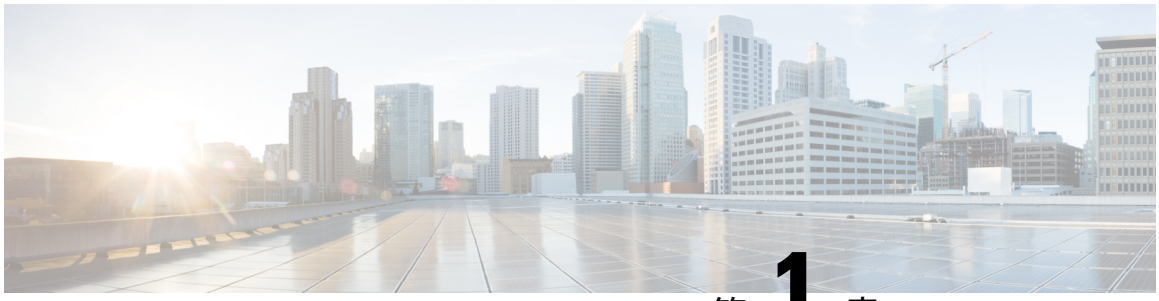
# Cisco ACI Multi-Site トラブルシューティング ガイド、リリース 3.1(x)

**初版**：2020 年 5 月 11 日

**最終更新**：2020 年 5 月 11 日

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

# 目 次



第 1 章

# 新機能と変更情報

この章は、次の項で構成されています。



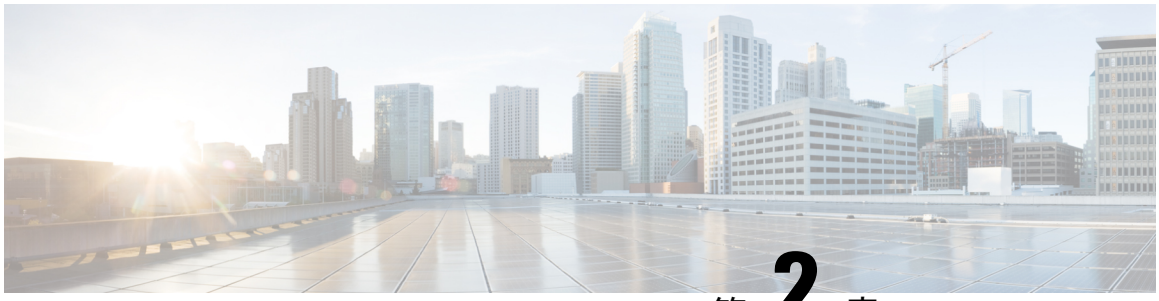
## 新機能と変更情報

次の表に、このガイドの最初に発行されたリリースから現在のリリースまでに、このガイドの編成と機能に加えられた大幅な変更の概要を示します。テーブルは、ガイドに加えられたすべての変更のすべてを網羅したリストを提供しているわけではありません。

**表 *1*：最新のアップデート**

| リリース | 新機能またはアップデート | 参照先 |
|---|---|---|
| 3.0(1) | -- | -- |

新機能と変更情報



第 2 章

# 概要およびトラブルシューティング基礎

この章は、次の項で構成されています。



## このガイドで取り上げるトピック

このガイドの各章では、Cisco ACI マルチサイト Orchestrator の一般的な問題を解決するために使用できるトラブルシューティング ツールおよびヒントについて説明します。各章で取り上げるトピックを以下に要約します。

トラブルシューティング ツール：次に関する情報を提供します。

- Multi-Site トラブルシューティング ツールを使用して、トラブルシューティング レポートを生成し、API コール ログを生成し、データ収集のために VM にログオンし、実行ログを読み取り、マイクロサービスがアクティブであることを確認し、Cisco APIC サイトでポリシー解決を確認する方法について説明します。
  - 整合性チェッカー
  - API コール ログの生成
  - Docker コンテナ情報
  - ログの実行
  - Multi-Site マイクロサービス
  - APIC ポリシー解決

インストール、アップグレード、および再起動 — 以下に関する情報を提供します。

- Orchestrator ノードの再起動、置換、または再配置
- MongoDB のバックアップと復元

- 初期インストール後の NTP 設定の構成
- Orchestrator のシークレット ファイルとキー ファイルの変更

ユーザー — ユーザー認証の問題のトラブルシューティングに関する情報を提供します。

プラットフォームのヘルス — 以下に関する情報を提供します。

- トラブルシューティング レポートの生成およびダウンロード
- Docker サービスの検査
- 欠落しているノード ラベルの問題の解決
- サイト間トラフィック フローと BGP セッション
- ユニキャストおよびマルチキャスト トラフィック障害

テナントとスキーマ — 以下に関する情報を提供します。

- ポリシー展開エラー
- REST API を使用したテナント ポリシー レポート
- テンプレートおよびスキーマの展開解除

*Multipod* および *Multi-Site* — Multipod および Multi-Site の問題のトラブルシューティングに関する情報を提供します。

*NX-OS* ハードウェア テーブルの確認—以下に関する情報を提供します。

- Endpoint manager 学習
- BGP EVPN ルーティング テーブル
- VNID、S クラス、および VTEP マッピング
- ライン カード ハードウェア テーブル

# Orchestrator GUI がロードされていません

場合によっては、一般的なブラウザが情報をキャッシュする方法が原因で、Orchestrator GUI が完全に起動せず、ロードの進行状況画面が引き続き表示されることがあります。

**ステップ 1** ブラウザ キャッシュをクリアします。

具体的な手順は、使用しているブラウザの種類によって異なります。

**ステップ 2** Orchestrator GUI をリロードします。

# トラブルシューティングの基本

このセクションでは、マルチサイトでの作業中に問題が発生したときに最初に実行する手順について説明します。このガイドの他の章では、1 つ以上の特定の機能に関連する問題について説明します。

### 始める前に

トラブルシューティング ツール （7 ページ） にリストされているツールに慣れましょう。

**ステップ 1** 問題が マルチサイト に関連しているかどうかを確認します。

マルチサイト Orchestrator で問題が発生している場合は、まず次の点を確認して、問題が マルチサイト に関連しているかどうかを判断してください。いずれかの質問に対する答えが「いいえ」の場合、その問題はマルチサイトに関連している可能性があります。すべての答えが「いいえ」の場合、APIC、スイッチ、サイト間ネットワーク、または WAN に関連している可能性があります。

a) マルチサイト はアクセス可能ですか？

b) トラフィックがフローしていない場合...

REST API を使用したテナント ポリシー レポートの生成 （58 ページ） の説明に従って、APIC ポリシー レポートを生成します。次のコマンドの出力を確認します。

- 予想されるすべての MO が APIC サイトに展開されていますか？
- 予想されるすべての MO は、APIC サイト上で正しいプロパティ値を持っていますか？
- VRF、BD、EPG、および L3InstP は、すべてのサイトで正しいマッピングを持っていますか？
- EPG に正しいピア コンテキスト Dn がありますか？

c) トラフィックがフローしているが、そうなるべきでない場合...

APIC サイトでのポリシー解決の確認 （18 ページ） の説明に従って、ポリシー解像度情報を収集します。次のコマンドの出力を確認します。

- すべての MO がそこに存在せず、実際には APIC サイトに展開されていないことが良そうされていますか？
- 関連するすべての MO は、APIC サイトで正しいプロパティ値を持っていますか？

**ステップ 2** マルチサイト のどの部分に問題があるかを特定します。

a) すべての Docker サービスが稼働していることを確認します。

詳細については、Docker コンテナ情報の収集 （14 ページ）を参照してください。

b) 実行ログにエラーがないか確認する

詳細については、実行ログの読み取り （17 ページ）を参照してください。

c) 問題がないか、APIC ポリシー レポートを確認してください。

詳細については、REST API を使用したテナント ポリシー レポートの生成 （58 ページ）を参照してください。

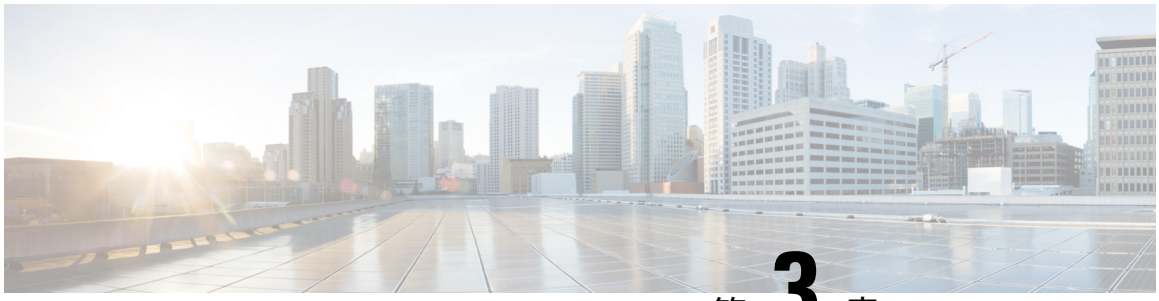
d) 接続の問題がないかどうかを確認します。

GUI の [ダッシュボード（Dashboard）] タブに接続の問題が表示された場合は、次のいずれかを確認してください。

- 「BGP ピアリングで構成されているサイトがありません」または「BGP セッションに失敗しました」エラーが表示された場合は、サイト間 BGP セッションのトラブルシューティング （48 ページ）を参照してください。
- ユニキャスト/マルチキャストの失敗が表示された場合は、ユニキャストまたはマルチキャスト トラフィック障害のトラブルシューティング （50 ページ）またはMulti-Site マルチキャスト機能のトラブルシューティング （51 ページ）を参照してください。

**ステップ 3** スキーマおよびテンプレートを再展開します。

スキーマまたはテンプレートに関連する問題を特定して修正したら、スキーマおよびテンプレートの展開解除 （58 ページ）の説明に従って、それらを展開解除してから再展開します。



第 3 章

# トラブルシューティング ツール

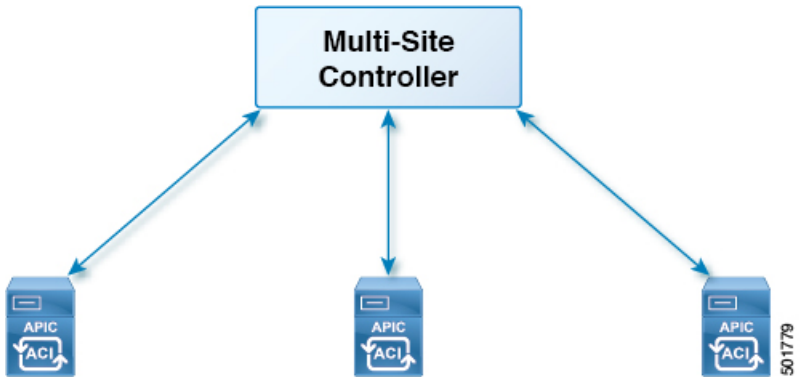
この章は、次の項で構成されています。



## 整合性チェッカーの概要

整合性チェッカーは、最初の展開操作の後に展開を検証し、このツールの結果を Cisco ACI マルチサイト ユーザー インターフェイスに統合します。 この機能は、クロス マッピングを検証します。展開されたテンプレートでのみ使用でき、少なくとも 2 つのサイトにまたがっており、次のポリシーの少なくとも 1 つを含んでいます。

