logging fault</code>	ポリシーに対する障害ログを有効にします。
ステップ 6	[no] logging severity {alert critical debug emergency error info notice warning} 例 : <code>apic1(config-callhome)# logging severity notice</code>	ロギングする最小の重大度を指定します。
ステップ 7	[no] periodic-inventory notification schedule <i>scheduler</i> 例 : <code>apic1(config-callhome)# periodic-inventory notification schedule EveryEightHours</code>	定期的な通知スケジューラを設定します。スケジューラは、前もって設定する必要があります。
ステップ 8	show callhome common[destination-profile query-profile transport-email] 例 : <code>apic1(config-callhome)# show callhome common</code>	Call Home の設定を示します。

例

次に、基本的な Call Home ポリシーを設定する例を示します。

```
apic1# configure
apic1(config)# callhome common
apic1(config-callhome)# logging event
apic1(config-callhome)# logging fault
apic1(config-callhome)# logging severity notice
```

```

apic1(config-callhome)# periodic-inventory notification schedule EveryEightHours
apic1(config-callhome)# end
apic1# show callhome common
Callhome : common

Logging Enabled : event,faults
Logging Severity : notice

Destination-Profile :

Admin State      : Enabled
Contract-id      : 12345678
Customer-id      : ABCDEFG
Email-addr       : admin@example.com
From email-addr  : apic1@example.com
Reply-To email-addr : apic1@example.com
Phone Number     : +14085551212
SMTP Port num    : 25

SMTP Server : smtp.example.com

Destination      Email-addr      Format      Message-Size      Message-Level
-----
SanJose          admin@example.com  xml         40000              alert

Query-Profile :

Query Name      Query Type      Dn/Class      Target      Responses Subtree      Response Include
-----
myUserQuery     class           User          self        children
ep-records,fault-records,stats

```

次のタスク

宛先プロファイルと（オプションで）クエリ プロファイルを設定します。

Call Home 宛先プロファイルの設定

少なくとも 1 つの Call Home の宛先プロファイルを設定する必要があります。宛先プロファイルが電子メールメッセージ送信を使用している場合、シンプルメール転送プロトコル（SMTP）サーバを指定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	callhome common 例 : apic1(config)# callhome common	Call Home 共通ポリシー コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	[no] destination-profile 例 : <pre>apicl(config-callhome)# destination-profile</pre>	宛先プロファイルを設定します。
ステップ 4	[no] destination dest-name 例 : <pre>apicl(config-callhome-destnprof)# destination SanJose</pre>	送信するためのメッセージと重大度のレベルの形式を含む、Call Home メッセージが送信される宛先を設定します。 (注) 複数の宛先を設定できます。
ステップ 5	[no] email-addr 電子メール 例 : <pre>apicl(config-callhome-destnprof-destn)# email-addr alerts@example.com</pre>	Call Home メッセージを受信する電子メールアドレスを設定します。最大 255 文字の英数字を E メールアドレスフォーマットで指定できます。
ステップ 6	[no] format {aml xml short-txt} 例 : <pre>apicl(config-callhome-destnprof-destn)# format xml</pre>	次の形式で送信可能な Call Home メッセージの形式を設定します。 • aml : 適応型メッセージング言語 (AML) の XML スキーマの定義 (XSD) • xml : XML 形式は、Cisco Systems Technical Assistance Center (TAC) とのやり取りの中でも使用されます。 • short-txt : 1 または 2 行で障害を説明する、ポケットベルや印刷レポートに適したショートテキストフォーマット
ステップ 7	[no] message-level {alert critical debug emergency error info notice warning} 例 : <pre>apicl(config-callhome-destnprof-destn)# message-level alert</pre>	メッセージを送信するための最小の重大度レベルを設定します。
ステップ 8	[no] message-size size 例 : <pre>apicl(config-callhome-destnprof-destn)# message-size 40000</pre>	メッセージのサイズを設定します。範囲は 0 ~ 5000000 文字です。

	コマンドまたはアクション	目的
ステップ 9	exit 例 : <pre>apic1(config-callhome-destnprof-destn)# exit</pre>	宛先プロファイル コンフィギュレーション モードに戻ります。
ステップ 10	宛先プロファイルを設定します。 例 : <pre>apic1(config-callhome-destnprof)# (various commands)</pre>	次で説明されているのコマンドを使用します。 Call Home 宛先プロファイル 設定コマンド (523 ページ)
ステップ 11	show callhome common[destination-profile query-profile transport-email] 例 : <pre>apic1(config-callhome-destnprof)# show callhome common transport-email</pre>	Call Home の設定を示します。

例

この例では、alerts@example.com に重大度「alert」またはそれ以上の電子メール メッセージを送信するために、Call Home を設定する方法を示します。

```
apic1# configure
apic1(config)# callhome common
apic1(config-callhome)# destination-profile
apic1(config-callhome-destnprof)# destination SanJose
apic1(config-callhome-destnprof-destn)# email-addr alerts@example.com
apic1(config-callhome-destnprof-destn)# format xml
apic1(config-callhome-destnprof-destn)# message-level alert
apic1(config-callhome-destnprof-destn)# message-size 40000
apic1(config-callhome-destnprof-destn)# exit
apic1(config-callhome-destnprof)# contract-id 12345678
apic1(config-callhome-destnprof)# customer-id ABCDEFG
apic1(config-callhome-destnprof)# description "Example Corporation"
apic1(config-callhome-destnprof)# site-id XYZ123
apic1(config-callhome-destnprof)# street-address "1 Cisco Way"
apic1(config-callhome-destnprof)# phone-contact +14085551212
apic1(config-callhome-destnprof)# email-contact admin@example.com
apic1(config-callhome-destnprof)# transport email from apic1@example.com
apic1(config-callhome-destnprof)# transport email reply-to apic1@example.com
apic1(config-callhome-destnprof)# transport email mail-server smtp.example.com mgmtapg
inb port 25
apic1(config-callhome)# end
apic1# show callhome common transport-email
From email-addr : apic1@example.com
SMTP Port num   : 25

SMTP Server : smtp.example.com
```

Call Home 宛先プロファイル設定コマンド

These commands are entered in the Call Home destination profile (**config-callhome-destnprof**) configuration mode.

コマンド	Purpose
contract-id <i>contract-id</i>	お客様の Call Home 契約番号。
customer-id <i>customer-id</i>	ライセンス上のサポート契約の契約番号を含む Cisco Connection Online (CCO) ID。
description <i>text</i>	このお客様サイトに関する説明テキスト。
email-contact 電子メール	主要連絡先の電子メール アドレス。
phone-contact <i>phone-num</i>	主要連絡先の電話番号。
site-id <i>site-id</i>	お客様のサイトに固有の Call Home ID。
street-address <i>address</i>	主要連絡先の住所。
transport email from 電子メール	システムによって送信される Call Home アラート メッセージの [From] フィールドに表示される電子メール アドレス。
transport email reply-to 電子メール	システムによって送信される Call Home アラート メッセージの [From] フィールドに表示される返信電子メール アドレス。
transport email mail-server <i>smtp</i> サーバ mgmtcp { <i>inb</i> <i>oob</i> } port ポート番号	SMTP サーバの IP アドレスまたはホスト名およびシステムが SMTP サーバへの通信に使用するポート番号。

Call Home クエリの設定

イベントが、Call Home レポートの送信をトリガーすると、選択したクエリ情報がレポートに含まれます。クラス名または識別名に基づいてクエリを設定することができ、さらにサブツリーに基づいてクエリを設定できます。

始める前に

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 :	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	<code>apicl# configure</code>	
ステップ 2	callhome common 例 : <code>apicl(config)# callhome common</code>	Call Home 共通ポリシー コンフィギュレーション モードを開始します。
ステップ 3	[no] query-profile 例 : <code>apicl(config-callhome)# query-profile</code>	Call Home クエリ プロファイル コンフィギュレーション モードを開始します。
ステップ 4	[no] query query-name type {class class-name dn name} 例 : <code>apicl(config-callhome-queryprof)# query myUserQuery type class User</code>	クエリ プロファイルを設定します。
ステップ 5	[no] response-subtree {full children no} 例 : <code>apicl(config-callhome-queryprof-query)# response-subtree children</code>	応答サブツリーを設定します。完全なサブツリー、1つの子、またはサブツリー情報を含めることができます。
ステップ 6	[no] response-incl {option[,option[,option...]]} 例 : <code>apicl(config-callhome-queryprof-query)# response-incl ep-records,fault-records,stats</code>	応答に含まれる特定のサブツリー情報カテゴリを設定します。複数のカテゴリはカンマ区切りのリストで指定できます。使用できるドライブを クエリサブツリーカテゴリ (525 ページ) に示します。
ステップ 7	[no] target {children self subtree} 例 : <code>apicl(config-callhome-queryprof-query)# target self</code>	クエリのターゲットを設定します
ステップ 8	show callhome common[destination-profile query-profile transport-email] 例 : <code>apicl(config-callhome-queryprof-query)# show callhome common query-profile</code>	Call Home の設定を示します。

例

この例では、Call Home クエリを設定する方法を示します。

```
apicl# configure
```

```

apic1(config)# callhome common
apic1(config-callhome)# query-profile
apic1(config-callhome-queryprof)# query myUserQuery type class User
apic1(config-callhome-queryprof-query)# response-subtree children
apic1(config-callhome-queryprof-query)# response-incl ep-records,fault-records,stats
apic1(config-callhome-queryprof-query)# target self
apic1(config-callhome)# end
apic1# show callhome common destination-profile
Query-Profile :

Query Name      Query Type      Dn/Class      Target      Responses Subtree      Response Include
-----
myUserQuery      class           User           self        children
ep-records,fault-records,stats

```

クエリ サブツリー カテゴリ

クエリのカテゴリ	説明
add-mo-list	
audit-logs	
config-only	
count	
custom-path-hop	
deployment	
deployment-records	
ep-records	
event-logs	
fault-count	
fault-records	
障害	
full-deployment	
health	
health-records	
local-prefix	
no-scoped	
none	

port-deployment	
record-subtree	
relations	
relations-with-parent	
required	
state	
stats	
タスク	

TACACS 外部ログの設定

NX OS スタイル CLI を使用して TACACS 外部ロギング宛先グループの作成

NX OS スタイルのコマンドラインインターフェイス (CLI) を使用して、TACACS 通知先グループと宛先を設定することができます。TACACS 宛先グループを使用すると、ログデータを送信する AAA をリモートの TACACS サーバの宛先のリストを作成できます。グループには 1 つ以上の宛先を作成できます。宛先グループの作成後は、TACACS のソースと、ファブリックポリシーや、外部アクセス ポリシーで設定されている特定のテナント ポリシーのいずれかを関連付けることができます、Cisco Application Policy Infrastructure Controller (Cisco APIC)。



(注) CLI では、NX OS スタイル TACACS 外部ロギング コマンドにアクセスする権限は必須です。

次の例の CLI コマンドは、TACACS 宛先グループと NX OS スタイル CLI を使用して宛先を設定する方法を示します。

手順

ステップ 1 コンフィギュレーション モードを開始します。

例 :

```
apic1# config
```

ステップ 2 Create a TACACS destination group.