- EPG
- VRF
- BD
- 外部 EPG



# サイト全体に展開されたテンプレートの確認

このセクションでは、サイト全体に展開されたテンプレートを検証する方法について説明します。

### 始める前に

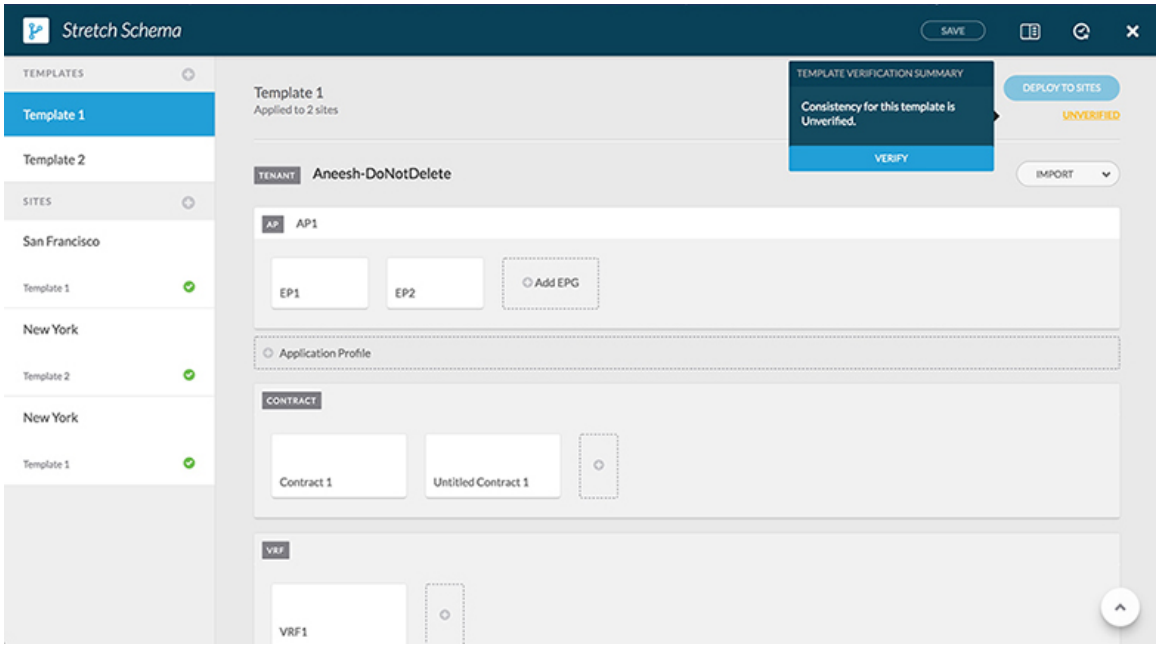
- 少なくとも 2 つのストレッチされたサイトに分散され、次のポリシーの少なくとも 1 つを含むテンプレート：
  - EPG
  - VRF
  - BD
  - 外部 EPG

**ステップ 1** マルチサイト GUI にログインします。

**ステップ 2** **[メイン メニュー（Main Menu）]**で **[スキーマ（Schemas）]**をクリックし、**[スキーマ リスト（Schema List）] ページで適切な schema_name を選択します。**

**ステップ 3** 展開されたテンプレートをクリックします。

**ステップ 4** 右上隅の **unverified** をクリックします。




**ステップ 5** **[テンプレート検証の概要（TEMPLATE VERIFICATION SUMMARY）] ダイアログ ボックスで、[検証（VERIFY）] をクリックします。**

ポップアップ メッセージが表示されます。

整合性検証が正常にトリガされました。

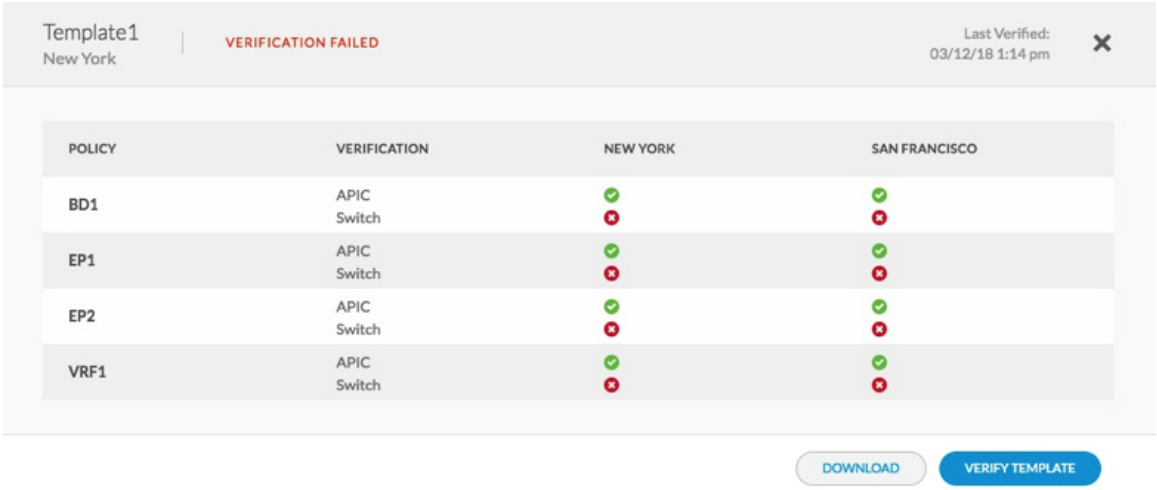
**ステップ 6** 検証ステータスは次のいずれかになります。

- **検証成功** — 何もする必要はありません。
- **検証失敗** — アクションが必要です。

a) 検証に失敗した場合は、**[検証に失敗しました（VERIFICATION FAILED）]** をクリックします。

b) **[テンプレート検証の概要（TEMPLATE VERIFICATION SUMMARY）]** ダイアログ ボックスで、失敗したサイトの鉛筆アイコンをクリックすると、テンプレートの詳細レポートが表示されます。

**例**：




問題の説明については、赤い **[x]** にカーソルを合わせます。この問題は、**[見つかりません（Not Found）]**(検索不可能）または **[不一致（Mismatch）]**（正しく構成されていない）のどちらかです。

c) **[ダウンロード（DOWNLOAD）]** または **[テンプレート検証（VERIFY TEMPLATE）]** のどちらかをクリックできます。

- **ダウンロード**—現在のサイトのみにレポートを提供します。
- **テンプレート検証**—すべてのサイト上で検証されたテンプレートを提供します。

# 展開されたテンプレートすべてにスケジュールされた検証のセットアップ

このセクションでは、展開されたすべてのテンプレートに対して、テナントごとにスケジュールされた検証を設定する方法について説明します。

**ステップ 1** マルチサイト GUI にログインします。

**ステップ 2** **[メイン メニュー（Main Menu）]**で **[テナント（Tenant）]** をクリックし、[テナント リスト（Tenant List）] ページで適切な **tenant_name** の *[スケジュールの設定（Set Schedule）]* をクリックします。

**ステップ 3** **整合性チェッカー[スケジューラ設定（Scheduler Settings）]** で、**[無効化スケジュール（Disabler Schedule）]** のチェックをオフにして、時間と頻度を選択します。

a) [OK] をクリックします。

# エラーのトラブルシューティング

このセクションでは、エラーをトラブルシューティングする方法を説明します。

**ステップ 1** マルチサイト GUI にログインします。

**ステップ 2** **[ダッシュボード（Dashboard）]** の **[スキーマの健全性（SCHEMA HEALTH）]** セクションのビュー別フィールドで、スキーマ検証アイコンをクリックします。

サイト内の小さな四角は、スキーマ内のテンプレートを表します。

ひと目で、何が合格、不合格、または未検証かが分かります。

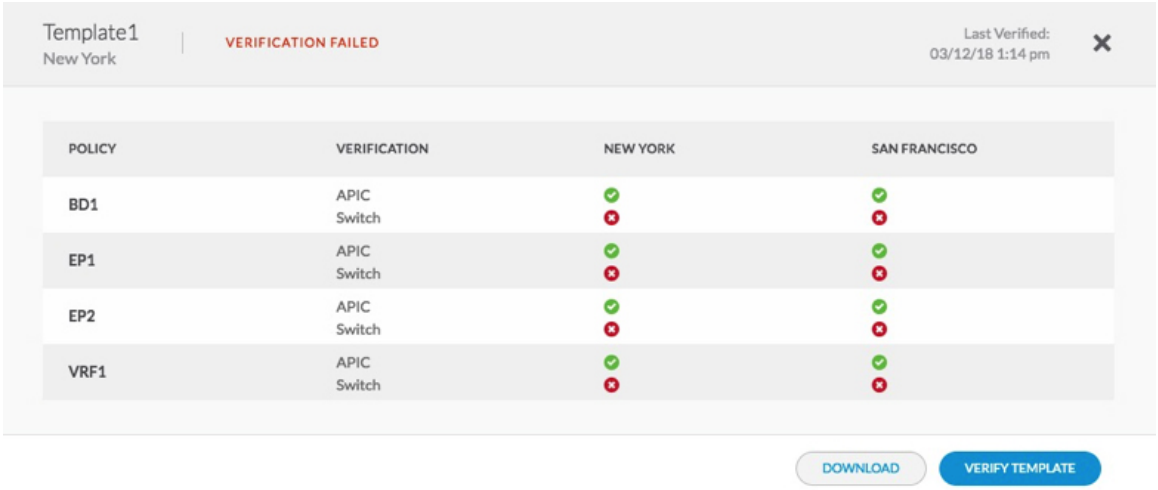
- **合格** — 緑色で表示されます。
- **不合格** — 赤色で表示されます。
- **未検証** — 黄色で表示されます。

**ステップ 3** サイトを含むスキーマを赤で展開して、テンプレートを表示します。

**ステップ 4** 赤いサイトにカーソルを合わせると、**[不合格（FAILED）]** と表示されます。

**ステップ 5** [不合格（FAILED）] サイトをクリックすると、より詳細なレポートが表示されます。

**例**：

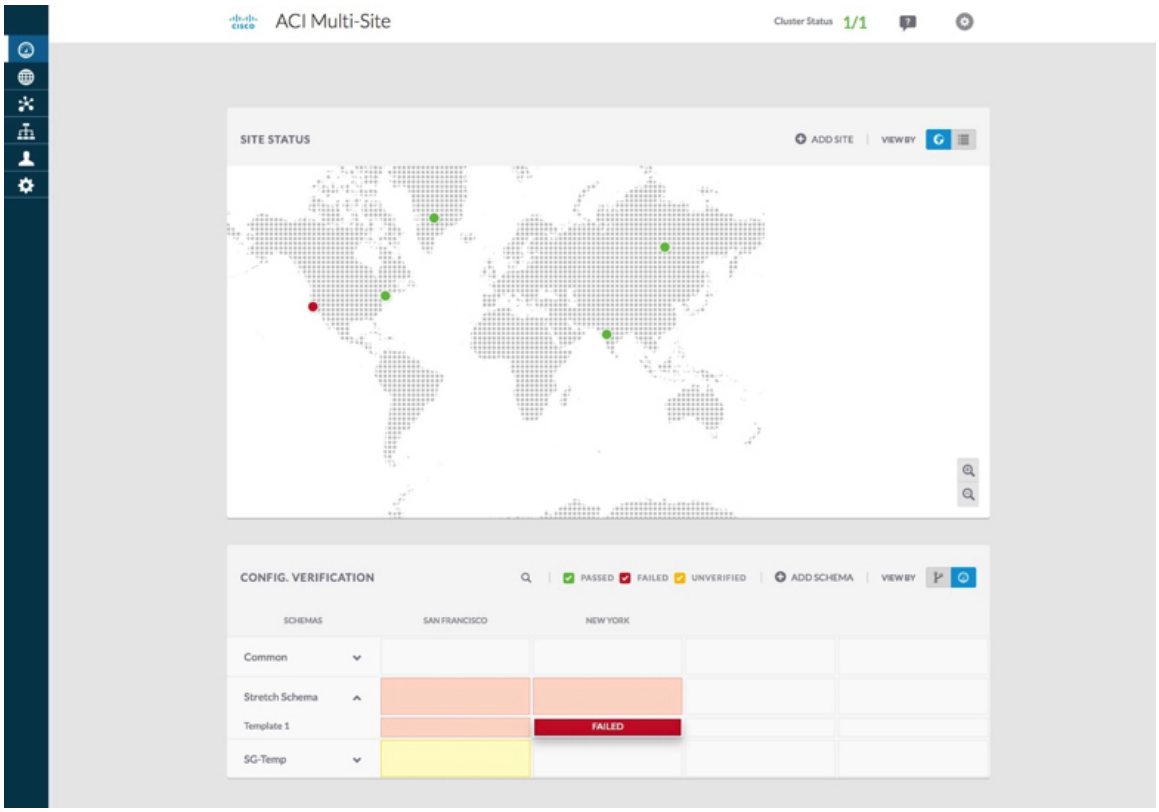



問題の説明の赤い **[x]** にカーソルを合わせる場合。この問題は、**[見つかりません（Not Found）]**(検索不可能）または **[不一致（Mismatch）]**（正しく構成されていない）のどちらかです。