例 :

次のコマンドでは、「tacacs-宛先-grp-1」という名前の TACACS 宛先グループが作成されます。

```
apicl(config)# tacacslog-group tacacs-dest-grp-1
```

ステップ 3 新しい宛先グループで TACACS 宛先を作成します。

例：

次のコマンドでは、「1.1.1.1」の IP アドレスとリモート TACACS 宛先が作成され、デフォルトポート番号 49 が含まれています。

```
apicl(config-tacacslog-group)# remote-dest 1.1.1.1 port 49
```

(注) ログの他のポート番号を含むポート キーワードの後で同じ IP アドレスで複数のポートに送信されることができます。

ステップ 4 新しい TACACS リモート接続先の特定のパラメータを設定します。

例：

次のコマンド例では、次の特性は新しい remote destination(リモート宛先、リモート接続先)を設定します。

- 認証キー: 12345
- Authentication protocol: MS-CHAP
- 管理 EPG: アウトオブバンド

```
apicl(config-remote-dest)# key
Enter Key: 12345
Enter Key again: 12345
apicl(config-remote-dest)# protocol mschap
apicl(config-remote-dest)# management-epg oob
```

この設定の結果は、TACACS サーバのリモート接続先を含む TACACS 宛先グループを作成します。同じ AAA ロギング データが複数のリモート TACACS サーバに送信する場合は、3 と 4 に必要な数の手順を繰り返すことができます。

NX OS スタイル CLI を使用して TACACS 外部ログの送信元の作成

NX OS スタイル CLI を使用すると、TACACS の送信元を設定できます。この設定では、送信元は TACACS 宛先グループに関連付けられます。TACACS 送信元を作成すると一連の AAA ロギング データの送信先を決定します。例えば、ファブリック ポリシーで TACACS 送信元を作成する場合、Cisco Application Policy Infrastructure Controller (Cisco APIC) でサポートされている Cisco Application Centric Infrastructure (Cisco ACI) のすべての AAA ロギング データは、関連付けられている TACACS 宛先に送信されます。異なる宛先グループをサポートする 1 つ以上の送信元を作成することができます。

次の例の CLI コマンドは、NX OS スタイル CLI を使用して TACACS 送信元を設定する方法を示します。

手順

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# config
```

ステップ 2 TACACS 送信元を作成します。

例：

次のコマンドでは、「tacacs-src-1」という名前の TACACS 送信元が作成されます。

```
apic1(config)# tacacslog-monitoring common tacacslog-src tacacs-src-1
```

ステップ 3 TACACS 送信元を TACACS 宛先グループに関連付けます。

例：

次のコマンドでは、「tacacs-dest-grp-1」という名前の TACACS 宛先グループが新しい TACACS 送信元に関連付けられます。

```
apic1(config-tacacslog-monitoring)# server-group tacacs-dest-grp-1
```

この設定の結果が、ファブリック全体の TACACS 送信元の作成と、TACACS サーバのリモート宛先を含む宛先グループの関連付けです。ファブリック全体のすべての AAA ロギングデータは、関連付けられている TACACS 宛先に送信されます。

NX-OS スタイルの CLI を使用したオンデマンドの techsupport ファイルの送信



(注) APIC にエクスポートする場合や、帯域幅と計算のリソースが不足している外部サーバにエクスポートする場合は特に、同時に 5 つを超えるノードから techsupport ファイルの回収をトリガーしないでください。

APIC の過剰なストレージの使用を避けるには、ローカルに保存された techsupport ファイルをすぐに削除してください。

始める前に

techsupport ファイルをエクスポートするリモート パスを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	trigger techsupport {all controllers switch node-id} [remotename remote-path-name] 例 : <pre>apic1# trigger techsupport switch 101,103 remotename remote5</pre>	コントローラ、スイッチ、またはすべてからリモートパスへの techsupport ファイルのエクスポートをトリガーします。スイッチの場合、範囲またはカンマ区切りのリストを指定できます。リモートホストを指定しない場合、ファイルはコントローラ自体に収集されます。
ステップ 2	trigger techsupport host host-id 例 : <pre>apic1# trigger techsupport host</pre>	指定されたホストからリモートパスへの techsupport ファイルのエクスポートをトリガーします。リモートホストを指定しない場合、ファイルはコントローラ自体に収集されます。
ステップ 3	trigger techsupport local 例 : <pre>apic1# trigger techsupport local</pre>	リモートホストへのローカル techsupport ファイルのエクスポートをトリガーします。リモートホストを指定しない場合、ファイルはコントローラ自体に収集されます。
ステップ 4	show techsupport {all controllers switch node-id} status 例 : <pre>apic1# show techsupport switch 101 status</pre>	techsupport ファイルがトリガーされると、このコマンドによって techsupport レポートのステータスが表示されます。

例

次に、apic1 コントローラのローカルに保存されるようにスイッチ 101 の techsupport ファイルをトリガーする例を示します。

```
apic1# trigger techsupport switch 101
```

```
Triggering techsupport for Switch 101 using policy supNode101, setting filters to default value
```

```
Triggered on demand tech support successfully for Switch 101, will be available at: /data/techsupport on the controller. Use 'show techsupport' with your options to check techsupport status.
```

ファイルエクスポートのリモートパスの設定

ACIファブリックでは、techsupportまたはコンフィギュレーションファイルをエクスポートする1つ以上のリモート宛先を設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apicl# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] remote path remote-path-name 例： apicl(config)# remote path myFiles	リモート パスのコンフィギュレーション モードを開始します。
ステップ 3	user username 例： apicl(config-remote)# user admin5	リモート サーバにログインするユーザ名を設定します。パスワードを入力するように求められます。
ステップ 4	path {ftp scp sftp} host[:port] [remote-directory] 例： apicl(config-remote)# path sftp filehost.example.com:21 remote-directory /reports/apic	リモート サーバへのパスとプロトコルを設定します。パスワードを入力するように求められます。

例

次に、ファイルをエクスポートするためにリモートパスを設定する例を示します。

```

apicl# configure
apicl(config)# remote path myFiles
apicl(config-remote)# user admin5
You must reset the password when modifying the path:
Password:
Retype password:
apicl(config-remote)# path sftp filehost.example.com:21 remote-directory /reports/apic
You must reset the password when modifying the path:
Password:
Retype password:

```

モニタリングの Show コマンドの使用方法

show コマンドの使用について

障害、イベント、ヘルス、統計情報、および監査ログの **show** コマンドは、特定のタイプの情報または、コントローラ、リーフスイッチ、スパインスイッチ、またはテナントなどの特定のエンティティからの情報を表示するためにフィルタリングできます。

広範なクエリはシステムリソースおよびストレージの面で費用がかかります。たとえば、エンティティのフィルタを使用せずに障害、イベント、監査の **show** コマンドを使用して、システム全体からすべてのログまたはレコードを検索します。シスコは、可能な限りクエリを絞り込むために、利用可能なデータおよびエンティティのフィルタを使用することを推奨します。たとえば、次のコマンドでは、直近の 45 分間にクエリを制限することで、迅速かつ多くのフィルタリングされた応答が得られます。

```
show audits last-minutes 45
```



ヒント

コマンドの各ポイントで、「?」を入力すると、それぞれの簡単な説明とその時点で使用可能なすべてのキーワードおよびオプションが表示されます。

show faults コマンドの使用方法

The **show faults** command can combine several data filters and an entity filter to deliver a specific set of faults. コマンド構文は次のとおりです。

show faults [*filter1* [*filter2*...]] [*entity-filter*]

エンティティのフィルタは、出力をコントローラ、リーフ、スパイン、またはテナントの障害に限定します。使用できるエンティティ フィルタを [show コマンドのエンティティ フィルタ \(536 ページ\)](#) に示します。

データフィルタは、障害のクエリのタスクを簡単にするためにユーザに提供されます。使用可能なデータ フィルタは次のとおりです。

フィルタ	説明
ack {yes no}	確認応答の状態
cause <i>name</i>	cause
code <i>fault-code</i>	障害コード
controller	コントローラ情報
detail	詳細な障害情報

フィルタ	説明
end-time <i>YYYY-MM-DDTHR-MM:SS</i>	この時間までの障害アクティビティ
history	履歴情報
id <i>fault-id</i>	障害 ID
l4l7-cluster [<i>cluster name</i> <i>tenant name</i>]	L4 L7 デバイス情報
l4l7-graph [<i>cluster name</i> <i>tenant name</i>]	L4 L7 グラフ情報
last-days <i>days</i>	最後の N 日間の障害アクティビティ
last-hours <i>hours</i>	最後の N 時間の障害アクティビティ
last-minutes <i>minutes</i>	最後の N 分間の障害アクティビティ
lc <i>lc-state</i>	lifecycle state
leaf [<i>leaf-id</i>]	リーフ スイッチ情報
microsoft domain <i>name</i>	マイクロソフトのドメイン情報
min-severity <i>severity-value</i>	最小値、重大度の
severity <i>severity-value</i>	severity
spine [<i>spine-id</i>]	スパイン スイッチ情報
start-time <i>YYYY-MM-DDTHR-MM:SS</i>	この時間以降の障害アクティビティ
tenant [<i>name</i>]	テナント情報
type <i>fault-type</i>	障害の種類
vmware domain <i>name</i>	VMware のドメイン情報

例

この例では、コード「F110473」、重要度「warning」、ライフサイクル「raised」およびテナント TSW_Tenant0 の確認応答の状態「no」の過去 5 日間に発生したすべての障害を示します。

```
apic1# show faults code F110473 last-days 5 severity warning lc raised ack no tenant
TSW_Tenant0
Code          : F110473
Severity      : warning
Last Transition : 2015-11-03T01:19:04.913+00:00
Lifecycle     : raised
DN            : uni/tn-TSW_Tenant0/BD-tsw0ctx0BD1/fault-F110473
Description   : TCA: ingress drop bytes rate(l2IngrBytesAg15min:dropRate)
               value 160462 raised above threshold 100000
```

show events コマンドの使用法

The **show events** command can combine several data filters and an entity filter to deliver a specific set of events. コマンド構文は次のとおりです。

show events [*filter1* [*filter2...*]] [*entity-filter*]

エンティティのフィルタは、出力をコントローラ、リーフ、スパイン、またはテナントのイベントに限定します。使用できるエンティティフィルタを [show コマンドのエンティティフィルタ](#) (536 ページ) に示します。

データフィルタは、イベントのクエリのタスクを簡単にするためにユーザに提供されます。使用可能なデータ フィルタは次のとおりです。

フィルタ	説明
cause <i>fault-value</i>	cause
code <i>event-code</i>	イベント コード
controller	コントローラ情報
detail	詳細なイベント情報
end-time <i>YYYY-MM-DDTHR-MM:SS</i>	この時間までのイベント アクティビティ
id <i>event-id</i>	イベント ID
last-days <i>days</i>	最後の N 日間のイベント アクティビティ
last-hours <i>hours</i>	最後の N 時間のイベント アクティビティ
last-minutes <i>minutes</i>	最後の N 分間のイベント アクティビティ
leaf [<i>leaf-id</i>]	リーフ スイッチ情報
spine [<i>spine-id</i>]	スパイン スイッチ情報
start-time <i>YYYY-MM-DDTHR-MM:SS</i>	この時間以降のイベント アクティビティ
tenant [<i>name</i>]	テナント情報

例

この例では、leaf 101 上のすべてのイベントを示します。

```
apic1# show events leaf 101
```

```
Severity      : info
Affected Object : topology/pod-1/node-101/sys/phys-[eth1/28]
Code          : E4208843
ID            : 8589934758
Cause         : transition
```

```

Description      : PhysIf eth1/28 modified
Creation Time    : 2015-11-03T01:11:16.763+00:00

```

show health コマンドの使用法

The **show health** command can combine several data filters and an entity filter to deliver a specific health report. コマンド構文は次のとおりです。

show health [*filter1* [*filter2...*]] [*entity-filter*]

エンティティのフィルタは、出力をコントローラ、リーフ、スパイン、またはテナントのヘルス スコアに限定します。使用できるエンティティ フィルタを [show コマンドのエンティティ フィルタ \(536 ページ\)](#) に示します。

データ フィルタは、ヘルスのクエリのタスクを簡単にするためにユーザに提供されます。使用可能なデータ フィルタは次のとおりです。

フィルタ	説明
end-time YYYY-MM-DDTHR-MM:SS	この時間までのヘルス アクティビティ
history	履歴情報
max-change percentage	ヘルス スコア率の最小変化
min-hs 得点	最大ヘルス スコア
start-time YYYY-MM-DDTHR-MM:SS	この時間以降のヘルス アクティビティ

例

この例では、すべてのテナントの簡単なヘルス レポートを示します。

```

apic1# show health tenant
Tenant          Score    Change(%)    Created
-----
infra           100        0          2015-05-12 18:45:47PDT
common          100        0          2015-05-12 18:45:47PDT
TSW_Tenant0     98         0          2015-05-12 18:20:58PDT
mgmt            100        0          2015-05-12 18:45:47PDT

```

この例では、テナント TSW_Tenant0 の最小変化 10% で、最大 75 のヘルス スコアのすべての履歴ヘルスレコードを 11 月 4 日から示します。

```

apic1# show health max-hs 75 min-change 10 start-time 2015-11-04T01:55:48 history tenant
TSW_Tenant0

```

show audits コマンドの使用方法

The **show audits** command can be used to view the audit-logs as well as the session logs for an entity. コマンドは、監査ログの指定のセットを提供するために複数のデータフィルタとエンティティのフィルタを組み合わせることができます。コマンド構文は次のとおりです。

show audits [*filter1* [*filter2...*]] [*entity-filter*]

エンティティのフィルタは、出力をコントローラ、リーフ、スパイン、またはテナントのログに限定します。使用できるエンティティ フィルタを [show コマンドのエンティティ フィルタ \(536 ページ\)](#) に示します。

データフィルタは、監査ログのクエリのタスクを簡単にするためにユーザに提供されます。使用可能なデータ フィルタは次のとおりです。

フィルタ	説明
action { creation deletion failure modification special state-transition }	オブジェクトのアクション インジケータ
controller	コントローラ情報
detail	詳細なログ情報
end-time <i>YYYY-MM-DDTHR-MM:SS</i>	この時間までのログ アクティビティ
id <i>log-id</i>	ログ ID
last-days <i>days</i>	最後の N 日間のログ アクティビティ
last-hours <i>hours</i>	最後の N 時間のログ アクティビティ
last-minutes <i>minutes</i>	最後の N 分間のログ アクティビティ
leaf [<i>leaf-id</i>]	リーフ スイッチ情報
spine [<i>spine-id</i>]	スパイン スイッチ情報
start-time <i>YYYY-MM-DDTHR-MM:SS</i>	この時間以降のログ アクティビティ
tenant [<i>name</i>]	テナント情報
user <i>user-name</i>	ユーザの名前

例

この例では、テナント TSW_Tenant0 の最後の 45 分間のすべての監査ログに示します。

```
apic1# show audits last-minutes 45 tenant TSW_Tenant0
Creation Time   : 2015-11-03T01:11:05.708+00:00
ID              : 12884902085
User            : admin
Action          : creation
```

```
Affected Object : uni/tn-TSW_Tenant0/out-T0-sub-L3OUT-1/instP-
                  13extInstP-1/extsubnet-[192.5.1.0/24]
Description      : Subnet 192.5.1.0/24 created
```

show stats コマンドの使用法

The **show stats** command can combine data filters and an entity filter to deliver a specific set of statistics. コマンド構文は次のとおりです。

show stats granularity *granularity* [cumulative] [history] [entity-filter]

エンティティのフィルタは、出力をリーフ、スパイン、またはテナントの統計情報に限定します。使用できるエンティティ フィルタを [show コマンドのエンティティ フィルタ \(536 ページ\)](#) に示します。

データフィルタは、統計情報のクエリのタスクを簡単にするためにユーザに提供されます。使用可能なデータ フィルタは次のとおりです。

フィルタ	説明
cumulative	累積統計情報
granularity {5min 15min 1h 1d 1w 1mo 1qtr 1year}	5 分、15 分、1 時間、1 日、1 週間、1 ヶ月、1 四半期、または 1 年のサンプリング間隔サイズ
history	履歴統計情報

例

この例では、テナント TSW_Tenant0 の 15 分単位の粒状統計を示します。

```
apic1# show stats granularity 15min tenant TSW_Tenant0
```

この例では、特定のポートの 15 分単位の粒状統計を示します。

```
apic1# show stats granularity 15min leaf 101 interface ethernet 1/1
```

show コマンドのエンティティ フィルタ

エンティティフィルタは、コントローラ、リーフ、スパイン、またはテナントの障害に出力を制限する多くの show コマンドを拡張できます。使用可能なエンティティ フィルタは次のとおりです。

フィルタ
controller

フィルタ
leaf <i>node-id</i> [fex]
leaf ノード <i>id</i> interface [ethernet <i>slot/port</i> l3instance [インスタンス名] mgmt [mgmt0] portchannel tunnel [トンネル名]]
leaf ノード <i>id</i> inventory { chassis [number] fans [番号] module [番号] powersupply [番号] supervisor [number] }
leaf <i>node-id</i> protocol { arp bgp coop ipv4 ipv6 isis lldp ospf ospfv3 }
leaf <i>node-id</i> vpc {
leaf <i>node-id</i> vrf [vrf-name]
spine <i>node-id</i>
spine ノード <i>id</i> interface [ethernet <i>slot/port</i> l3instance [インスタンス名] mgmt [mgmt0] tunnel [トンネル名]]
spine ノード <i>id</i> inventory { chassis [number] fabric [番号] fans [番号] module [番号] powersupply [番号] supervisor [番号] system [number] }
spine <i>node-id</i> protocol { arp bgp coop ipv4 ipv6 isis lldp ospf ospfv3 }
spine <i>node-id</i> vrf [vrf-name]
tenant <i>tenant-name</i>
tenant <i>tenant-name</i> application [app-name] [epg]
tenant <i>tenant-name</i> bridge-domain [bd-name]
tenant <i>tenant-name</i> interface bridge-domain [bd-name]

SNMP の設定

始める前に

SNMP 通信を可能にするには、SNMP トラフィック（通常、UDP 161 上）を許可するアウトオブバンド コントラクトを設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 :	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	<code>apic1# configure</code>	
ステップ 2	template snmp-fabric <i>snmp</i> ファブリック テンプレート名 例 : <code>apic1(config)# template snmp-fabric Poll</code>	テンプレート snmp-fabric ファブリックモードを開始します
ステップ 3	[no] snmp-server protocol enable 例 : <code>apic1(config-template-snmp-fabric)# snmp-server protocol enable</code>	SNMP プロトコルのサポートをイネーブル（またはディセーブル）にします。
ステップ 4	[no] snmp-server community <i>community-name</i> 例 : <code>apic1(config-template-snmp-fabric)# snmp-server community mysecret</code>	コミュニティが必要なのは SNMPv2 だけです。
ステップ 5	snmp-server contact <i>contact-name</i> 例 : <code>apic1(config-template-snmp-fabric)# snmp-server contact admin80</code>	。
ステップ 6	snmp-server location <i>location-name</i> 例 : <code>apic1(config-template-snmp-fabric)# snmp-server location SanJose</code>	SNMP サーバの場所を設定します。
ステップ 7	exit 例 : <code>apic1(config-template-snmp-fabric)# exit</code>	グローバル コンフィギュレーションモードに戻ります。
ステップ 8	template pod-group <i>pod-group-template-name</i> 例 : <code>apic1(config)# template pod-group allPods</code>	ポッドグループ テンプレート（ポリシー）を設定します。
ステップ 9	inherit snmp-fabric <i>snmp-fabric-template-name</i> 例 : <code>apic1(config-pod-group)# inherit snmp-fabric Poll</code>	事前に設定したポッドグループにポッドプロファイルを関連付けます。

例

次の例では、ファブリックでSNMPトラフィックを許可する、アウトオブバンド契約を設定します。

```
apicl# configure
apicl(config)# template snmp-fabric Poll
apicl(config-template-snmp-fabric)# snmp-server protocol enable
apicl(config-template-snmp-fabric)# snmp-server community mysecret
apicl(config-template-snmp-fabric)# snmp-server contact admin80
apicl(config-template-snmp-fabric)# snmp-server location SanJose
apicl(config-template-snmp-fabric)# exit
apicl(config)# template pod-group allPods
apicl(config-pod-group)# inherit snmp-fabric Poll
apicl(config-pod-group)# exit
apicl(config)#
```

CLI を使用した SNMP ポリシーの設定

SNMP ポリシーを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	コンフィギュレーションモードに入ります。
ステップ 2	template snmp-fabric default 例 : apicl(config)# template snmp-fabric default	SNMP ポリシーを作成します。
ステップ 3	snmp-server clientgroup	SNMP クライアント グループを設定します。クライアントのグループには、ルータやスイッチに SNMP アクセスを許可するクライアントの IP アドレスのグループです。
ステップ 4	snmp-server community 例 : apicl(config-template-snmp-fabric)# snmp-server community abc	SNMP コミュニティを設定します。SNMP コミュニティ プロファイルには、モニタリングのルータまたはスイッチの統計情報へのアクセスが有効になります。

	コマンドまたはアクション	目的
ステップ 5	snmp-server contact	SNMP 連絡先情報を設定します。
ステップ 6	snmp-server host 例 : <pre>apic1(config-template-snmp-fabric)# snmp-server host 2001:420:28e:2020::10 traps-version 2c abc apic1(config-template-snmp-fabric)# snmp-server host 2001:420:28e:2020::2 traps-version 2c abc apic1(config-template-snmp-fabric)# snmp-server host 2001:420:28e:2020::11 traps-version 2c abc</pre>	SNMP トラップ ホストを設定します。
ステップ 7	snmp-server location	SNMP ロケーションを設定します。
ステップ 8	snmp-server protocol 例 : <pre>apic1(config-template-snmp-fabric)# snmp-server protocol enable</pre>	SNMP プロトコルを設定します。
ステップ 9	snmp-server trap-fwd-server 例 : <pre>apic1(config-template-snmp-fabric)# snmp-server trap-fwd-server 172.31.128.199</pre>	サーバの SNMP トラップ転送を設定します。
ステップ 10	snmp-server user 例 : <pre>apic1(config-template-snmp-fabric)# snmp-server user test_user auth hmac-md5-96 '' priv none privacy-passphrase ''</pre>	SNMP ユーザを設定します。SNMP ユーザプロファイルは、ユーザに関連付ける SNMP ポリシー、ネットワーク内のデバイスを監視するために使用されます。
ステップ 11	show running-config	設定を確認します。

Smart Callhome の設定

スマート Callhome について

スマート Callhome Callhome として同様の方法で重要システム ポリシーの電子メールベースの通知を提供します。ただし、スマート Callhome email messages(電子メール メッセージ、E メール メッセージ)に提供する障害の詳細な選択を収集します。



(注) スマート Callhome のみ を収集し、障害を提供します。

スマート Callhome 機能の一般的な障害のトリガーでは、ネットワークの中断を脅かすイベントの種類に対応します。次に例を示します。

- 温度障害: センサーの温度がしきい値を超えています。
- ファン/電源装置障害: ファンまたは電源装置のがオフラインになります。
- ディスク使用率の障害: デバイスのディスク使用率がしきい値を超えています。

スマート Callhome では、障害を収集し、ネットワーク サポート エンジニア、ネットワーク オペレーション センターでは、または Cisco Smart Callhome サービス テクニカル アシスタンス センター (TAC) とのケースを生成するには、それらを電子メールします。

NX OS スタイル CLI を使用して Smart Callhome 宛先グループを作成する

Smart Callhome は障害を収集し、ネットワーク サポート エンジニア、ネットワーク オペレーション センター、または Cisco Smart Callhome サービスにメールで送信して、テクニカル アシスタンス センター (TAC) とのケースを生成します。

NX OS スタイル CLI を使用すると、Smart Callhome 宛先グループと宛先を設定します。Smart Callhome 宛先グループでは、障害のデータが送信される電子メール宛先のリストを作成できます。グループには 1 つ以上の宛先を作成できます。宛先グループの作成後、Cisco Application Policy Infrastructure Controller (Cisco APIC) にサポートされているスイッチ ファブリック全体か、特定のテナントに対して、Smart Callhome と関連付けることができます。

次の例の CLI コマンドは、NX OS スタイル CLI を使用して Smart Callhome 宛先グループと宛先を設定する方法を示します。

手順

ステップ 1 コンフィギュレーション モードを開始します。

例 :