a) **[ダウンロード（DOWNLOAD）]** または **[テンプレート検証（VERIFY TEMPLATE）]** のどちらかをクリックできます。

- **ダウンロード**—現在のサイトのみにレポートを提供します。
- **テンプレート検証**—すべてのサイト上で検証されたテンプレートを提供します。

**ステップ 6** 合格、不合格、または未検証のテンプレートを確認することもできます。




**ステップ 7** （オプション）スキーマ全体を検証し、**[...]** をクリックして、**[スキーマの検証（Verify Schema）]** を選択できます。

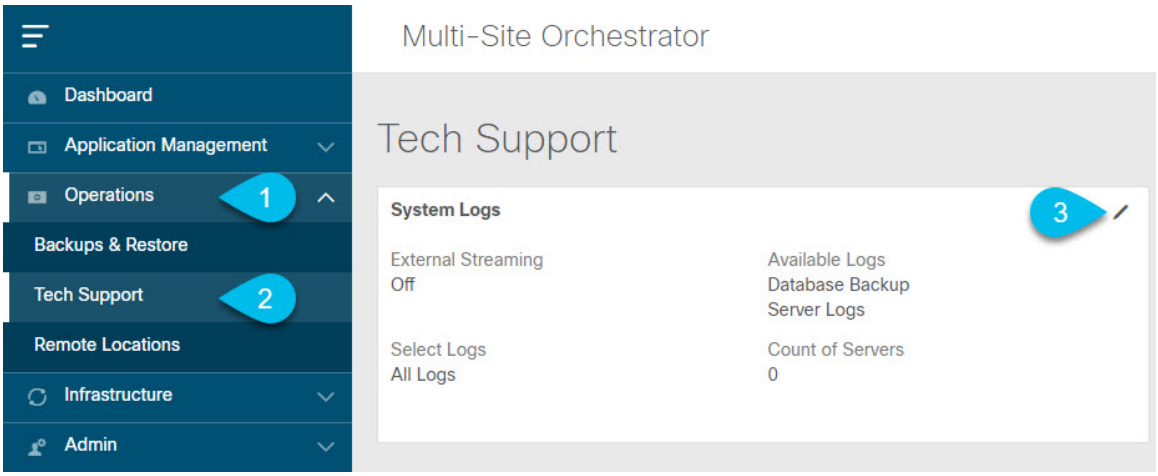
**ステップ 8** （オプション）EPG、BD、VRF、または外部 EPG で検索して、このポリシーが含まれているスキーマを見つけることができます。

# システム ログのダウンロード

このセクションでは、Cisco ACI マルチサイト Orchestrator により管理されているすべてのスキーマ、サイト、テナント、およびユーザのトラブルシューティング レポートとインフラストラクチャ ログ ファイルを生成します。

**ステップ 1** マルチサイト Orchestrator GUI にログインします。

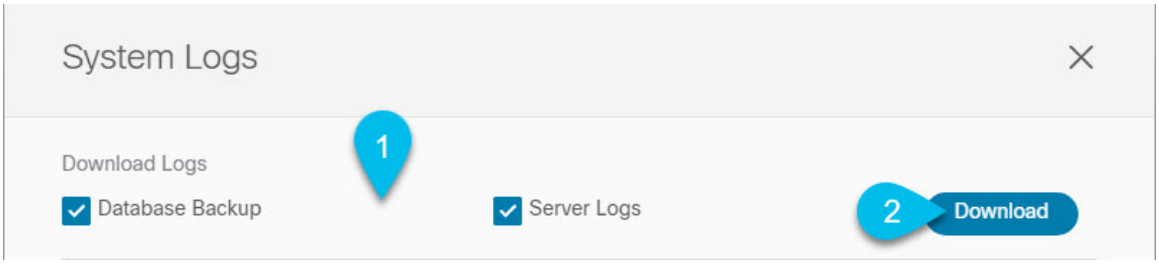
**ステップ 2** **[システムログ (System Logs)]** 画面を開きます。




a) メインメニューで、**[操作 (Operations)]** > **[テクニカル サポート (Tech Support)]**を選択します。

b) **[システム ログ (System Logs)]** フレームの右上隅にある編集ボタンをクリックします。

**ステップ 3** ログをダウンロードします。




a) ダウンロードするログを選択します。

b) [ダウンロード（Download）] ボタンをクリックします。

選択した項目のアーカイブがシステムにダウンロードされます。このレポートには、次の情報が含まれています。

- JSON フォーマットでのすべてのスキーマ
- JSON フォーマットでのすべてのサイト定義
- JSON フォーマットでのすべてのテナント定義
- JSON フォーマットでのすべてのユーザ定義
- `infra_logs.txt` ファイル内のコンテナのすべてのログ

# Docker コンテナ情報の収集

Orchestrator VM の 1 つにログインして、特定のコンテナの Docker サービスとそのログに関する情報を収集できます。次のチート シートには、多くの便利な Docker コマンドが記載されています。https://www.docker.com/sites/default/files/Docker_CheatSheet_08.09.2016_0.pdf

### Docker コンテナの健全性の検査

Docker サービスの正常性を検査するには、`docker service ls` コマンドを使用できます。コマンドの出力には、各サービスの現在のヘルス ステータスが一覧表示されます。`[REPLICAS]` 列に表示されるように、すべてのサービスですべてのコンテナが複製されている必要があります。いずれかがダウンしている場合は、対処が必要な問題が発生している可能性があります。

```
# docker service ls
ID                  NAME                        MODE                REPLICAS  [...]
ve5m9lwb1qc4        msc_auditservice            replicated          1/1       [...]
bl0op2eli7bp        msc_authyldapservice        replicated          1/1       [...]
uxc6pgzficls        msc_authytacacsservice      replicated          1/1       [...]
qcws6ta7abwo        msc_backupservice           global              3/3       [...]
r4p3opyf5dkm        msc_cloudsecservice         replicated          1/1       [...]
xrm0c9vof3r8        msc_consistencyservice      replicated          1/1       [...]
le4gy9kov7ey        msc_endpointservice         replicated          1/1       [...]
micd93h5gj97        msc_executionengine         replicated          1/1       [...]
6wxh4mgnnfi9        msc_jobschedulerservice     replicated          1/1       [...]
lrj1764xw91g        msc_kong                    global              3/3       [...]
n351htjnks75        msc_kongdb                  replicated          1/1       [...]
xcikdpx9o3i6        msc_mongodb1                replicated          1/1       [...]
u9b9ihxxnztn        msc_mongodb2                replicated          1/1       [...]
m0byoou6zuv5        msc_mongodb3                replicated          1/1       [...]
logqawe8k3cg        msc_platformservice         global              3/3       [...]
m3sxof6odn74        msc_schemaservice           global              3/3       [...]
3wd4zrqf6kbk        msc_siteservice             global              3/3       [...]
ourza0yho7ei        msc_syncengine              global              3/3       [...]
ojb8jkkrawqr        msc_ui                      global              3/3       [...]
zm94hzmzzelg        msc_userservice             global              3/3       [...]
```

### コンテナ ID の取得

`docker ps` コマンドを使用して、実行中のすべてのコンテナ ID のリストを取得できます。

```
# docker ps
CONTAINER ID    IMAGE                          COMMAND                  [...]
05f75d088dd1    msc-ui:2.1.2g                  "/nginx.sh"              [...]
0ec142fc639e    msc-authyldap:v.4.0.6          "/app/authyldap.bin"     [...]
b08d78533b3b    msc-cloudsecservice:2.1.2g     "bin/cloudsecservice"    [...]
685f54b70a0d    msc-executionengine:2.1.2g     "bin/executionengine"    [...]
0c719107adce    msc-schemaservice:2.1.2g       "bin/schemaservice"      [...]
f2e3d144738c    msc-userservice:2.1.2g         "bin/userservice"        [...]
edd0d4604e27    msc-syncengine:2.1.2g          "bin/syncengine"         [...]
001616674a00    msc-siteservice:2.1.2g         "bin/siteservice"        [...]
7b30c61f8aa7    msc-platformservice:2.1.2g     "bin/platformservice"    [...]
d02923992d77    msc-backupservice:2.1.2g       "bin/backupservice"      [...]
9de72d291aaa    msc-kong:2.1.2g                "/docker-entrypoint.…"   [...]
6135f9de5dd2    msc-mongo:3.6                  "sh -c 'sleep 3 && e…"   [...]
```

`docker ps | grep <service-name>` コマンドを使用して、特定のサービスの実行中のコンテナ ID を取得できます。

```
# docker ps | grep executionengine
685f54b70a0d    msc-executionengine:2.1.2g   "bin/executionengine"    [...]
```

終了したものを含むサービスのすべてのコンテナ ID を取得するには、`docker ps -a | grep <service-name>` コマンドを使用できます。

```
# docker ps -a | grep executionengine
685f54b70a0d   msc-executionengine:2.1.2g   "bin/executionengine"    Up 2 weeks (healthy)
3870d8031491    msc-executionengine:2.1.2g   "bin/executionengine"    Exited (143) 2
weeks ago
```

### コンテナ ログの表示

`docker logs <container-id>` コマンドを使用して、コンテナのログを表示します。転送するファイルが多くコンテナのログが大きくなる可能性があるため、コマンドを実行するときはネットワーク速度を考慮してください。

コンテナのログ ファイルのサンプルの場所は、`/var/lib/docker/containers/<container>` です。複数の `<container>-json.log` ファイルが存在する場合があります。

```
# cd /var/lib/docker/containers
# ls -al
total 140
drwx------. 47 root root 4096 Jul  9 14:25 .
drwx--x--x. 14 root root 4096 May  7 08:31 ..
drwx------.  4 root root 4096 Jun 24 09:58
051cf8e374dd9a3a550ba07a2145b92c6065eb1071060abee12743c579e5472e
drwx------.  4 root root 4096 Jul 11 12:20
0eb27524421c2ca0934cec67feb52c53c0e7ec19232fe9c096e9f8de37221ac3
[...]
# cd 051cf8e374dd9a3a550ba07a2145b92c6065eb1071060abee12743c579e5472e/
# ls -al
total 48
drwx------.  4 root root 4096 Jun 24 09:58 .
drwx------. 47 root root 4096 Jul  9 14:25 ..
-rw-r-----.  1 root root 4572 Jun 24 09:58
051cf8e374dd9a3a550ba07a2145b92c6065eb1071060abee12743c579e5472e-json.log
drwx------.  2 root root    6 Jun 24 09:58 checkpoints
-rw-------.  1 root root 4324 Jun 24 09:58 config.v2.json
-rw-r--r--.  1 root root 1200 Jun 24 09:58 hostconfig.json
-rw-r--r--.  1 root root   13 Jun 24 09:58 hostname
-rw-r--r--.  1 root root  173 Jun 24 09:58 hosts
drwx------.  3 root root   16 Jun 24 09:58 mounts
-rw-r--r--.  1 root root   38 Jun 24 09:58 resolv.conf
-rw-r--r--.  1 root root   71 Jun 24 09:58 resolv.conf.hash
```

### Docker ネットワークの表示

`docker network list` コマンドを使用して、Docker が使用するネットワークのリストを表示できます。

```
# docker network list
NETWORK ID          NAME                DRIVER              SCOPE
c0ab476dfb0a        bridge              bridge              local
79f5e2d63623        docker_gwbridge     bridge              local
dee475371fcb        host                host                local
99t2hdts7et0        ingress             overlay             swarm
588qhaj3mrj1        msc_msc             overlay             swarm
a68901087366        none                null                local
```

# API コール ログの生成

マルチサイト Orchestrator API コール ログには、トラブルシューティング レポートのインフラ ログからアクセスできます。トラブルシューティングの生成に関する詳細については、システム ログのダウンロード （12 ページ） を参照してください。