```
apic1# config
```

ステップ 2 Smart Callhome 共通ポリシー設定モードを入力します。

例 :

```
apic1(config)# smartcallhome common
```

(注) 共通ポリシー設定モードのデフォルト名は、「一般」です。作成できる名前はこれだけです。

ステップ 3 Smart Callhome 宛先グループを作成します。

例：

次のコマンドでは、Smart Callhome 宛先グループが作成されます。

```
apic1(config-smartcallhome)# destination-profile
```

ステップ 4 新しい宛先グループで SMTP サーバを設定します。

例：

次のコマンドでは、「10.10.10.2」の、IP アドレスを持つ SMTP サーバが宛先グループに追加されます。

```
apic1(config-callhome-destnprof)# transport email mail-server 10.10.10.2
```

ステップ 5 新しい Smart Callhome 宛先グループに関するプロファイルパラメータを設定します。

例：

次のコマンドは、宛先グループに関する追加情報を提供します。

- **Contract-id:** 顧客のサービス契約 ID。
- **Customer-id:** 顧客 ID。
- **説明** : Smart Callhome 宛先プロファイルの説明。
- **Email-contact:** 顧客の連絡先のメールアドレス。
- **Phone-contact:** 顧客の連絡先電話番号。
- **site-id:** ネットワークが展開されているサイトの ID。
- **Street-address:** サイトのストリートアドレス。

ステップ 6 新しい宛先グループで Smart Callhome 宛先を作成します。

例：

次のコマンドでは、「sch-dest-1」という名前のリモート Smart Callhome 宛先が作成されます。

```
apic1(config-callhome-destnprof)# destination sch-dest-1
```

ステップ 7 新しいリモート Smart Callhome 宛先の特定のパラメータを設定します。

例：

次のコマンド例では、次の特性は新しいリモート宛先に設定されます。

- 電子メールアドレス: net-admin@site.com
- メッセージ形式: 短いテキスト
- RFC 準拠: True

```
apicl(config-callhome-destnprof-destn)# email-addr net-admin@site.com
apicl(config-callhome-destnprof-destn)# format short-txt
apicl(config-callhome-destnprof-destn)# rfc-compliant true
```

この設定の結果では、電子メールのリモート宛先を含む Smart Callhome 宛先グループを作成です。同じ Smart Callhome 障害データを複数の電子メール宛先に送信する場合は、手順 5 と 6 を必要な回数繰り返します。



第 20 章

SPAN の設定

- [SPAN および ERSPAN の設定 \(545 ページ\)](#)

SPAN および ERSPAN の設定

ACI ファブリックでは、SPAN 機能を以下の 3 つのカテゴリで設定できます。

- アクセス：リーフ ノードのアクセス ポートから発信されたトラフィックを監視します
- ファブリック：リーフまたはスパイン ノードのファブリック ポートからのトラフィックを監視します
- テナント：テナント内のエンドポイントグループ (EPG) からのトラフィックを監視します

次の表に、各セッションごとに異なる構成要素を示します。

Session Type	Sources	Filters	Destination
アクセス ローカル	1 つのリーフ へのローカル アクセス ポート、ポート チャネル	EPG	送信元と同じリーフ へのローカル ポート
アクセス ERSPAN	1 つ以上のリーフ ノード間のアクセス ポート、ポート チャネル、VPC	EPG	ファブリック内のいずれかの EPG
ファブリック ERSPAN	1 つ以上のリーフ またはスパイン ノード内のファブリック ポート	BD または VRF	ファブリック内のいずれかの EPG
テナント ERSPAN	ファブリック内のいずれかの EPG	-	ファブリック内のいずれかの EPG

SPAN の注意事項と制約事項

- SPAN はトラブルシューティングのためにのみ使用してください。SPAN トラフィックはスイッチ リソースのユーザ トラフィックと競合します。負荷を最小限にするには、分析対象の特定のトラフィックだけをコピーするように SPAN を設定します。
- SPAN 送信元として `l3extLifP` のレイヤ 3 サブインターフェイスを指定することはできません。外部ソースからのトラフィックをモニタリングするためにはポート全体を使用します。
- FEX インターフェイス用のローカル SPAN では、FEX インターフェイスは、SPAN 送信元としてのみ使用できます。SPAN の宛先としては使用できません。第 1 世代のスイッチ (スイッチ名に EX または FX の付かない Cisco Nexus 9000 シリーズのスイッチ) では、Tx SPAN はどのレイヤ 3 のスイッチ トラフィックでも機能しません。第 2 世代のスイッチ (スイッチ名に EX または FX の付くもの) では、トラフィックがレイヤ 2 またはレイヤ 3 でスイッチされたものである場合、Tx SPAN は機能しません。Rx SPAN に制限はありません。
- テナントおよびアクセス SPAN は カプセル化リモート拡張 SPAN (ERSPAN) タイプ I を使用し、ファブリック SPAN は ERSPAN タイプ II を使用します。ERSPAN ヘッダーについては、IETF の Internet Draft (<https://tools.ietf.org/html/draft-foschiano-erspan-00>) を参照してください。
- ファブリックでは ERSPAN の宛先 IP をエンドポイントとして学習する必要があります。
- SPAN は IPv6 トラフィックをサポートしますが、ERSPAN の宛先 IP を IPv6 アドレスにすることはできません。
- アクティブな SPAN セッションの最大数など、SPAN 関連の制限については、『*Verified Scalability Guide for Cisco ACI*』ドキュメントを参照してください。

アクセス モードの SPAN の設定

これは、アクセス リーフ ノードにローカルに適用される従来の SPAN 設定です。1 つ以上のアクセス ポートまたはポートチャネルから送信されるトラフィックは監視でき、同じリーフ ノードにローカルな宛先ポートに送信できます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>apic1# configure</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] monitor access session session-name 例 : <pre>apic1(config)# monitor access session mySession</pre>	アクセス モニタリングセッションの設定を作成します。

	コマンドまたはアクション	目的
ステップ 3	[no] description text 例 : <pre>apicl(config-monitor-access)# description "This is my SPAN session"</pre>	このアクセス モニタリングセッションについての説明を追加します。テキストにスペースが含まれている場合は、単一引用符で囲む必要があります。
ステップ 4	[no] destination interface ethernet slot/port leaf node-id 例 : <pre>apicl(config-monitor-access)# destination interface eth 1/2 leaf 101</pre>	宛先インターフェイスを指定します。宛先インターフェイスを FEX ポートまたはポートチャネルにすることはできません。
ステップ 5	[no] source interface ethernet {[fex/]slot/port port-range} leaf node-id 例 : <pre>apicl(config-monitor-access)# source interface eth 1/2 leaf 101</pre>	送信元インターフェイスポートまたはポート範囲を指定します。
ステップ 6	[no] direction {rx tx both} 例 : <pre>apicl(config-monitor-access-source)# direction tx</pre>	監視対象のトラフィックの方向を指定します。方向は各送信元ポート範囲で個別に設定できます。
ステップ 7	[no] filter tenant tenant-name application application-name epg epg-name 例 : <pre>apicl(config-monitor-access-source)# filter tenant t1 application appl epg epg1</pre>	監視対象のトラフィックをフィルタします。フィルタは、各送信元ポート範囲で個別に設定できます。
ステップ 8	exit 例 : <pre>apicl(config-monitor-access-source)# exit</pre>	アクセス モニタセッション コンフィギュレーション モードに戻ります。
ステップ 9	[no] source interface port-channel port-channel-name-list leaf node-id [fex fex-id] 例 : <pre>apicl(config-monitor-access)# source interface port-channel pc5 leaf 101</pre>	送信元インターフェイスのポートチャネルを指定します (ここに示されていないトラフィック方向とフィルタ設定を入力します)。
ステップ 10	[no] shutdown 例 : <pre>apicl(config-monitor-access)# no shut</pre>	モニタリングセッションをディセーブル (またはイネーブル) にします。

例

次に、ローカル アクセス モニタリング セッションを設定する例を示します。

```
apic1# configure terminal
apic1(config)# monitor access session mySession
apic1(config-monitor-access)# description "This is my SPAN session"
apic1(config-monitor-access)# destination interface eth 1/2 leaf 101
apic1(config-monitor-access)# source interface eth 1/1 leaf 101
apic1(config-monitor-access-source)# direction tx
apic1(config-monitor-access-source)# filter tenant t1 application appl epg epg1
apic1(config-monitor-access-source)# exit
apic1(config-monitor-access)# no shut
apic1(config-monitor-access)# show run
# Command: show running-config monitor access session mySession
# Time: Fri Nov 6 23:55:35 2015
monitor access session mySession
  description "This is my SPAN session"
  destination interface eth 1/2 leaf 101
  source interface eth 1/1 leaf 101
  direction tx
  filter tenant t1 application appl epg epg
  exit
exit
```

アクセス モードの ERSPAN の設定

ACI ファブリックでは、アクセス モード ERSPAN の設定は、1 つ以上のリーフ ノードのアクセス ポート、ポート チャネル、および vPC から発信されたトラフィックの監視に使用できます。

ERSPAN セッションの場合、宛先は常にファブリック内のどこにでも展開できるエンドポイント グループ (EPG) です。監視対象のトラフィックは、EPG が移動するたびに宛先に転送されます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] monitor access session session-name 例 : apic1(config)# monitor access session mySession	アクセス モニタリング セッションの設定を作成します。
ステップ 3	[no] description text 例 :	このモニタリングセッションの説明を追加します。テキストにスペースが含まれます。

	コマンドまたはアクション	目的
	<pre>apicl(config-monitor-access)# description "This is my access ERSPAN session"</pre>	まれている場合は、単一引用符で囲む必要があります。
ステップ 4	[no] destination tenant <i>tenant-name</i> application <i>application-name</i> epg <i>epg-name</i> destination-ip <i>dest-ip-address</i> source-ip-prefix <i>src-ip-address</i> 例 : <pre>apicl(config-monitor-access)# destination tenant t1 application appl epg epg1 destination-ip 192.0.20.123 source-ip-prefix 10.0.20.1</pre>	宛先インターフェイスをテナントとして指定し、宛先コンフィギュレーションモードを開始します。
ステップ 5	[no] erspan-id <i>flow-id</i> 例 : <pre>apicl(config-monitor-access-dest)# erspan-id 100</pre>	ERSPAN セッションの ERSPAN ID を設定します。ERSPAN の範囲は 1 ～ 1023 です。
ステップ 6	[no] ip dscp <i>dscp-code</i> 例 : <pre>apicl(config-monitor-access-dest)# ip dscp 42</pre>	ERSPAN トラフィックのパケットの DiffServ コードポイント (DSCP) 値を設定します。指定できる範囲は 0 ～ 64 です。
ステップ 7	[no] ip ttl <i>ttl-value</i> 例 : <pre>apicl(config-monitor-access-dest)# ip ttl 16</pre>	ERSPAN トラフィックの IP 存続可能時間 (TTL) 値を設定します。範囲は 1 ～ 255 です。
ステップ 8	[no] mtu <i>mtu-value</i> 例 : <pre>apicl(config-monitor-access-dest)# mtu 9216</pre>	ERSPAN セッションの最大伝送単位 (MTU) サイズを設定します。有効な範囲は 64 ～ 9216 バイトです。
ステップ 9	exit 例 : <pre>apicl(config-monitor-access-dest)#</pre>	モニタアクセスコンフィギュレーションモードに戻ります。
ステップ 10	[no] source interface ethernet <i>{[fex/]slot/port port-range}</i> leaf <i>node-id</i> 例 : <pre>apicl(config-monitor-access)# source interface eth 1/2 leaf 101</pre>	送信元インターフェイスポートまたはポート範囲を指定します。
ステップ 11	[no] source interface port-channel <i>port-channel-name-list</i> leaf <i>node-id</i> [fex <i>fex-id</i>]	送信元インターフェイスポートチャネルを指定します。

	コマンドまたはアクション	目的
	例 : <code>apic1(config-monitor-access)# source interface port-channel pc1 leaf 101</code>	
ステップ 12	[no] source interface vpc vpc-name-list leaf node-id1 node-id2 [fex fex-id1 fex-id2] 例 : <code>apic1(config-monitor-access)# source interface vpc pc1 leaf 101 102</code>	送信元インターフェイス vPC を指定します。
ステップ 13	[no] direction {rx tx both} 例 : <code>apic1(config-monitor-access-source)# direction tx</code>	監視対象のトラフィックの方向を指定します。方向は各発信元ポート範囲で個別に設定できます。
ステップ 14	[no] filter tenant tenant-name application application-name epg epg-name 例 : <code>apic1(config-monitor-access-source)# filter tenant t1 application appl epg epg1</code>	監視対象のトラフィックをフィルタします。フィルタは、各送信元ポート範囲で個別に設定できます。
ステップ 15	exit 例 : <code>apic1(config-monitor-access-source)# exit</code>	アクセス モニタ セッション コンフィギュレーション モードに戻ります。
ステップ 16	[no] shutdown 例 : <code>apic1(config-monitor-access)# no shut</code>	モニタリングセッションをディセーブル（またはイネーブル）にします。

例

この例では、ERSPAN アクセス モニタリング セッションを設定する方法を示します。

```

apic1# configure terminal
apic1(config)# monitor access session mySession
apic1(config-monitor-access)# description "This is my access ERSPAN session"
apic1(config-monitor-access)# destination tenant t1 application appl epg epg1
apic1(config-monitor-access)# destination-ip 192.0.20.123 source-ip-prefix 10.0.20.1
apic1(config-monitor-access-dest)# erspan-id 100
apic1(config-monitor-access-dest)# ip dscp 42
apic1(config-monitor-access-dest)# ip ttl 16
apic1(config-monitor-access-dest)# mtu 9216
apic1(config-monitor-access-dest)# exit
apic1(config-monitor-access)# source interface eth 1/1 leaf 101
apic1(config-monitor-access-source)# direction tx
apic1(config-monitor-access-source)# filter tenant t1 application appl epg epg1

```

```

apic1(config-monitor-access-source)# exit
apic1(config-monitor-access)# no shut
apic1(config-monitor-access)# show run
# Command: show running-config monitor access session mySession
# Time: Fri Nov 6 23:55:35 2015
monitor access session mySession
  description "This is my ERSPAN session"
  source interface eth 1/1 leaf 101
  direction tx
  filter tenant t1 application appl1 epg epg1
  exit
  destination tenant t1 application appl1 epg epg1 destination-ip 192.0.20.123
source-ip-prefix 10.0.20.1
  ip dscp 42
  ip ttl 16
  erspan-id 9216
  mtu 9216
  exit
exit

```

この例では、監視元としてポート チャネルを設定する方法を示します。

```

apic1(config-monitor-access)# source interface port-channel pc3 leaf 105

```

この例では、監視元として vPC レッグの 1 つを設定する方法を示します。

```

apic1(config-monitor-access)# source interface port-channel vpc3 leaf 105

```

この例では、監視元として FEX 101 からポートの範囲を設定する方法を示します。

```

apic1(config-monitor-access)# source interface eth 101/1/1-2 leaf 105

```

ファブリック モードの ERSPAN の設定

ACI ファブリックのファブリック モード ERSPAN 設定を使用すると、リーフ/スパイン ノードの 1 つ以上のファブリック ポートから発信されるトラフィックをモニタリングできます。ローカル SPAN はファブリック モードではサポートされていません。

ERSPAN セッションの場合、宛先は常にファブリック内のどこにでも展開できるエンドポイント グループ (EPG) です。監視対象のトラフィックは、EPG が移動するたびに宛先に転送されます。ファブリック モードでは、送信元としてファブリック ポートのみが許可されますが、リーフ スイッチとスパイン スイッチは両方とも許可されます。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	[no] monitor fabric session <i>session-name</i> 例 : <pre>apic1(config)# monitor fabric session mySession</pre>	ファブリック モニタリングセッション設定を作成します。
ステップ 3	[no] description <i>text</i> 例 : <pre>apic1(config-monitor-fabric)# description "This is my fabric ERSpan session"</pre>	このモニタリングセッションの説明を追加します。テキストにスペースが含まれている場合は、単一引用符で囲む必要があります。
ステップ 4	[no] destination tenant <i>tenant-name</i> application <i>application-name</i> epg <i>epg-name</i> destination-ip <i>dest-ip-address</i> source-ip-prefix <i>src-ip-address</i> 例 : <pre>apic1(config-monitor-fabric)# destination tenant t1 application app1 epg epg1 destination-ip 192.0.20.123 source-ip-prefix 10.0.20.1</pre>	宛先インターフェイスをテナントとして指定し、宛先コンフィギュレーションモードを開始します。
ステップ 5	[no] erspan-id <i>flow-id</i> 例 : <pre>apic1(config-monitor-fabric-dest)# erspan-id 100</pre>	ERSpan セッションの ERSpan ID を設定します。ERSpan の範囲は 1 ～ 1023 です。
ステップ 6	[no] ip dscp <i>dscp-code</i> 例 : <pre>apic1(config-monitor-fabric-dest)# ip dscp 42</pre>	ERSpan トラフィックのパケットの DiffServ コードポイント (DSCP) 値を設定します。指定できる範囲は 0 ～ 64 です。
ステップ 7	[no] ip ttl <i>ttl-value</i> 例 : <pre>apic1(config-monitor-fabric-dest)# ip ttl 16</pre>	ERSpan トラフィックの IP 存続可能時間 (TTL) 値を設定します。範囲は 1 ～ 255 です。
ステップ 8	[no] mtu <i>mtu-value</i> 例 : <pre>apic1(config-monitor-fabric-dest)# mtu 9216</pre>	ERSpan セッションの最大伝送単位 (MTU) サイズを設定します。有効な範囲は 64 ～ 9216 バイトです。
ステップ 9	exit 例 : <pre>apic1(config-monitor-fabric-dest)#</pre>	モニタアクセスコンフィギュレーションモードに戻ります。

	コマンドまたはアクション	目的
ステップ 10	[no] source interface ethernet {slot/port port-range} switch node-id 例 : <pre>apicl(config-monitor-fabric)# source interface eth 1/2 switch 101</pre>	送信元インターフェイスポートまたはポート範囲を指定します。
ステップ 11	[no] direction {rx tx both} 例 : <pre>apicl(config-monitor-fabric-source)# direction tx</pre>	監視対象のトラフィックの方向を指定します。方向は各発信元ポート範囲で個別に設定できます。
ステップ 12	[no] filter tenant tenant-name bd bd-name 例 : <pre>apicl(config-monitor-fabric-source)# filter tenant t1 bd bd1</pre>	ブリッジドメインによりトラフィックをフィルタ処理します。
ステップ 13	[no] filter tenant tenant-name vrf vrf-name 例 : <pre>apicl(config-monitor-fabric-source)# filter tenant t1 vrf vrf1</pre>	VRF によりトラフィックをフィルタ処理します。
ステップ 14	exit 例 : <pre>apicl(config-monitor-fabric-source)# exit</pre>	アクセス モニタ セッション コンフィギュレーション モードに戻ります。
ステップ 15	[no] shutdown 例 : <pre>apicl(config-monitor-fabric)# no shut</pre>	モニタリングセッションをディセーブル（またはイネーブル）にします。

例

次に、ERSPAN ファブリック モニタリング セッションを設定する例を示します。

```
apicl# configure terminal
apicl(config)# monitor fabric session mySession
apicl(config-monitor-fabric)# description "This is my fabric ERSPAN session"
apicl(config-monitor-fabric)# destination tenant t1 application appl epg epg1
apicl(config-monitor-fabric)# destination-ip 192.0.20.123 source-ip-prefix 10.0.20.1
apicl(config-monitor-fabric-dest)# erspan-id 100
apicl(config-monitor-fabric-dest)# ip dscp 42
apicl(config-monitor-fabric-dest)# ip ttl 16
apicl(config-monitor-fabric-dest)# mtu 9216
apicl(config-monitor-fabric-dest)# exit
apicl(config-monitor-fabric)# source interface eth 1/1 switch 101
apicl(config-monitor-fabric-source)# direction tx
apicl(config-monitor-fabric-source)# filter tenant t1 bd bd1
```