次の手順で API コール ログ マルチサイト にアクセスすることもできます。

**ステップ 1** 次の例のように、`msc-executionengine` サービスが実行されているワーカー ノードを見つけます。

**例**：

```
[root@worker1 ~]# docker ps
CONTAINER ID  IMAGE                         COMMAND                  CREATED      STATUS
   PORTS      NAMES
1538a9289381  msc-kong:latest               "/docker-entrypoin..."  2 weeks ago  Up 2 weeks
   7946/tcp,  msc_kong.1.ksdw45p0qhb6c08i3c8i4ketc

   8000-8001/tcp, 8443/tcp
cc693965f502  msc-executionengine:latest  "bin/executionengine"   2 weeks ago  Up 2 weeks (healthy)
  9030/tcp   msc_executionengine.1.nv4j5uj5786yj621wjxsxvgxl
00f627c6804c  msc-platformservice:latest  "bin/platformservice"   2 weeks ago  Up 2 weeks (healthy)
  9050/tcp   msc_platformservice.1.fw58jr62dfcme4noh67am0s73
```

この場合、`cc693965f502` のイメージは `msc-executionengine:latest` で、マルチサイトから APIC コントローラへの API コールを含む `-json.log` を見つけます。

**ステップ 2** 次の例にコマンドを入力します。

**例**：

```
# cd /var/lib/docker/containers/cc693965f5027f291d3af4a6f2706b19f4ccdf6610de3f7ccd32e1139e31e712
# ls
cc693965f5027f291d3af4a6f2706b19f4ccdf6610de3f7ccd32e1139e31e712-json.log checkpoints config.v2.json
 hostconfig.json hostname
hosts resolv.conf resolv.conf.hash shm

# less \
cc693965f5027f291d3af4a6f2706b19f4ccdf6610de3f7ccd32e1139e31e712-json.log | grep intersite
{"log":" \u003cfvBD name=\"internal\" arpFlood=\"yes\" intersiteBumTrafficAllow=\"yes\"
unkMacUcastAct=\"proxy\"
intersiteL2Stretch=\"yes\"\u003e\n","stream":"stdout","time":"2017-07-25T08:41:51.241428676Z"}
{"log":" \"intersiteBumTrafficAllow\" :
true,\n","stream":"stdout","time":"2017-07-27T07:17:55.418934202Z"}
{"log":" \"intersiteBumTrafficAllow\" :
true,\n","stream":"stdout","time":"2017-07-29T10:46:15.077426434Z"}
{"log":" \u003cfvBD name=\"internal\" arpFlood=\"yes\" intersiteBumTrafficAllow=\"yes\"
unkMacUcastAct=\"proxy\"
intersiteL2Stretch=\"yes\"\u003e\n","stream":"stdout","time":"2017-07-29T10:46:15.334099333Z"}
{"log":" \"intersiteBumTrafficAllow\" :
true,\n","stream":"stdout","time":"2017-07-29T11:57:09.361401249Z"}
{"log":" \"intersiteBumTrafficAllow\" :
true,\n","stream":"stdout","time":"2017-07-29T11:58:05.491624285Z"}
{"log":" \u003cfvBD name=\"internal\" arpFlood=\"yes\" intersiteBumTrafficAllow=\"yes\"
unkMacUcastAct=\"flood\"
intersiteL2Stretch=\"yes\"\u003e\n","stream":"stdout","time":"2017-07-29T11:58:05.673341176Z"}
{"log":" \u003cfvBD name=\"internal\" arpFlood=\"yes\" intersiteBumTrafficAllow=\"yes\"
unkMacUcastAct=\"flood\"
```

```
intersiteL2Stretch=\"yes\"\u003e\n","stream":"stdout","time":"2017-07-29T11:58:05.680167766Z"}
{"log":" \"intersiteBumTrafficAllow\" :
true,\n","stream":"stdout","time":"2017-07-29T11:58:44.826160838Z"}
{"log":" \u003cfvBD name=\"internal\" arpFlood=\"yes\" intersiteBumTrafficAllow=\"yes\"
unkMacUcastAct=\"proxy\"
intersiteL2Stretch=\"yes\"\u003e\n","stream":"stdout","time":"2017-07-29T11:58:45.008739316Z"}
{"log":" \u003cfvBD name=\"internal\" arpFlood=\"yes\" intersiteBumTrafficAllow=\"yes\"
unkMacUcastAct=\"proxy\"
intersiteL2Stretch=\"yes\"\u003e\n","stream":"stdout","time":"2017-07-29T11:58:45.008812862Z"}
```


# 実行ログの読み取り

実行ログは、3 種類のログ情報を提供します。

- 5 分ごとに出力される Websocket 更新情報。

```
2017-07-11 18:02:45,541 [debug] execution.serice.monitor.WSAPicActor - WebSocket
connection open
2017-07-11 18:02:45,542 [debug] execution.serice.monitor.WSAPicActor - Client 3
intialized
2017-07-11 18:02:45,551 [debug] execution.serice.monitor.WSAPicActor - WSAPicActor
stashing message Monitor Policy(WSMonitorQuery(/api/class/fvRsNodeAtt,?subscript
2017-07-11 18:02:45,551 [debug] execution.serice.monitor.WSAPicActor - WSAPicActor
stashing message RefreshClientTokenFailed()
2017-07-11 18:02:45,551 [debug] execution.serice.monitor.WSAPicActor - WSAPicActor
stashing message RefreshClientToken()
2017-07-11 18:02:45,551 [debug] execution.serice.monitor.WSAPicActor - WSAPicActor
stashing message RefreshClientToken()
2017-07-11 18:02:50,042 [debug] execution.serice.monitor.WSAPicActor - Websocket
connection open
2017-07-11 18:02:50,042 [debug] execution.serice.monitor.WSAPicActor - Client 3
intialized
2017-07-11 18:02:50,043 [debug] execution.serice.monitor.WSAPicActor - Initiate WS
subscription for WSMonitorQuery(/api/class/fvRsNodeAtt,?subscript-yes&page-s
2017-07-11 18:02:50,047 [debug] execution.serice.monitor.WSAPicActor - WSAPicActor
stashing message RefreshClientToken()
2017-07-11 18:02:50,047 [debug] execution.serice.monitor.WSAPicActor - WSAPicActor
stashing message RefreshClientToken()
2017-07-11 18:02:50,180 [debug] execution.serice.monitor.WSAPicActor - WSAPicActor
stashing message akka.actor.LightArrayRevolerScheduler$TaskHolder@13d740ff
2017-07-11 18:02:55,221 [debug] execution.serice.monitor.WSAPicActor - Websocket
connection open
2017-07-11 18:02:55,222 [debug] execution.serice.monitor.WSAPicActor - Client 3
intialized
2017-07-11 18:02:55,233 [debug] execution.serice.monitor.WSAPicActor - Token Refreshed
2017-07-11 18:02:55,323 [debug] execution.serice.monitor.WSAPicActor - Token Refreshed
```

- プッシュするスキーマと生成されるプラン。
- クロス VNID プログラミングのための Websocket モニタリング VNID。

次のエラーの兆候に注意してください。

- 赤いエラーで始まるログ行。
- 例外のスタックトレース。

# APIC サイトでのポリシー解決の確認

このタスクでは、ローカル APIC サイトまたはスイッチで REST API MO クエリを使用して、Cisco ACI マルチサイト 管理対象サイトの APIC で解決されたポリシーを表示します。

管理対象オブジェクト（MO）の関係の図については、『*Cisco APIC* 管理情報モデルリファレンス（*MIM*）』を参照してください。たとえばMIM では、`fv:FabricExtConnP` の図を参照してください。

**ステップ 1** ファブリック外部接続プロファイル（fabricExtConnP）の下の論理 MO の詳細を表示するには、APIC CLI にログオンして、次の MO クエリを入力します。

**例**：

```
admin@apic1:~> moquery -c fvFabricExtConnP -x "query-target=subtree"
| egrep "#|dn"
# fv.IntersiteMcastConnP
dn: uni/tn-infra/fabricExtConnP-1/intersiteMcastConnP
# fv.IntersitePeeringP
dn: uni/tn-infra/fabricExtConnP-1/ispeeringP
# fv.IntersiteConnP
dn: uni/tn-infra/fabricExtConnP-1/podConnP-1/intersiteConnP-[5.5.5.1/32]
# fv.Ip
dn: uni/tn-infra/fabricExtConnP-1/podConnP-1/ip-[5.5.5.4/32]
# fv.PodConnP
dn: uni/tn-infra/fabricExtConnP-1/podConnP-1
# fv.IntersiteConnP
dn: uni/tn-infra/fabricExtConnP-1/siteConnP-6/intersiteConnP-[6.6.6.1/32]
# fv.IntersiteMcastConnP
dn : uni/tn-infra/fabricExtConnP-1/siteConnP-6/intersiteMcastConnP
# fv.SiteConnP
dn: uni/tn-infra/fabricExtConnP-1/siteConnP-6
# l3ext.FabricExtRoutingP
dn: uni/tn-infra/fabricExtConnP-1/fabricExtRoutingP-default
# fv.FabricExtConnP
dn: uni/tn-infra/fabricExtConnP-1
```

**ステップ 2** マルチサイト接続に使用される L3Out の論理 MO を表示するには、APIC CLI にログオンして、次のような MO クエリを入力します。

**例**：

```
admin@apic1:~> moquery -c l3extOut -x "query-target=subtree" | egrep
"#|dn.*intersite" | grep -B 1 dn
# bgp.ExtP
dn: uni/tn-infra/out-intersite/bgpExtP
# fv.RsCustQosPol
dn: uni/tn-infra/out-intersite/instP-intersiteInstP/rscustQosPol
# l3ext.InstP
dn: uni/tn-infra/out-intersite/instP-intersiteInstP
# bgp.AsP
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/infraPeerP-[6.6.6.3]/as
# bgp.RsPeerPfxPol
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/infraPeerP-[6.6.6.3]/rspeerPfxPol
# bgp.InfraPeerP
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/infraPeerP-[6.6.6.3]
# l3ext.RsEgressQosDppPol
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/lifp-port-1-1/rsegressQosDppPol
```

```
# l3ext.RsIngressQosDppPol
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/lifp-port-1-1/rsingressQosDppPol
# l3ext.RsNdIfPol
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/lifp-port-1-1/rsNdIfPol
# l3ext.RsPathL3OutAtt
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/lifp-port-1-1/rspathL3OutAtt-
[topology/pod-1/paths-501/pathep-[eth1/1]]
# ospf.RsIfPol
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/lifp-port-1-1/ospfIfP/rsIfPol
# ospf.IfP
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/lifp-port-1-1/ospfIfP
# l3ext.LIfP
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/lifp-port-1-1
# l3ext.InfraNodeP
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/rsnodeL3OutAtt-
[topology/pod-1/node-501]/infranodep
# l3ext.IntersiteLoopBackIfP
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/rsnodeL3OutAtt-
[topology/pod-1/node-501]/sitelbp-[5.5.5.3]
# l3ext.RsNodeL3OutAtt
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile/rsnodeL3OutAtt-
[topology/pod-1/node-501]
# l3ext.LNodeP
dn: uni/tn-infra/out-intersite/lnodep-node-501-profile
# l3ext.RsEctx
dn: uni/tn-infra/out-intersite/rsectx
# l3ext.RsL3DomAtt
dn: uni/tn-infra/out-intersite/rsl3DomAtt
# ospf.ExtP
dn: uni/tn-infra/out-intersite/ospfExtP
# l3ext.Out
dn: uni/tn-infra/out-intersite--
# l3ext.ConfigOutDef
dn: uni/tn-infra/out-intersite/instP-intersiteInstP/configOutDef
```

**ステップ 3** APIC ローカル サイトの解決された MO を表示するには、APIC CLI にログオンし、次のような MO クエリを入力します。

**例**：

```
admin@apic1:~> moquery -c fvSite -x "query-target=subtree" | egrep "#|dn"
# fv.RemoteBdDef
dn: resPolCont/sitecont/site-6/remotebddef-[uni/tn-msite-tenant-welkin/BD-internal]
# fv.RemoteCtxDef
dn: resPolCont/sitecont/site-6/remotectxdef-[uni/tn-msite-tenant-welkin/ctx-dev]
# fv.RemoteEPgDef
dn: resPolCont/sitecont/site-6/remoteepgdef-[uni/tn-msite-tenant-welkin/ap-Ebiz/epg-data]
# fv.RemoteEPgDef
dn: resPolCont/sitecont/site-6/remoteepgdef-[uni/tn-msite-tenant-welkin/ap-Ebiz/epg-web]
# fv.Site
dn: resPolCont/sitecont/site-6
# fv.LocalBdDef
dn: resPolCont/sitecont/site-5/localbddef-[uni/tn-msite-tenant-welkin/BD-internal]
# fv.LocalCtxDef
dn: resPolCont/sitecont/site-5/localctxdef-[uni/tn-msite-tenant-welkin/ctx-dev]
# fv.LocalEPgDef
dn: resPolCont/sitecont/site-5/localepgdef-[uni/tn-msite-tenant-welkin/ap-Ebiz/epg-web]
# fv.LocalEPgDef
dn: resPolCont/sitecont/site-5/localepgdef-[uni/tn-msite-tenant-welkin/ap-Ebiz/epg-data]
# fv.Site
dn: resPolCont/sitecont/site-5
```

**ステップ 4** マルチサイト サイトのスイッチの具体的な MO を表示するには、スイッチにログオンし、次のような MO クエリを入力します。

**例**：

```
spine501# moquery -c dci.LocalSite -x "query-target=subtree" | egrep "#|dn"
# l2.RtToLocalBdSubstitute       //(site5 vrf 2195456 -> bd 15794150 is translated to
site6 vrf 2326528 -> bd 16449430)
dn: sys/inst-overlay-1/localSite-5/localCtxSubstitute-[vxlan-2195456]/localBdSubstitute-
[vxlan-15794150]/rttoLocalBdSubstitute-[sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-
[vxlan-2326528]/remoteBdSubstitute-[vxlan-16449430]]
# l2.LocalBdSubstitute
dn: sys/inst-overlay-1/localSite-5/localCtxSubstitute-[vxlan-2195456]/localBdSubstitute-
[vxlan-15794150]
# l2.RtToLocalPcTagSubstitute    //(site5 vrf 2195456 -> pcTag 49154 is translated to
site6 vrf 2326528 - > pcTag 32770)
dn: sys/inst-overlay-1/localSite-5/localCtxSubstitute-[vxlan-2195456]/localPcTagSubstitute-
49154/rttoLocalPcTagSubstitute-[sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-
[vxlan-2326528]/remotePcTagSubstitute-32770]
# l2.LocalPcTagSubstitute
dn: sys/inst-overlay-1/localSite-5/localCtxSubstitute-[vxlan-2195456]/localPcTagSubstitute-
49154# l2.RtToLocalPcTagSubstitute    //(site5 vrf 2195456 -> pcTag 16387 is translated to site6
vrf 2326528 - > pcTag 16386)
dn: sys/inst-overlay-1/localSite-5/localCtxSubstitute-[vxlan-2195456]/localPcTagSubstitute-
16387/rttoLocalPcTagSubstitute-[sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-
[vxlan-2326528]/remotePcTagSubstitute-16386]
# l2.LocalPcTagSubstitute
dn: sys/inst-overlay-1/localSite-5/localCtxSubstitute-[vxlan-2195456]/localPcTagSubstitute-
16387# l3.RtToLocalCtxSubstitute      //(site5 vrf 2195456  is translated to site6 vrf 2326528)
dn: sys/inst-overlay-1/localSite-5/localCtxSubstitute-[vxlan-2195456]/rttoLocalCtxSubstitute-
[sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]]
# l3.LocalCtxSubstitute
dn: sys/inst-overlay-1/localSite-5/localCtxSubstitute-[vxlan-2195456]
# dci.LocalSite
dn: sys/inst-overlay-1/localSite-5
```

**確認事項**：出力には、サイト間で変換されたデータが表示されます。この例では、サイトの元のデータは次のとおりです。

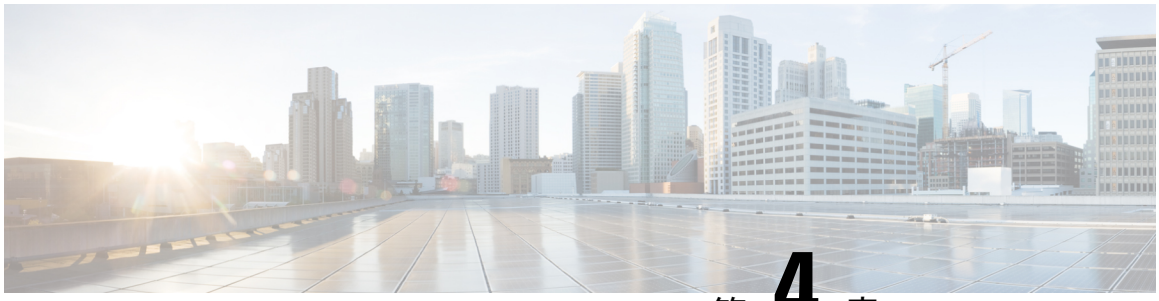
- site5 vrf msite-tenant-welkin:dev -> vxlan 2195456, bd internal -> vxlan 15794150, epg web: access-encap 200 → pcTag 49154, access-encap 201 → pcTag 16387
- site6 vrf msite-tenant-welkin:dev -> vxlan 2326528, bd internal -> vxlan 16449430, epg web: access-encap 200 ->pcTag 32770,access-encap 201 ->pcTag 16386

**ステップ 5** リモート サイトの具体的な MO を確認するには、次のような MO クエリを入力します。

**例**：

```
spine501# moquery -c dci.RemoteSite -x "query-target=subtree"
| egrep "#|dn"
# dci.AnycastExtn
dn: sys/inst-overlay-1/remoteSite-6/anycastExtn-[6.6.6.1/32]
// attribute is_unicast is Yes, Unicast ETEP
# dci.AnycastExtn
dn: sys/inst-overlay-1/remoteSite-6/anycastExtn-[6.6.6.2/32]
// attribute is_unicast is No, Multicast ETEP
# l2.RsToLocalBdSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remoteBdSubstitute-
[vxlan-16449430]/rsToLocalBdSubstitute
# l2.RemoteBdSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remoteBdSubstitute-
```

```
[vxlan-16449430]
# l2.RsToLocalPcTagSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remotePcTagSubstitute-
32770/rsToLocalPcTagSubstitute
# l2.RemotePcTagSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remotePcTagSubstitute-
32770# l2.RsToLocalPcTagSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remotePcTagSubstitute-
16386/rsToLocalPcTagSubstitute
# l2.RemotePcTagSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remotePcTagSubstitute-
16386# l3.RsToLocalCtxSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/rsToLocalCtxSubstitute
# l3.RemoteCtxSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]
# dci.RemoteSite
dn: sys/inst-overlay-1/remoteSite-6
```



第 4 章

# インストール、アップグレード、リブートのトラブルシューティング

この章は、次の項で構成されています。



## Orchestrator VM の CPU サイクル予約の増加

Cisco ACI Multi-Site Orchestrator VM には、一定量の専用 CPU サイクルが必要です。新しい展開では CPU サイクルの予約が自動的に適用されますが、リリース 2.1(1) より前のリリースから Orchestrator をアップグレードする場合は、各 Orchestrator VM の設定を手動で更新する必要があります。

適切な CPU サイクル予約を構成すると、次のようなランダムに見える多くの問題を解決または防止できます。

- ロードに 1 回以上の再試行が必要な Orchestrator GUI アイテム。
- 1 つまたは複数のノードが [不明（Unknown）] ステータスに変化し、後でそれ自体が [準備完了（Ready）] に解決されます。

```
# docker node ls
ID                            HOSTNAME    STATUS      AVAILABILITY    MANAGER STATUS
  ENGINE VERSION
```

```
t8wl1zoke0vpxdl9fysqu9otb     node1       Ready       Active          Reachable
   18.03.0-ce
kyriihdfhy1k1tlggan6e1ahs *   node2       Unknown     Active          Reachable
   18.03.0-ce
yburwactxd86dorindmx8b4y1     node3       Ready       Active          Leader
   18.03.0-ce
```

- 次のログ エントリの例では、Orchestrator ログ（`/var/log/messages` にあります）で一時的なハートビートの欠落が発生します。

```
node2 dockerd: [...] level=error msg="agent: session failed" backoff=100ms
    error="rpc error: code = Canceled desc = context canceled" module=node/agent
[...]
node2 dockerd: [...] level=error msg="heartbeat to manager [...] failed"
    error="rpc error: code = Canceled desc = context canceled" [...]
```

CPU サイクルの予約設定を更新するには、Orchestrator VM ごとに次の手順を繰り返します。

**ステップ 1** vSphere クライアントにログインします。

**ステップ 2** Orchestrator VM が配置されている ESX ホストに移動します。

**ステップ 3** VM をシャットダウンします。

**ステップ 4** VM を右クリックし、**[設定の編集（Edit Settings）]** を選択します。

**ステップ 5** **[仮想ハードウェア（Virtual Hardware）]** タブで、**CPU** カテゴリを展開します。

**ステップ 6** **[予約（Reservation）]** フィールドに、`10 GHz` と入力します。

**ステップ 7** [OK] をクリックして変更を保存します。

**ステップ 8** VM の電源を入れ、すべてのノードが正常な状態で Orchestrator クラスタが安定するのを待ちます。

# Orchestrator ノードの NTP の有効化

Orchestrator ノードにクロック同期が構成されていないと、認証トークンの有効期限切れによるランダムな GUI セッション ログオフなどの問題が発生する可能性があります。

通常、Multi-Site Orchestrator のインストール中に、Orchestrator ノードの Network Time Protocol（NTP）サーバの詳細を指定します。ただし、何らかの理由で NTP 設定を指定していない場合は、次の手順を使用して設定できます。

**ステップ 1** Orchestrator VM に直接ログインします。

**ステップ 2** スクリプト ディレクトリに変更します。

```
# cd /opt/cisco/msc/scripts
```

**ステップ 3** ノードに NTP 設定を構成します。

次のコマンド：

- '**-tz** *<time-zone>*' では、今いるタイムゾーンを指定します

- '**-ne** ' は NTP を有効にします
- '**-ns** *<ntp-server>*' は NTP サーバを指定します

```
# ./svm-msc-tz-ntp  -tz <time-zone> -ne -ns <ntp-server>
```

次に例を示します。

```
# ./svm-msc-tz-ntp -tz US/Pacific -ne -ns ntp.esl.cisco.com
svm-msc-tz-ntp: Start
svm-msc-tz-ntp: Executing timedatectl set-timezone US/Pacific
svm-msc-tz-ntp: Executing sed -i 's|^server|\# server|' /etc/ntp.conf
svm-msc-tz-ntp: Executing timedatectl set-ntp true
svm-msc-tz-ntp: Sleeping 10 seconds
svm-msc-tz-ntp: Checking NTP status
svm-msc-tz-ntp: Executing ntpstat;ntpq -p
unsynchronised
   polling server every 64 s
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
mtv5-ai27-dcm10 .GNSS.           1 u    -   64    1    1.581   -0.002   0.030
```

**ステップ 4** NTP 構成を確認します。

次のコマンドを使用して、NTP が有効になっていることを確認できます。

```
# ntpstat;ntpq -p
unsynchronised
   polling server every 64 s
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*mtv5-ai27-dcm10 .GNSS.           1 u   14   64    1    3.522   -0.140   0.128
```

正しい日付と時刻が設定されていることも確認できます。

```
# date
Mon Jul  8 14:19:26 PDT 2019
```

**ステップ 5** 各ノードでこの手順を繰り返します。

# DNS の更新

このセクションでは、Multi-Site Orchestrator クラスタの DNS サーバ アドレスを更新する方法について説明します。この手順は、VMware ESX の MSO OVA 展開にのみ適用され、アプリケーション サービス エンジンまたは Nexus ダッシュボードの展開には適用されないことに注意してください。