```
apic1(config-monitor-fabric-source)# filter tenant t1 vrf vrf1
apic1(config-monitor-fabric-source)# exit
apic1(config-monitor-fabric)# no shut
```

テナント モードの ERSPAN の設定

ACI ファブリックのテナント モード ERSPAN 設定を使用すると、テナント内のエンドポイント グループから発信されるトラフィックをモニタリングできます。

テナント モードでは、送信元 EPG から発信されるトラフィックは同じテナント内の宛先 EPG に送信されます。トラフィックのモニタリングは、送信元または宛先 EPG がファブリック内を移動した場合に影響を受けません。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] monitor tenant tenant-name session session-name 例 : apic1(config)# monitor tenant session mySession	テナントモニタリングセッション設定を作成します。
ステップ 3	[no] description text 例 : apic1(config-monitor-tenant)# description "This is my tenant ERSPAN session"	このアクセスモニタリングセッションについての説明を追加します。テキストにスペースが含まれている場合は、単一引用符で囲む必要があります。
ステップ 4	[no] destination tenant tenant-name application application-name epg epg-name destination-ip dest-ip-address source-ip-prefix src-ip-address 例 : apic1(config-monitor-tenant)# destination tenant t1 application appl1 epg epg1 destination-ip 192.0.20.123 source-ip-prefix 10.0.20.1	宛先インターフェイスをテナントとして指定し、宛先コンフィギュレーション モードを開始します。
ステップ 5	[no] erspan-id flow-id 例 : apic1(config-monitor-tenant-dest)# erspan-id 100	ERSPAN セッションの ERSPAN ID を設定します。ERSPAN の範囲は 1 ～ 1023 です。

	コマンドまたはアクション	目的
ステップ 6	[no] ip dscp dscp-code 例 : <pre>apicl(config-monitor-tenant-dest)# ip dscp 42</pre>	ERSPAN トラフィックのパケットの DiffServ コードポイント (DSCP) 値を設定します。指定できる範囲は 0 ~ 64 です。
ステップ 7	[no] ip ttl ttl-value 例 : <pre>apicl(config-monitor-tenant-dest)# ip ttl 16</pre>	ERSPAN トラフィックの IP 存続可能時間 (TTL) 値を設定します。範囲は 1 ~ 255 です。
ステップ 8	[no] mtu mtu-value 例 : <pre>apicl(config-monitor-tenant-dest)# mtu 9216</pre>	ERSPAN セッションの最大伝送単位 (MTU) サイズを設定します。有効な範囲は 64 ~ 9216 バイトです。
ステップ 9	exit 例 : <pre>apicl(config-monitor-tenant-dest)#</pre>	モニタアクセスコンフィギュレーションモードに戻ります。
ステップ 10	[no] source application application-name epg epg-name 例 : <pre>apicl(config-monitor-tenant)# source application app2 epg epg5</pre>	送信元インターフェイスポートまたはポート範囲を指定します。
ステップ 11	[no] direction {rx tx both} 例 : <pre>apicl(config-monitor-tenant-source)# direction tx</pre>	監視対象のトラフィックの方向を指定します。方向は各発信元ポート範囲で個別に設定できます。
ステップ 12	exit 例 : <pre>apicl(config-monitor-tenant-source)# exit</pre>	アクセス モニタ セッション コンフィギュレーションモードに戻ります。
ステップ 13	[no] shutdown 例 : <pre>apicl(config-monitor-tenant)# no shut</pre>	モニタリングセッションをディセーブル (またはイネーブル) にします。

例

次に、ERSPAN テナント モニタリング セッションを設定する例を示します。

```
apicl# configure terminal
```

```
apic1(config)# monitor access session mySession
apic1(config-monitor-tenant)# description "This is my tenant ERSPAN session"
apic1(config-monitor-tenant)# destination tenant t1 application appl1 epq epq1
destination-ip 192.0.20.123 source-ip-prefix 10.0.20.1
apic1(config-monitor-tenant-dest)# erspan-id 100
apic1(config-monitor-tenant-dest)# ip dscp 42
apic1(config-monitor-tenant-dest)# ip ttl 16
apic1(config-monitor-tenant-dest)# mtu 9216
apic1(config-monitor-tenant-dest)# exit
apic1(config-monitor-tenant)# source application app2 epq epq5
apic1(config-monitor-tenant-source)# direction tx
apic1(config-monitor-tenant-source)# exit
apic1(config-monitor-tenant)# no shut
```



第 21 章

出力の設定を実行している別 Cisco APIC に show の適用

このセクションでは使用する方法を説明、設定のエクスポート および 設定をインポート CLI を使用する、設定を実行している show 別 Cisco APIC 出力します。

- [インポートおよびエクスポートの設定について \(557 ページ\)](#)
- [インポートおよびエクスポート設定の注意事項と制限事項 \(557 ページ\)](#)
- [CLI 設定のエクスポート \(558 ページ\)](#)
- [CLI 設定のインポート \(559 ページ\)](#)

インポートおよびエクスポートの設定について

import config および **export config** コマンドを使用すると、別の Cisco APIC に **show running config** 出力が適用されます。このセクションではこれらのコマンドの注意事項を含み、コマンドを実行する方法を説明しています。

インポートおよびエクスポート設定の注意事項と制限事項

このセクションでは、**export config** および **import config** コマンドの注意事項および制限事項を説明します。

- 設定ファイルには、パスワードおよびその他の暗号化済みファイルは含まれません。
- 一部の REST API 設定は CTI 設定と互換性がない場合があります、これにより、Cisco APIC に設定ファイルを適用する際にエラーが発生する可能性があります。
- 一部の機能には、特定の順序の設定が必要です。CLI から実行されたときに、これらの設定が検証されます。ただし、REST API を使用した設定は検証されず、失われた設定によりインポートしたファイルの実行時にエラーが発生する可能性があります。

- インタラクティブ コマンドは「#」プレフィックスが付き、設定ファイルを実行している場合は無視されます。

CLI 設定のエクスポート

この手順では、テキスト ファイルへ設定をエクスポートする方法を示します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : dev4-ifc1# configure	コンフィギュレーション モードに入ります。
ステップ 2	leaf ID 例 : dev4-ifc1(config)# leaf 101	エクスポートする設定でリーフを特定します。
ステップ 3	interface ethernet slot/port 例 : dev4-ifc1(config-leaf)# interface ethernet 1/34	既存のイーサネット インターフェイスのスロット番号とポート番号を特定します。
ステップ 4	export-config result-file-name 例 : dev4-ifc1(config-leaf-if)# export-config /tmp/showRunnLeaf101.txt	指定されたファイル名に設定をエクスポートします。

例

この例では、エクスポート設定を設定する方法を示します。

```
dev4-ifc1# config
dev4-ifc1(config)# leaf 101
dev4-ifc1(config-leaf)# interface ethernet 1/34
dev4-ifc1(config-leaf-if)# export-config /tmp/showRunnLeaf101.txt
dev4-ifc1(config-leaf-if)# cat /tmp/showRunnLeaf101.txt
config
# Command: show running-config leaf 101 interface ethernet 1 / 34
# Time: Fri Sep 23 16:03:48 2016
  leaf 101
    interface ethernet 1/34
      switchport trunk allowed vlan 602 tenant t1 external-svi l3out l3ext1sub1
    exit
  exit
dev4-ifc1(config-leaf-if)#
```

CLI 設定のインポート

この手順では、テキスト ファイルから設定をインポートする方法を示します。

手順

	コマンドまたはアクション	目的
ステップ 1	import-config <i>file-name</i> 例 : <pre>dev4-ifc1(config-tenant)# import-config /tmp/showRunnLeaf101.txt config # Command: show running-config leaf 101 interface ethernet 1 / 34 # Time: Fri Sep 23 16:03:48 2016 leaf 101 interface ethernet 1/34 switchport trunk allowed vlan 602 tenant t1 external-svi l3out l3extlsub1 exit exit dev4-ifc1(config)#</pre>	



第 22 章

スケール プロファイル ポリシーの転送を設定する

- [スケール プロファイル ポリシーの転送についての概要 \(561 ページ\)](#)
- [スケール プロファイル ポリシーの転送のサポートされるプラットフォーム \(564 ページ\)](#)
- [NXOS スタイル CLI を使用したスケール プロファイル ポリシー転送の設定 \(564 ページ\)](#)

スケール プロファイル ポリシーの転送についての概要

スケール プロファイルの転送ポリシーは、別のスケーラビリティのオプションを提供します。次に例を示します。

- **デュアル スタック** : IPv6 設定に最大 12,000 エンドポイントのスケーラビリティと、IPv4 設定に最大 24,000 エンドポイントを提供します。
- **高 LPM** : 最長のプレフィックス一致 (LPM) スケールが 128,000 であり、ポリシー スケールが 8,000 である場合を除き、デュアルスタック プロファイルと同様のスケーラビリティを提供します。
- **IPv4 スケール** : IPv6 設定を持たないシステムが 48,000 IPv4 エンドポイントまでスケーラビリティを拡張できます。
- **高デュアル スタック** : 64,000 MAC エンドポイントおよび 64,000 Ipv4 エンドポイントまでのスケーラビリティを提供します。IPv6 エンドポイント スケールはスイッチハードウェア モデルによって 24,000/48,000 まで可能です。



(注) Cisco APIC リリース 3.2(1) では、TOR スイッチ ハードウェアによって、**[高デュアルスタック]** オプションを持つ転送スケールプロファイルには異なるスケールがあります。

- スイッチ名に FX がつく Cisco Nexus 9000 シリーズ TOR スイッチでは、高デュアルスタック オプションは、24,000 ではなく 48,000 Ipv6 エンドポイント、8,000 ではなく 128,000 ポリシーのスケラビリティを有します。
- スイッチ名に EX がつく Cisco Nexus 9000 シリーズ TOR スイッチでは、高デュアルスタック オプションは以前の APIC リリースと同じスケール値を有します。

詳細については、次の表を参照してください。

表 21: スケール プロファイル ポリシー転送のスケラビリティ

スケール プロファイル ポリシー転送のオプション	名前に EX がつく TOR スイッチ	名前に FX がつく TOR スイッチ
デュアルスタック	• EP MAC: 24,000 • EP IPv4: 24,000 • EP IPv6: 12,000 • LPM: 20,000 • Policy: 64,000 • Multicast: 8,000	以前のスイッチのデュアルスタック スケールと同じスケラビリティ数を有します。
高デュアルスタック	• EP MAC: 64,000 • EP IPv4: 64,000 • EP IPv6: 24,000 • LPM: 38,000 • Policy: 8,000 • Multicast: 0	• EP MAC: 64,000 • EP IPv4: 64,000 • EP IPv6: 48,000 • LPM: 38,000 • Policy: 128,000 • Multicast: 512

スケール プロファイル ポリシー転送のオプション	名前に EX がつく TOR スイッチ	名前に FX がつく TOR スイッチ
高 LPM	最長プレフィックス一致 (LPM) スケールが 128,000 であり、ポリシー スケールが 8,000 である場合を除き、デュアルスタック プロファイルと同様のスケラビリティを提供します。	以前のスイッチと同じスケラビリティ数を有します。
IPv4 スケール	• EP MAC: 48,000 • EP IPv4: 48,000 • EP IPv6: 0 • LPM: 38,000 • Policy: 60,000 • Multicast: 8,000	以前のスイッチの IPv4 スケールと同じスケラビリティ数を有します。



(注)

- IPv4 転送スケール プロファイル ポリシーが IPv6 設定をサポートしていないため、IPv4 転送スケール プロファイルで設定されたスイッチからすべての IPv6 設定を削除する必要があります。
- 高デュアルスタック プロファイルがコントラクトポリシー (8,000) のスケールサポートを削減したため、コントラクトスケールは以前のプロファイル展開に沿って削減される必要があります。
- 高デュアルスタックのように最小テナントマルチキャストスケールリーフプロファイルへ移行する前に、レイヤ 2 IGMP スヌーピング、レイヤ 3 IGMP、PIM 関連の設定を最初に無効にして、ハードウェアに古いマルチキャストの状態が含まれないようにすることをお勧めします。
- ノードにスケールプロファイルを適用すると、そのノードを手動でリロードする必要があります。サポートされていないスイッチは無視されます。サポートされているスイッチのリストについては、『[スケール プロファイル ポリシーの転送のサポートされるプラットフォーム \(564 ページ\)](#)』[英語]を参照してください。
- 異なるスケール プロファイルの設定項目に関連付けられている VPC はサポートされていません。VPC メンバーは、同じスケール プロファイル設定で設定する必要があります。

スケールプロファイルポリシーの転送のサポートされるプラットフォーム

スケール プロファイル ポリシーの転送は、次のスイッチでのみサポートされます。

- Cisco Nexus 9300-EX シリーズ スイッチ
- N9K-C9348GC-FXP
- N9K-C93108TC-FX
- N9K-C93180YC-FX

NX OS スタイル CLI を使用したスケール プロファイル ポリシー転送の設定

始める前に

スケール プロファイル ポリシー転送は、次のスケーリング オプションのいずれかに設定できます。

- [デュアルスタック] オプションは、IPv4 設定に最大 12,000 個のエンドポイントのスケラビリティおよび IPv4 設定に最大 24,000 個のエンドポイントのスケラビリティを提供します。
- [高 LPM] オプションは、最長プレフィックス一致 (LPM) スケールが 128,000 でありポリシー スケールが 8,000 の場合を除き、デュアルスタック同様のスケラビリティを提供します。
- [高デュアルスタック] オプションは、最大 64,000 MAC エンドポイントおよび 64,000 IPv4 エンドポイントのスケラビリティを提供します。スイッチ ハードウェア モデルに応じて、IPv6 エンドポイント スケールは 24,000/48,000 にできます。
- [Ipv4] オプションでは、IPv6 設定がないシステムが、最大 48,000 IPv4 エンドポイントに拡張できます。

スケール プロファイル ポリシー転送には、サポートされているスイッチが必要です。サポートされているスイッチのリストについては、『[スケールプロファイルポリシーの転送のサポートされるプラットフォーム \(564 ページ\)](#)』[英語]を参照してください。



(注)

- スケール プロファイル ポリシー転送をサポートするスイッチは、スケール プロファイル ポリシー転送が適用された後手動でリロードする必要があります。
- VPC の個別メンバのスケール プロファイルを変更することはできません。同じ VPC のメンバが異なるリーフ プロファイルに関連付けられている場合は、適用される新しいリーフ プロファイルがメンバとスケール プロファイルで作成される必要があります。

このセクションでは、NX OS スタイル CLI を使用してスケール プロファイル ポリシー転送を設定する方法を示します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apicl# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	no scale-profile name 例 : <code>apicl(config)# scale-profile testFwdScaleProf</code>	スケール プロファイル ポリシーを定義します。
ステップ 3	profile-type {dual-stack high-dual-stack high-lpm ipv4 } 例 : <code>apicl(config-scale-profile)# profile-type ipv4</code>	スケール プロファイル タイプ 転送を設定します。
ステップ 4	exit 例 : <code>apicl(config-scale-profile)# exit</code>	グローバル 設定に戻ります。
ステップ 5	template leaf-policy-group leaf_group_name 例 : <code>apicl(config)# template leaf-policy-group samplePolicyGrp</code>	リーフ ポリシー グループを定義します。
ステップ 6	scale-profile name 例 : <code>apicl(config-leaf-policy-group)# scale-profile testFwdScaleProf</code>	スケール プロファイル ポリシーおよびリーフ ポリシー グループの関係を設定します。

	コマンドまたはアクション	目的
		(注) スケール プロファイル ポリシー転送をサポートするスイッチは、スケールプロファイルポリシー転送が適用された後手動でリロードする必要があります。スケール プロファイル ポリシー転送をサポートするスイッチのリストについては、 スケール プロファイルポリシーの転送のサポートされるプラットフォーム (564 ページ) を参照してください。。
ステップ 7	exit 例 : <code>apic1(config-leaf-policy-group)# exit</code>	グローバル設定に戻ります。
ステップ 8	leaf-profile <i>leaf_profile_name</i> 例 : <code>apic1(config)# leaf-profile sampleLeafProf</code>	リーフ プロファイルを設定します。
ステップ 9	leaf-group <i>leaf_group_name</i> 例 : <code>apic1(config-leaf-profile)# leaf-group sampleLeafGrp</code>	リーフ スイッチのグループを指定します。
ステップ 10	leaf <i>leaf_group_number</i> 例 : <code>apic1(config-leaf-profile)# leaf 201</code>	リーフ グループにリーフ スイッチを追加します。
ステップ 11	leaf-policy-group <i>leaf_policy_group_name</i> 例 : <code>apic1(config-leaf-group)# leaf-policy-group samplePolicyGrp</code>	リーフ スイッチに関連付けられるリーフ ポリシー グループを指定します。
ステップ 12	exit 例 : <code>apic1(config-leaf-policy-group)# exit</code>	コマンド モードを終了します。
ステップ 13	[show] running-config 例 :	現在の実行コンフィギュレーションを表示します。

	コマンドまたはアクション	目的
	<pre> apicl(config)# show running-config # Command: show running-config scale-profile testFwdScaleProf # Time: Thu Jul 27 22:31:29 2017 scale-profile testFwdScaleProf profile-type ipv4 exit apicl(config-scale-profile)# </pre>	
ステップ 14	[show] template leaf-policy-group 例 :	現在の実行コンフィギュレーションを表示します。

例

この例では、IPv4 スケール プロファイル ポリシーを設定する方法を示します。