**ステップ 1** `root` ユーザーとしてクラスタ ノードの 1 つに SSH で接続します。

**ステップ 2** DNS 構成を更新します。

`nmcli` コマンドを使用して、DNS サーバの IP アドレスを更新します。

```
# nmcli connection modify eth0 ipv4.dns "<dns-server-ip>"
```

複数の DNS サーバの IP を指定する場合は、スペースで区切ったリストを使用します。

```
# nmcli connection modify eth0 ipv4.dns "<dns-server-ip-1> <dns-server-ip-2>"
```

**ステップ 3** 更新したネットワーク インターフェイスを再起動します。

変更を適用するには、eth0 インターフェイスを再起動する必要があります。

```
# nmcli connection down eth0 && nmcli connection up eth0
```

**ステップ 4** ノードをリブートします。

**ステップ 5** 他の 2 つのノードについても前の手順を繰り返します。

# 一時的にダウンした場合クラスタの単一ノードを再起動する

このセクションでは、クラスタの1つのノードが一時的にダウンした場合に再起動する方法について説明します。

ダウンしたノードを再起動します。追加の手順は必要なく、クラスタは自動的に回復します。

# 一時的にダウンしているクラスタの 2 つのノードを再起動する

このセクションでは、一時的にダウンしたクラスタの2つのノードを再起動する方法について説明します。

**ステップ 1** 現時点では、Docker スウォームに 3 つのマネージャー ノードのクォーラムがないため、マルチサイト は利用できません。リカバリを試みる前に、MongoDB をバックアップすることをお勧めします。

詳細については、MongoDB のバックアップ Cisco ACI マルチサイト （26 ページ）を参照してください。

**ステップ 2** ダウンしていた 2 つのノードを再起動します。追加で必要な手順はありません。クラスタが自己回復します。

# MongoDB のバックアップ Cisco ACI マルチサイト

このセクションで説明するように、CiscoはCisco ACIマルチサイトOrchestratorのアップグレードまたはダウングレードの前にMongoDBをバックアップすることを推奨します。

（注） データベースをバックアップできるのは、Orchestrator クラスタが稼働している場合のみです。これには、少なくとも 2 つのノードが稼働している必要があります。

**ステップ 1** Cisco ACI マルチサイト Orchestrator 仮想マシン（VM）にログインします。

**ステップ 2** Cisco ACI マルチサイト Orchestrator バックアップ スクリプトを実行します。

```
# ~/msc_scripts/msc_db_backup.sh
```

msc_backup_<*date+%Y%m%d%H%M*>.archive ファイルが作成されます。

**ステップ 3** msc_backup_<*date+%Y%m%d%H%M*>.archive ファイルを安全な場所にコピーします。

# Cisco ACI マルチサイト 向け MongoDB の復元

このセクションでは、Cisco ACI マルチサイト 向け MongoDB を復元する方法について説明します。

**ステップ 1** マルチサイト 仮想マシン（VM）にログインします。

**ステップ 2** msc_backup_<*date+%Y%m%d%H%M*>.archive ファイルを VM にコピーします。

**ステップ 3** マルチサイト DB 復元スクリプトを実行します。

```
# ~/msc_scripts/msc_db_restore.sh
```

**ステップ 4** Python スクリプトを実行して、スキーマを再度プッシュします。

```
# msc_push_schemas.py
```

# カスタム証明書のトラブルシューティング

ここでは、マルチサイト Orchestrator でカスタム SSL 証明書を使用する場合の一般的な問題を解決する方法について説明します。

## Orchestrator GUI をロードできません

カスタム証明書をインストールしてアクティブ化した後に Orchestrator GUI ページをロードできない場合は、各 Orchestrator ノードに証明書が正しくコピーされていない可能性があります。この問題を解決するには、デフォルトの証明書を回復してから、新しい証明書のインストール手順を再度繰り返します。

デフォルトの Orchestrator 証明書を回復するには、次のようにします。

**ステップ 1** 各 Orchestrator ノードに直接ログインします。

**ステップ 2** 証明書ディレクトリに移動します。

```
# cd /data/msc/secrets
```

**ステップ 3** msc.key および msc.crt ファイルを、msc.key_backup および msc.crt_backup ファイルに個別に置き換えます。

msc.key および msc.crt ファイルを、msc.key_backup および msc.crt_backup ファイルに個別に置き換えます。

**ステップ 4** Orchestrator GUI サービスを再起動します。

```
# docker service update msc_ui --force
```

**ステップ 5** 前のセクションで説明したように、新しい証明書を再インストールしてアクティブにします。

# クラスタへの新しい Orchestrator ノードの追加

マルチサイト Orchestrator クラスタに新しいノードを追加する場合は、次のようにします。

**ステップ 1** Orchestrator GUI にログインします。

**ステップ 2** 前のセクションで説明したように、使用しているキーを再度アクティブにします。

# デフォルトのキーリングの有効期限が切れた後に新しいキーリングをインストールできない

デフォルトのキーリングの有効期限が切れた後に新しいキーリングをインストールできない場合は、カスタムキーリングがクラスタノードにインストールされていない可能性があります。

この問題を解決するには、以下の手順を使用して、古いデフォルトのキーリングを削除し、新しいキーリングを作成します。

**ステップ 1** クラスタのすべてのノードで次のコマンドを実行します。

```
cd /data/msc/secrets
rm -rf /data/msc/secrets/msc.key
rm -rf /data/msc/secrets/msc.crt
rm -rf /data/msc/secrets/msc.key_backup
rm -rf /data/msc/secrets/msc.crt_backup
!
!
openssl req -newkey rsa:2048 -nodes -keyout /data/msc/secrets/msc.key -x509 -days 365 -out
```

```
/data/msc/secrets/msc.crt -subj '/CN=MSC'
cp /data/msc/secrets/msc.key /data/msc/secrets/msc.key_backup
cp /data/msc/secrets/msc.crt /data/msc/secrets/msc.crt_backup
cd /data/msc/secrets
chmod 777 msc.key
chmod 777 msc.key_backup
chmod 777 msc.crt
chmod 777 msc.crt_backup
```

**ステップ 2** 次のコマンドを実行して、msc_ui サービスの更新を強制します。

```
# docker service update msc_ui --force
```

**ステップ 3** 更新が完了したら、msc_ui のすべての複製が正常かどうかを確認します。

```
[root@node1 ~]# docker service ls
...
rqs0607lgixg     msc_ui     global     3/3    msc-ui:3.1.1i
*:443->443/tcp
```

**ステップ 4** 任意のブラウザを使用して、任意の MSO ノードにログインします。証明書の詳細を受け入れるときにブラウザがループでスタックするか、白い画面が表示される場合は、GUI が再び正常に表示されるまでページを 1 ～ 2 回更新します。

**ステップ 5** ユーザー名とパスワードを使用してログインし、「カスタム キーリングのアクティブ化」セクションで説明されている手順に従ってキーリングをアクティブ化します。

# クラスタの単一ノードを新しいノードに置き換える

このセクションでは、クラスタの単一ノードを新しいノードに置き換える方法について説明します。

このシナリオでは、ノード 1 がダウンし、ノード 1 を新しいノードに置き換える必要があります。

**ステップ 1** 既存のノードで、ダウンしているノード (node1) の ID を取得します。次のコマンドを実行します。

```
root@node2 ~]# docker node ls
ID                           HOSTNAME  STATUS  AVAILABILITY   MANAGER STATUS
11624powztg5tl9nlfoubydtp *  node2     Ready   Active         Leader
fsrca74nl7byt5jcv93ndebco    node3     Ready   Active         Reachable
wnfs9oc687vuusbzd3o7idllw    node1     Down    Active         Unreachable
```

**ステップ 2** node1 を降格し、次のコマンドを実行する必要があります。

```
[root@node2 ~]# docker node demote <node ID>
Manager <node ID> demoted in the swarm.
```

<*node ID*> は、手順 1 でノード ID を受け取った場所です。

**ステップ 3** 新しいノードを追加する前にダウンしている node1 を削除し、次のコマンドを実行します。

```
[root@node2 ~]# docker node rm <node ID>
```

**ステップ 4** 既存のノードで、/opt/cisco/msc/builds/<*build_number*>/prodhaディレクトリに変更します。

例：

```
# cd /opt/cisco/msc/builds/<build_number>/prodha
```

**ステップ 5** トークンをメモします。既存のノードで、次のコマンドを実行します。

```
[root@node1 prodha]# docker swarm join-token manager
docker swarm join --token
SWMTKN-1-4yaodn4nj8nek0qghh4dfzn6zm9o9p29rjisdikhjpvwu8bgmw-0ig2g62e0fe62cq2hbexk6xgv \
1.1.1.1:2376
```

**ステップ 6** 新しいリーダーの IP アドレスを書き留めます。既存のノードで、次のコマンドを入力します。

例：

```
[root@node1 prodha]# docker node ls
ID                           HOSTNAME   STATUS   AVAILABILITY   MANAGER STATUS
pjicie1wlcgkoef1x9s0td7ac    node1      Down     Active         Reachable
qy6peh6wtsbsaf9cpyh2wr5f6    node2      Ready    Active         Leader
tfhhvzt7qx9lxkqalbxfwknsq    node3      Ready    Active         Reachable
```

**ステップ 7** リーダー ノード（node2）で、IP アドレスをメモします。

```
# ifconfig
inet 10.23.230.152 netmask 255.255.255.0 broadcast 192.168.99.255
```

**ステップ 8** 新しい 3 番目のノードを準備します。新しいノードの正しいホスト名を設定します。

例：

```
# hostnamectl set-hostname <node name>
```

**ステップ 9** /opt/cisco/msc/builds/<ビルド番号>/prodha ディレクトリに変更します。

例：

```
# cd /opt/cisco/msc/builds/<build_number>/prodha
```

**ステップ 10** 新しいノードをスウォームに参加させます:

例：

```
[root@node1 prodha]# ./msc_cfg_join.py <token> <address of leader>
```

<*token*> は、手順 5 でトークン情報を受け取った場所です。

<*address of leader*> は、手順 7 でリーダーの IP アドレスを受け取った場所です。

**ステップ 11** いずれかのノードで、/opt/cisco/msc/builds/<*build_number*>/prodha ディレクトリに変更します。

例：

```
# cd /opt/cisco/msc/builds/<build_number>/prodha
```

**ステップ 12** いずれかのノードで、次のコマンドを実行します。

```
[root@node1 prodha]# ./msc_deploy.py
```

---

この時点で、すべてのサービスが稼働し、データベースが複製されているはずです。

# クラスタの 2 つの既存のノードを新しいノードに置き換える

このセクションでは、クラスタの2つの既存のノードを新しいノードに置き換える方法について説明します。ここでの手順は、クラスタに Docker スウォームを使用する ESX VMware VM での Orchestrator の展開に関するものです。

この時点では、Docker スウォームに 3 つのマネージャ ノードのクォーラムがないため、マルチサイト Orchestrator は使用できません。リカバリを試みる前に、DB をバックアップすることをお勧めします。詳細については、MongoDB のバックアップ Cisco ACI マルチサイト （26 ページ）を参照してください。

**ステップ 1** 2 つの新しいノードを起動し、新しいノードごとに適切なノード名を設定します。

新しいノードを起動したら、次のコマンドを使用してそのノード名を割り当てることができます。

```
# hostnamectl set-hostname <node-name>
```