```

apicl# configure
apicl(config)# scale-profile testFwdScaleProf
apicl(config-scale-profile)# profile-type ipv4
apicl(config-scale-profile)# exit
apicl(config)# template leaf-policy-group samplePolicyGrp
apicl(config-leaf-policy-group)# scale-profile testFwdScaleProf
apicl(config-leaf-policy-group)# exit
apicl(config)# leaf-profile sampleLeafProf
apicl(config-leaf-profile)# leaf-group sampleLeafGrp
apicl(config-leaf-profile)# leaf 201
apicl(config-leaf-group)# leaf-policy-group samplePolicyGrp
apicl(config-leaf-group)# show running-config scale-profile testFwdScaleProf
# Command: show running-config scale-profile testFwdScaleProf
# Time: Thu Jul 27 22:31:29 2017
scale-profile testFwdScaleProf
profile-type ipv4
exit
apicl(config-leaf-group)# show running-config template leaf-policy-group samplePolicyGrp
# Command: show running-config template leaf-policy-group samplePolicyGrp
# Time: Tue Aug 1 11:19:44 2017
template leaf-policy-group samplePolicyGrp
scale-profile testFwdScaleProf
exit
apicl(config-leaf-group)# show running-config leaf-profile sampleLeafProf
# Command: show running-config leaf-profile sampleLeafProf
# Time: Tue Aug 1 11:19:58 2017
leaf-profile sampleLeafProf
leaf-group sampleLeafGrp
leaf 201
leaf-policy-group samplePolicyGrp
exit

```




付録 **A**

CLI を使用して確認されたスケーラビリティ

- CLI スケーラビリティの制限値 (569 ページ)

CLI スケーラビリティの制限値

設定可能なオプション	規模
テナントの数	500
レイヤ 3 (L3) コンテキストの数	300
エンドポイント グループ (EPG) の数	3,500
エンドポイント (EP) の数	20,000
ブリッジ ドメイン (BD) の数	3,500
BGP、OSPFセッション、およびEIGRP (外部接続用) の合計数	300
vPC の最大数	48
PC (アクセス ポート) の最大数	48
アクセス ポートあたりのカプセル化の最大数	1,750
マルチキャスト グループの数	8,000
vzAny で指定されたコントラクトの最大数	16
vzAny で使用されたコントラクトの最大数	16
エンドポイント グループあたりのカプセル化の最大数	スタティック 2、ダイナミック 1

設定可能なオプション	規模
セキュリティ TCAM サイズ	4,000
VRF の数	500
Separate-Config-Set	
テナント	100
エンドポイント グループ	1,000
ブリッジ ドメイン	500
VRF	100
SPAN 宛先	3
NTP サーバ	2
コントラクト	100
DNS サーバ	2
Syslog サーバ	1



付録 **B**

使用例：トランジット トポロジによる **Three-Tier** アプリケーション

- [トランジット トポロジによる Three-Tier アプリケーションの展開について \(571 ページ\)](#)
- [3 層アプリケーションの展開 \(573 ページ\)](#)
- [OSPF と BGP によるトランジット ルーティング \(575 ページ\)](#)

トランジット トポロジによる **Three-Tier** アプリケーション の展開について

通常、APIC ファブリックは、テナント ネットワーク内の **Three-Tier** アプリケーションをホストします。この例では、アプリケーションは 3 台のサーバ（Web サーバ、アプリケーションサーバ、およびデータベース サーバ）を使用して実行されます。**Three-Tier** アプリケーションの例については、次の図を参照してください。



コントラクトは、エンドポイントグループ間 (EPG 間) の通信をイネーブルにするポリシーです。これらのポリシーは、アプリケーション層間の通信を指定するルールです。コントラクト

が EPG に付属していない場合、EPG 間の通信はデフォルトでディセーブルになります。EPG 内の通信は常に許可されているので、EPG 内の通信には契約は必要ありません。

トランジット ルーティングについて

トランジット ルーティングは、他のルーティング ドメインとの双方向の再配送を行うために境界ルータを有効にします。双方向の再配布により、あるルーティング ドメインから別のルーティング ドメインへルーティング情報が渡されます。そのような再配布により、ACI ファブリックはさまざまなルーティング ドメイン間の完全な IP 接続を提供します。これにより、ルーティング ドメイン間のバックアップ パスを有効にすることで冗長接続を提供することもできます。詳細については、『*Cisco ACI Fundamentals Guide*』の「ACI Transit Routing」を参照してください。

3 層アプリケーションの展開

テナント VRF およびブリッジ ドメインを設定します。

```
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# contract enforce
apic1(config-tenant)# bridge-domain b1
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant)# interface bridge-domain b1
apic1(config-tenant-interface)# ip address 159.10.10.1/24 scope public
apic1(config-tenant-interface)# exit
```

Web、アプリケーション、データベースの 3 つの EPG を設定します。

```
apic1(config-tenant)# application retail
apic1(config-tenant-app)# epɡ web
apic1(config-tenant-app-epɡ)# bridge-domain member b1
apic1(config-tenant-app-epɡ)# contract provider web
apic1(config-tenant-app-epɡ)# contract consumer app
apic1(config-tenant-app)# epɡ app
apic1(config-tenant-app-epɡ)# bridge-domain member b1
apic1(config-tenant-app-epɡ)# contract provider app
apic1(config-tenant-app-epɡ)# contract consumer db
apic1(config-tenant-app)# epɡ db
apic1(config-tenant-app-epɡ)# bridge-domain member b1
apic1(config-tenant-app-epɡ)# contract provider db
```

VLAN ドメインを設定します。

```
apic1(config)# vlan-domain dom100
apic1(config-vlan)# vlan 100-200
```

ポート チャネルを作成し、Web EPG を展開します。

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/2-5
```

```

apic1(config-leaf-if)# channel-group po1

apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# vlan-domain member dom100
apic1(config-leaf-if)# switchport trunk allowed vlan 101 tenant t1 application retail
epg web

```

vPC を作成し、アプリケーションとデータベース EPG を展開します。

```

apic1(config)# leaf 101,102
apic1(config-leaf)# interface ethernet 1/6,1/7
apic1(config-leaf-if)# channel-group vpc1 vpc

apic1(config)# vpc domain explicit 100 leaf 101 102
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# interface vpc vpc1
apic1(config-vpc-if)# vlan-domain member dom100
apic1(config-vpc-if)# switchport trunk allowed vlan 102 tenant t1 application retail epg
app
apic1(config-vpc-if)# switchport trunk allowed vlan 103 tenant t1 application retail epg
db