3 つのノード名はすべてクラスタ内で一意である必要があることに注意してください。

**ステップ 2** 以前はスウォームの一部であった唯一のライブ ノードで、ダウンしている他のノードを削除します。

a) スウォームの一部である唯一のライブ ノードに SSH で接続します。

b) すべてのノードのステータスを表示します。

```
# docker node ls
ID                           HOSTNAME   STATUS   AVAILABILITY  MANAGER     STATUS
g3mebdulaed2n0cyywjrtum31    node2      Down     Active        Reachable
ucgd7mm2e2divnw9kvm4in7r7    node1      Ready    Active        Leader
zjt4dsodu3bff3ipn0dg5h3po *  node3      Down     Active        Reachable
```

c) **[ダウン（Down）]** ステータスのノードを削除します。

```
# docker node rm <node-id>
```

次に例を示します。

```
# docker node rm g3mebdulaed2n0cyywjrtum31
```

**ステップ 3** Docker スウォームを再起動します。

唯一のライブ ノードにログインしたままで、次の手順を実行します。

a) 既存のスウォームを残します。

```
# docker swarm leave --force
```

b) Orchestrator スクリプト ディレクトリに変更します。

```
# cd /opt/cisco/msc/builds/<build-number>/prodha
```

c) 新しいスウォームを再開します。

```
# ./msc_cfg_init.py
```

このコマンドは、2 つの新しいノードを新しいクラスタに参加させるために使用する必要があるトークンと IP アドレスを返します。

**ステップ 4** 2 つの新しいノードをクラスタに参加させます。

新しい各ノードで、次の手順を実行します。

a) ノードに SSH 接続します。

b) Orchestrator スクリプト ディレクトリに変更します。

```
# cd /opt/cisco/msc/builds/<build-number>/prodha
```

c) ノードをクラスタに参加させます。

以下のコマンド

- 前の手順で Docker スウォームを再起動したとき、<*token*> を **./msc_cfg_init.py** コマンドから受け取ったトークンに置き換えます。
- <*ip-address*> を前の手順でも受け取った最初のノードの IP アドレスに置換します。

```
# ./msc_cfg_join.py <token> <ip-address>
```

**ステップ 5** 新しい構成を展開します。

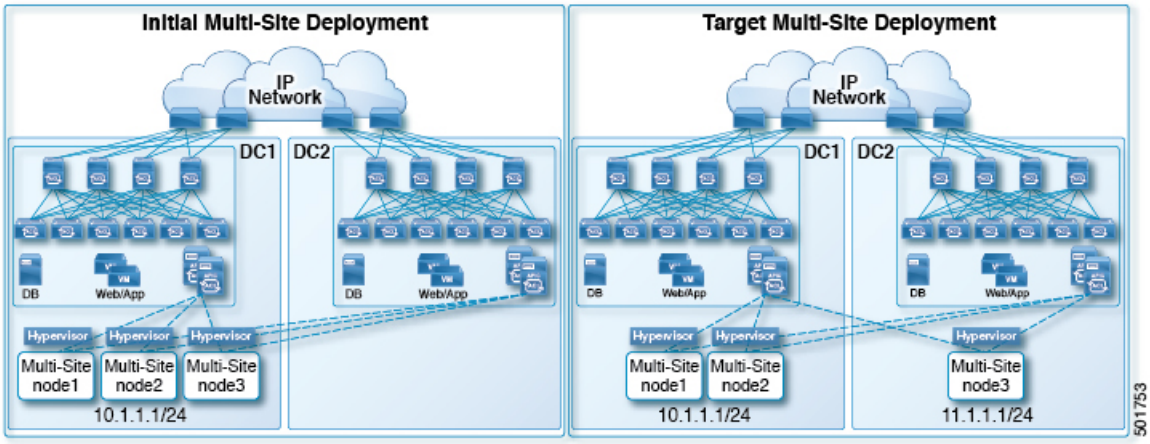
新しいくラスタのいずれかのノードから次のコマンドを実行できます。スクリプトは、同じ /opt/cisco/msc/builds/<*build-number*>/prodha スクリプト ディレクトリにあります。

```
# ./msc_deploy.py
```

# 別のサブネットへのノードの再配置マルチサイト

このセクションでは、1つ以上のマルチサイト ノードをあるサブネットから別のサブネットに再配置する方法について説明します。これは、マルチサイト が単一のデータ センター内に展開され、ノードを1つ以上のデータ センターに分散することが目標である場合の一般的なタスクです。移行中に冗長性を維持するために、一度に1つのノードを移動することが重要です。

図 *1 : Cisco ACI* マルチサイトの導入




以下の手順例は、管理サブネットが 10.1.1.1/24 サブネットを使用するデータセンター 1 から、管理サブネットが 11.1.1.1/24 サブネットを使用するデータセンター 2 への マルチサイト node3 の再配置を示しています。

**ステップ 1** node1 で、node3 を降格します。

**例**：

```
[root@node1 prodha]# docker node demote node3
```

**ステップ 2** node3 仮想マシン（VM）の電源を切ります。

**ステップ 3** クラスタから node3 を削除します。

**例**：

```
[root@node1 prodha]# docker node rm node3
```

**ステップ 4** 新しい マルチサイト VM（node1 および node2 と同じバージョン）をデータ センターに展開します。新しい IP の詳細を構成し、ホスト名「node3」が割り当てられていることを確認します。

**ステップ 5** データ センター 2 の node3 の電源を入れ、node1 と node2 への接続をテストします。

**例**：

```
[root@node3 prodha]# ping [node1_IP]
[root@node3 prodha]# ping [node2_IP]
```

**ステップ 6** node1 で、node1 から参加トークンを取得して、node3 をクラスタに参加させます。

**例**：

```
[root@node1 prodha]# docker swarm join-token manager
To add a manager to this swarm, run the following command:

docker swarm join --token \
SWMTKN-1-4plaanp2uqpkjm2nidsxg9u7it0dd8hkihwjq9wvrz5heykl2n-98eo0onpacvxrrgf84juczdve \
10.1.1.1:2377

[root@node1 prodha~]#
```

**ステップ 7** node3 で、手順 6 の参加トークンを使用してスウォームに参加します。

**例**：

```
[root@node3 prodha]# docker swarm join --token \
SWMTKN-1-4plaanp2uqpkjm2nidsxg9u7it0dd8hkihwjq9wvrz5heykl2n-98eo0onpacvxrrgf84juczdve \
10.1.1.1:2377
```

**ステップ 8** 任意のノードで、ノードが正常に稼働していることを確認します。各ノードについて、[STATUS] が Ready、[AVAILABILITY] が Active となっていて、[MANAGER STATUS] が 1 つのみ Leader と表示されることを除いて Reachable となっていることを確認します。

**例**：

```
[root@node1 ~]# docker node ls
ID                           HOSTNAME   STATUS   AVAILABILITY   MANAGER STATUS
p71zqw77kwnu8z6sr1w0uq2g0    node2      Ready    Active         Leader
q5orng9hd4f0vxneqeehixxwt    node3      Ready    Active         Reachable
ryag1u9ej33pfvrjvqgj4tjr4 *  node1      Ready    Active         Reachable
[root@node1 ~]#
```

**ステップ 9** node3 のスウォーム ラベルを更新します。

**例**：

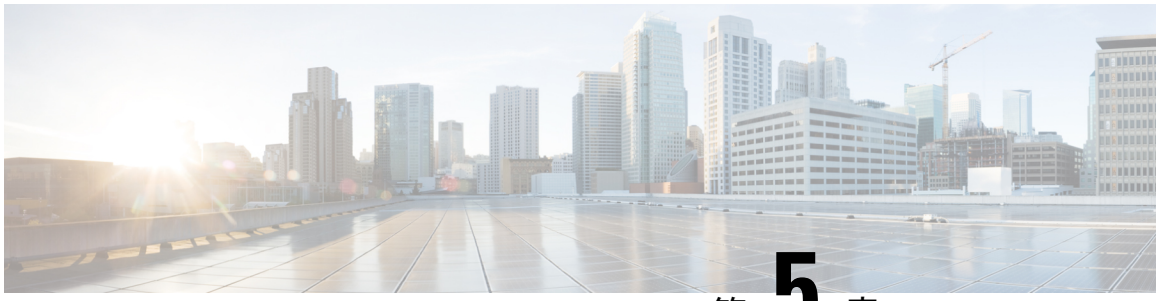
```
[root@node1 prodha]# docker node update node3 --label-add msc-node=msc-node3
```

**ステップ 10** 任意のノードで、すべての docker サービスのステータスを確認します。たとえば、1/1（1 のうち 1）または 3/3（3 のうち 3）と記載されていることを確認します。同期には最大 15 分かかる場合があります。

**例**：

```
[root@node1 ~]# docker service ls
ID             NAME                      MODE         REPLICAS   IMAGE
  PORTS
3kv2qtu3gjmk   msc_kongdb                replicated   1/1        msc-postgres:9.4
5fs0lg9bbbg1   msc_kong                  global       3/3        msc-kong:1.1
jrxade8o2nwn   msc_schemaservice         global       3/3        msc-schemaservice:1.2.0.206
kyq1myno38ry   msc_backupservice         global       3/3        msc-backupservice:1.2.0.206
ltx85gitz85u   msc_executionengine       replicated   1/1        msc-executionengine:1.2.0.206
n4skpiij90t1   msc_ui                    global       3/3        msc-ui:1.2.0.206
  *:80->80/tcp,*:443->443/tcp
o2h8vp3clznd   msc_mongodb1              replicated   1/1        msc-mongo:3.4
q2udphffzb7g   msc_consistencyservice    replicated   1/1        msc-consistencyservice:1.2.0.206
qr1zbd0y18u1   msc_platformservice       global       3/3        msc-platformservice:1.2.0.206
rsb7ki0zxafa   msc_mongodb2              replicated   1/1        msc-mongo:3.4
uiu25mz5h7m9   msc_userservice           global       3/3        msc-userservice:1.2.0.206
xjrp2jbws4pz   msc_auditservice          replicated   1/1        msc-auditservice:1.2.0.206
xtsdns1iy52i   msc_syncengine            replicated   1/1        msc-syncengine:1.2.0.206
ypie99rvielj   msc_mongodb3              replicated   1/1        msc-mongo:3.4
zn03gxpleu1s   msc_siteservice           global       3/3        msc-siteservice:1.2.0.206
[root@node1 ~]#
```

**ステップ 11** データセンター 1 で電源を切った元の node3 VM を削除します。



第 5 章

# ユーザーのトラブルシューティング

この章は、次の内容で構成されています。



## ローカル管理パスワードのリセット

このセクションでは、Multi-Site Orchestrator クラスタのローカル管理者パスワードをリセットする方法について説明します。この手順は、VMware ESX の MSO OVA 展開にのみ適用され、アプリケーション サービス エンジンまたは Nexus ダッシュボードの展開には適用されないことに注意してください。

**ステップ 1** root ユーザーとしていずれかのクラスタ ノードに SSH で接続します。

**ステップ 2** 管理者のログイン情報を削除します。

次のスクリプトを使用して、管理者のログイン情報を削除します。

```
# cd /opt/cisco/msc/builds/<build_version>/bin
# ./msc_delete_admin.sh
```

**ステップ 3** msc_userservice サービスを再起動します。

```
# docker service update --force --detach=false msc_userservice
```

これにより、管理者ユーザーのパスワードをデフォルトのパスワードにリセットします。デフォルトのパスワードは、実行している Multi-Site Orchestrator の特定のバージョンに依存することに注意してください。使用しているバージョンの『*Cisco Multi-Site* インストールおよびアップグレード ガイド』を参照してください。

# Cisco ACI Multi-Site の外部ユーザー認証に関するトラブルシューティング

次のヒントを使用して、外部ユーザー認証の問題をトラブルシューティングします。

**ステップ 1** 「認証方法が失敗しました」というエラーを調査するには、次のことを確認します。

- プロバイダ構成で指定されたキーが正しい
- Multi-Site（クライアント）の IP アドレスがリモート Cisco ACS サーバに登録されている

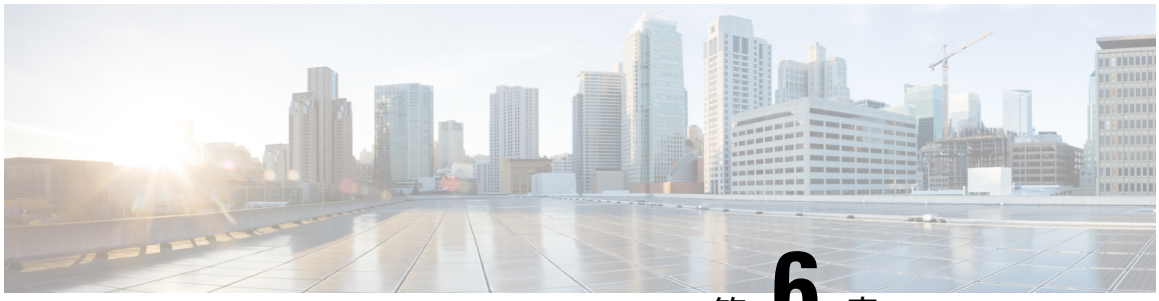
**ステップ 2** エラー無効なユーザー ログイン情報を調査するには、次のことを確認します。