```

MP-BGP を設定します。

```

apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 100
apic1(config-bgp-fabric)# route-reflector spine 104,105

```

外部 L3 EPG を設定します。

```

apic1(config-tenant)# external-l3 epg l3epg1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 173.10.1.0/24
apic1(config-tenant-l3ext-epg)# contract consumer web

```

リーフ、ルート マップ上で VRF を設定し、外部 L3 EPG を展開します。

```

apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg l3epg1
apic1(config-leaf-vrf)# route-map map1
apic1(config-leaf-vrf-route-map)# match bridge-domain b1

```

サブインターフェイス上で OSPF エリアを設定します。

```

apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.1 route-map map1 out
apic1(config-leaf)# interface ethernet 1/2
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# interface ethernet 1/2.150
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 169.10.10.1/24

```

```
apic1(config-leaf-if)# ip router ospf default area 0.0.0.1
```

フィルタを設定します。

```
apic1(config-tenant)# access-list http
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# match tcp dest 443

apic1(config-tenant)# access-list rmi
apic1(config-tenant-acl)# match tcp dest 1099

apic1(config-tenant)# access-list sql
apic1(config-tenant-acl)# match tcp dest 1521
```

コントラクトを設定します。

```
apic1(config-tenant)# contract rmi
apic1(config-tenant-contract)# subject rmi
apic1(config-tenant-contract-subj)# access-group rmi both

apic1(config-tenant)# contract web
apic1(config-tenant-contract)# subject web
apic1(config-tenant-contract-subj)# access-group http both

apic1(config-tenant)# contract db
apic1(config-tenant-contract)# subject sql
apic1(config-tenant-contract-subj)# access-group sql both
```

OSPF と BGP によるトランジットルーティング

この手順は、この章の *Three-Tier* アプリケーションの展開で説明されている *Three-Tier* アプリケーションの Site1 と Site2 との間のトランジットルーティングを設定します。

Site2 の外部 L3 EPG (l3epg2) を設定します。

```
apic1(config-tenant)# external-l3 epg l3epg2
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 174.10.1.0/24
apic1(config-tenant-l3ext-epg)# contract consumer transit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg l3epg2
```

Site1 に対応する外部 SVI およびエクスポート ルート上に BGP 接続を設定します。

```
apic1(config)# leaf 102
apic1(config-leaf-vrf)# route-map map200
apic1(config-leaf-vrf-route-map)# ip prefix-list p1 match 173.10.1.0/24
apic1(config-leaf-vrf-route-map)# match prefix-list p1
apic1(config-leaf-vrf-route-map-match)# set community extended 200:1 replace

apic1(config-leaf)# interface vlan 160
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
```

```
apic1(config-leaf-if)# ip address 208.1.1.2/24
apic1(config-leaf)# interface ethernet 1/11
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 160 tenant t1 external-svi

apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 208.1.1.1
apic1(config-leaf-bgp-vrf-neighbor)# remote-as 200
apic1(config-leaf-bgp-vrf-neighbor)# update-source vlan 160
apic1(config-leaf-bgp-vrf-neighbor)# route-map map200 out
```

l3epg2 (Site2) との接続を確立するために l3epg1 (Site1) にコントラクト プロバイダーを設定します。

```
apic1(config-tenant)# external-l3 epg l3epg1
apic1(config-tenant-l3ext-epg)# contract provider transit
```

Site2 に対応するルートをエクスポートするために Site1 にルート マップを設定します。

```
apic1(config)# leaf 103
apic1(config-leaf-vrf)# route-map map1
apic1(config-leaf-vrf-route-map)# ip prefix-list p1 match 174.10.1.0/24
apic1(config-leaf-vrf-route-map)# match prefix-list p1
apic1(config-leaf-vrf-route-map-match)# set metric 100
```

トランジット ルーティングの ACL とコントラクトを設定します。

```
apic1(config)# tenant t1
apic1(config-tenant)# access-list acl1
apic1(config-tenant-acl)# match ip
apic1(config-tenant)# contract transit
apic1(config-tenant-contract)# subject ip
apic1(config-tenant-contract-subj)# access-group acl1 both
```




付録 C

例：show コマンド

- [例：show コマンド \(577 ページ\)](#)

例：show コマンド

show running-config

「ローカル」の **running-config** を現在のモードに表示します。

```
apic1(config)# leaf 103
apic1(config-leaf)# interface ethernet 1/2.150
apic1(config-leaf-if)# show running-config
# Command: show running-config leaf 103 interface ethernet 1 / 2 . 150
# Time: Tue Dec 8 08:08:37 2015
leaf 103
  interface ethernet 1/2.150
    vrf member tenant t1 vrf v1
    ip address 169.10.10.1/24
    ip router ospf default area 0.0.0.1
  exit
exit
```

フィルタを使用して **running-config** を表示します。

```
apic1(config-leaf)# interface ethernet 1/2.150
apic1(config-leaf-if)# show running-config leaf 103
# Command: show running-config leaf 103
# Time: Tue Dec 8 08:10:02 2015
leaf 103
  vrf context tenant t1 vrf v1
    external-l3 epg l3epg1
    route-map map1
      ip prefix-list p1 permit 181.1.1.0/24
      match bridge-domain b1
      match prefix-list p1
    ...
```

show vpc, port-channel**vpc map** を表示します

```
apic1(config-leaf-if)# show vpc map
```

Legends:

N/D : Not Deployed

Virtual Port-Channel Name	Domain	VPC	Leaf Id, Name	Fex Id	PC Id	Ports
vpc1	100	1	101,leaf1		po2	eth1/6-7, eth1/40-41
vpc1	100	1	102,leaf2		po1	eth1/6-7, eth1/40-41

port-channel map を表示します

```
apic1(config-leaf-if)# show port-channel map
```

Legends:

N/D : Not Deployed

PC: Port Channel

VPC: Virtual Port Channel

Port-Channel Name	Type	Leaf ID, Name	Fex Id	Port Channel	Ports
po1	PC	101,leaf1		po1	eth1/2-5, eth1/32-33
po1	PC	102,leaf2		po2	eth1/32-33
vpc1	VPC	101,leaf1		po2	eth1/6-7, eth1/40-41
vpc1	VPC	102,leaf2		po1	eth1/6-7, eth1/40-41

show vlan-domain**vlan-domain name dom100** を表示します

```
apic1# show vlan-domain name dom100
```

Legend:

vlanscope: L (Portlocal). Default is global

vlan-domain : dom100 Type : All

vlan : 100-200(static)

Leaf Vlan	Interface	Vlan	Type	Usage	Operational State	Operational
101	PC: po1	101	App-Epg	Tenant: t1 App: retail Epg: web	b1: down web: down	b1: vlan-18 web: vlan-21

```

101,102   vPC: vpc1   102   App-Epg   Tenant: t1   b1: down      b1: vlan-18
                                     App: retail   app: down      app: vlan-19
                                     Epg: app
101,102   vPC: vpc1   103   App-Epg   Tenant: t1   b1: down      b1: vlan-18
                                     App: retail   db: down       db: vlan-20
                                     Epg: db
102        eth1/11   160   Ext-svi   Tenant: t1   12: down      vlan-18
                                     Vrf: v1       13: down
103        eth1/2    150   Ext-subIf Tenant: t1   -              eth1/2.14
                                     Vrf: v1

```

show tenant**tenant t1 detail** を表示します

```
apic1# show tenant t1 detail
```

Detailed view for Tenant t1

Security Information:

Security Domain

VRF Information:

VRF	Policy Enforcement
v1	enforced

Bridge-Domain Information:

BD	VRF
b1	v1

Static VLAN Information:

Node	VLANs	VLAN Domains
101	101	dom100
101 102	102,103	dom100

Static Application EPg Information:

Node	Interface	App:AEpg	BD
Contract			
-----	-----	-----	-----
101	port-channel po1	retail:web	b1
web,			
app			
101 102	vpc vpc1	retail:db,retail:app	b1
app,			
db			

Application EPg Information:

App:AEpg	BD
retail:app	b1

```

retail:db          b1
retail:web         b1

```

External L2 EPg Information:

```
external-l2        BD
-----
```

External L3 EPg Information:

```
external-l3        VRF
-----
```

```
l3epg1             v1
l3epg2             v1

```

show external-l3

external-l3 interfaces を表示します

```
apic1# show external-l3 interfaces
```

Node	Tenant	VRF	Interface	Oper Interface	IP Address
102	t1	v1	vlan-160 eth1/11	vlan18	208.1.1.2/24
up				eth1/11	
103	t1	v1	eth1/2.150	eth1/2.14	169.10.10.1/24
up					

external-l3 epg を表示します

```
apic1# show external-l3 epg
```

Name	Flags	Match	Node	Entry
Oper State				
t1:	vxlan: 2457600	173.10.1.0/24		
l3epg1	vrf: v1			
	Target dscp: unspecified			
	qosclass: unspecified			
	Contracts			
	Provided: transit			
	Consumed: web			
t1:	vxlan: 2457600	173.10.1.0/24	node-103	173.10.1.0/24
disabled				
l3epg2	vrf: v1		node-101	173.10.1.0/24
disabled				
	Target dscp: unspecified		node-102	173.10.1.0/24
disabled				
	qosclass: unspecified			
	Contracts			
	Provided:			
	Consumed: transit			

external-l3 ospf を表示します

```

apic1# show external-13 ospf tenant t1 vrf v1
Area Id      : 0.0.0.1
Tenant       : t1
Vrf          : v1
User Config :
Node ID      Area Properties
-----
103          Type: nssa, Cost: 1, Control: redistribute,summary

Configuration :          Operational

Node ID  Router ID      Route Map      Area Oper. Props
-----
103      10.1.0.103      map1          Type: nssa, Cost: 1, Control:
redistribute,summary, AreaId:
0.0.0.1

Interfaces :
Configuration :          Operational

Node ID  Interface      IP Address      Oper. Intf  Oper. State
-----
103      eth1/2.150      169.10.10.1/24  eth1/2.14   down

```

external-13 bgp を表示します

```

apic1# show external-13 bgp
flags_match : Properties in logical and concrete MOs are symmetric
Tenant, vrf : t1, v1

```

Node	Neighbor Session Status	Flags	RouteMap	SourceIf	Oper Peer Status
102	208.1.1.1	Allowed Self As Count: 3 TTL: 1	no (in) map200 (out)	Vlan 160	vlan18 flags_match

external-13 route-map を表示します

```

apic1# show external-13 route-map
Tenant : t1 VRF: v1
Table1: Route Map Configuration

```

Node	Routemap	Type	Name	Match	Set Attributes
102	map200	PfxList	p1	100.100.100.0/24	Community
				173.10.1.0/24	
103	map1	PfxList	p1	181.1.1.0/24	Metric: 100
103	map1	BD	b1	159.10.10.1/24	

Table 2 : Route Map Usage

Node	Routemap	Protocol	Neighbors	Operational Attributes

例 : show コマンド

102	map200	bgp	208.1.1.1	Pfx List: p1 100.100.100.0/24 173.10.1.0/24 ::/0
103	map1	ospf	0.0.0.1	Pfx List: p1 Metric: 100 181.1.1.0/24 ::/0