- Multi-Site ログイン画面に入力されたユーザー名は正しく、Cisco ACS サーバで構成されているものと一致する
- Multi-Site ログイン画面に入力されたパスワードが正しく、Cisco ACS サーバで構成されているものと一致する

**ステップ 3** ユーザーに [ロード中（Loading）] アイコンが表示され、続いて「ロード中...」および「認証方法が失敗しました」というエラーが表示される場合は、次のことを確認します。

- プロバイダ構成の IP アドレスが正しい
- プロバイダと Cisco ACS の IP アドレスに到達可能
- プロバイダ構成のポートとプロトコルが正しい
- 正しい認証方法（TACACS+ または RADIUS）がリモート ACS サーバで選択されています...**ネットワーク デバイスおよび AAA クライアント** > **認証オプション**
- 正しい共有シークレットがリモート ACS サーバのユーザー構成で提供されており、空ではありません

**ステップ 4** ユーザーがログインできても、Multi-Site GUI で何も表示されないか、タブを表示できない場合は、リモート ACS サーバで Cisco AV ペアとロールがそのユーザーに対して正しく設定されていることを確認します。



第 6 章

# プラットフォームの健全性問題のトラブルシューティング

この章は、次の項で構成されています。

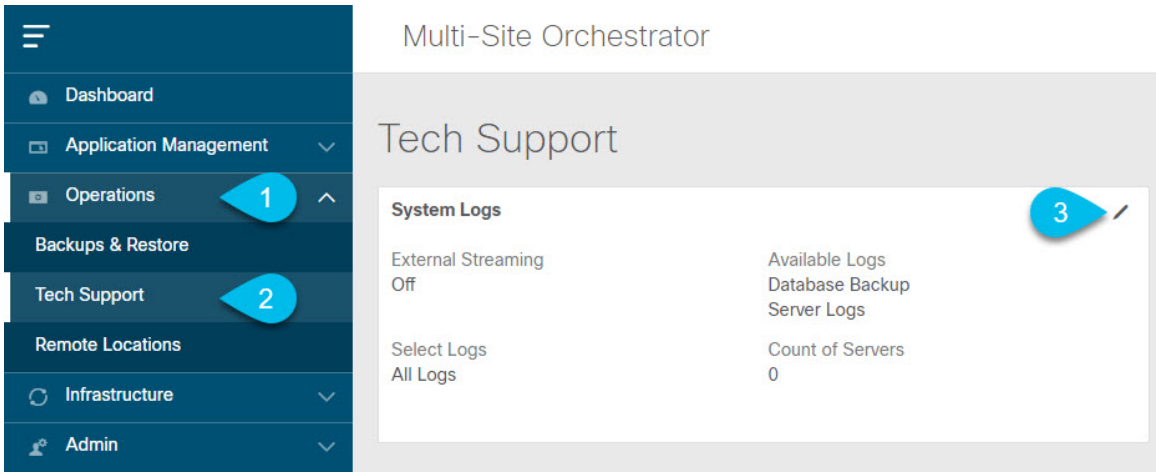


# システム ログのダウンロード

このセクションでは、Cisco ACI マルチサイト Orchestrator により管理されているすべてのスキーマ、サイト、テナント、およびユーザのトラブルシューティング レポートとインフラストラクチャ ログ ファイルを生成します。

**ステップ 1** マルチサイト Orchestrator GUI にログインします。

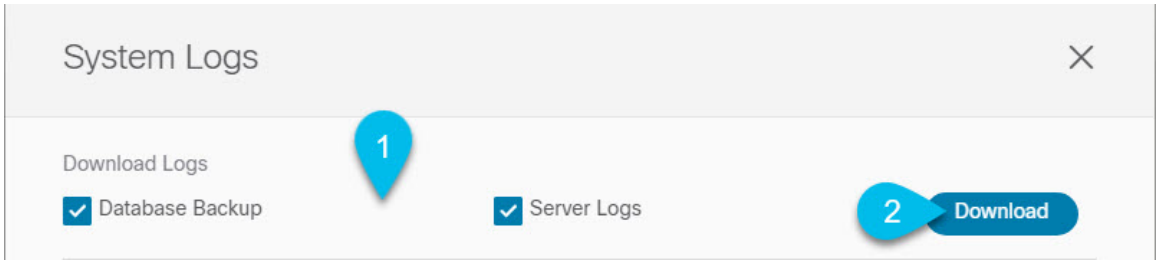
**ステップ 2** **[システムログ (System Logs)]** 画面を開きます。




a) メインメニューで、**[操作 (Operations)]** > **[テクニカル サポート (Tech Support)]**を選択します。

b) **[システム ログ (System Logs)]** フレームの右上隅にある編集ボタンをクリックします。

**ステップ 3** ログをダウンロードします。




a) ダウンロードするログを選択します。

b) [ダウンロード（Download）] ボタンをクリックします。

選択した項目のアーカイブがシステムにダウンロードされます。このレポートには、次の情報が含まれています。

- JSON フォーマットでのすべてのスキーマ
- JSON フォーマットでのすべてのサイト定義
- JSON フォーマットでのすべてのテナント定義
- JSON フォーマットでのすべてのユーザ定義
- `infra_logs.txt` ファイル内のコンテナのすべてのログ

# Docker コンテナ情報の収集

Orchestrator VM の 1 つにログインして、特定のコンテナの Docker サービスとそのログに関する情報を収集できます。次のチート シートには、多くの便利な Docker コマンドが記載されています。https://www.docker.com/sites/default/files/Docker_CheatSheet_08.09.2016_0.pdf

### Docker コンテナの健全性の検査

Docker サービスの正常性を検査するには、`docker service ls` コマンドを使用できます。コマンドの出力には、各サービスの現在のヘルス ステータスが一覧表示されます。`[REPLICAS]` 列に表示されるように、すべてのサービスですべてのコンテナが複製されている必要があります。いずれかがダウンしている場合は、対処が必要な問題が発生している可能性があります。

```
# docker service ls
ID                  NAME                        MODE                REPLICAS  [...]
ve5m9lwb1qc4        msc_auditservice            replicated          1/1       [...]
bl0op2eli7bp        msc_authyldapservice        replicated          1/1       [...]
uxc6pgzficls        msc_authytacacsservice      replicated          1/1       [...]
qcws6ta7abwo        msc_backupservice           global              3/3       [...]
r4p3opyf5dkm        msc_cloudsecservice         replicated          1/1       [...]
xrm0c9vof3r8        msc_consistencyservice      replicated          1/1       [...]
le4gy9kov7ey        msc_endpointservice         replicated          1/1       [...]
micd93h5gj97        msc_executionengine         replicated          1/1       [...]
6wxh4mgnnfi9        msc_jobschedulerservice     replicated          1/1       [...]
lrj1764xw91g        msc_kong                    global              3/3       [...]
n351htjnks75        msc_kongdb                  replicated          1/1       [...]
xcikdpx9o3i6        msc_mongodb1                replicated          1/1       [...]
u9b9ihxxnztn        msc_mongodb2                replicated          1/1       [...]
m0byoou6zuv5        msc_mongodb3                replicated          1/1       [...]
logqawe8k3cg        msc_platformservice         global              3/3       [...]
m3sxof6odn74        msc_schemaservice           global              3/3       [...]
3wd4zrqf6kbk        msc_siteservice             global              3/3       [...]
ourza0yho7ei        msc_syncengine              global              3/3       [...]
ojb8jkkrawqr        msc_ui                      global              3/3       [...]
zm94hzmzzelg        msc_userservice             global              3/3       [...]
```

### コンテナ ID の取得

`docker ps` コマンドを使用して、実行中のすべてのコンテナ ID のリストを取得できます。

```
# docker ps
CONTAINER ID    IMAGE                          COMMAND                  [...]
05f75d088dd1    msc-ui:2.1.2g                  "/nginx.sh"              [...]
0ec142fc639e    msc-authyldap:v.4.0.6          "/app/authyldap.bin"     [...]
b08d78533b3b    msc-cloudsecservice:2.1.2g     "bin/cloudsecservice"    [...]
685f54b70a0d    msc-executionengine:2.1.2g     "bin/executionengine"    [...]
0c719107adce    msc-schemaservice:2.1.2g       "bin/schemaservice"      [...]
f2e3d144738c    msc-userservice:2.1.2g         "bin/userservice"        [...]
edd0d4604e27    msc-syncengine:2.1.2g          "bin/syncengine"         [...]
001616674a00    msc-siteservice:2.1.2g         "bin/siteservice"        [...]
7b30c61f8aa7    msc-platformservice:2.1.2g     "bin/platformservice"    [...]
d02923992d77    msc-backupservice:2.1.2g       "bin/backupservice"      [...]
9de72d291aaa    msc-kong:2.1.2g                "/docker-entrypoint.…"   [...]
6135f9de5dd2    msc-mongo:3.6                  "sh -c 'sleep 3 && e…"   [...]
```

`docker ps | grep <service-name>` コマンドを使用して、特定のサービスの実行中のコンテナ ID を取得できます。

```
# docker ps | grep executionengine
685f54b70a0d    msc-executionengine:2.1.2g   "bin/executionengine"    [...]
```

終了したものを含むサービスのすべてのコンテナ ID を取得するには、`docker ps -a | grep <service-name>` コマンドを使用できます。

```
# docker ps -a | grep executionengine
685f54b70a0d   msc-executionengine:2.1.2g   "bin/executionengine"    Up 2 weeks (healthy)
3870d8031491    msc-executionengine:2.1.2g   "bin/executionengine"    Exited (143) 2
weeks ago
```

### コンテナ ログの表示

`docker logs <container-id>` コマンドを使用して、コンテナのログを表示します。転送するファイルが多くコンテナのログが大きくなる可能性があるため、コマンドを実行するときはネットワーク速度を考慮してください。

コンテナのログ ファイルのサンプルの場所は、`/var/lib/docker/containers/<container>` です。複数の `<container>-json.log` ファイルが存在する場合があります。

```
# cd /var/lib/docker/containers
# ls -al
total 140
drwx------. 47 root root 4096 Jul  9 14:25 .
drwx--x--x. 14 root root 4096 May  7 08:31 ..
drwx------.  4 root root 4096 Jun 24 09:58
051cf8e374dd9a3a550ba07a2145b92c6065eb1071060abee12743c579e5472e
drwx------.  4 root root 4096 Jul 11 12:20
0eb27524421c2ca0934cec67feb52c53c0e7ec19232fe9c096e9f8de37221ac3
[...]
# cd 051cf8e374dd9a3a550ba07a2145b92c6065eb1071060abee12743c579e5472e/
# ls -al
total 48
drwx------.  4 root root 4096 Jun 24 09:58 .
drwx------. 47 root root 4096 Jul  9 14:25 ..
-rw-r-----.  1 root root 4572 Jun 24 09:58
051cf8e374dd9a3a550ba07a2145b92c6065eb1071060abee12743c579e5472e-json.log
drwx------.  2 root root    6 Jun 24 09:58 checkpoints
-rw-------.  1 root root 4324 Jun 24 09:58 config.v2.json
-rw-r--r--.  1 root root 1200 Jun 24 09:58 hostconfig.json
-rw-r--r--.  1 root root   13 Jun 24 09:58 hostname
-rw-r--r--.  1 root root  173 Jun 24 09:58 hosts
drwx------.  3 root root   16 Jun 24 09:58 mounts
-rw-r--r--.  1 root root   38 Jun 24 09:58 resolv.conf
-rw-r--r--.  1 root root   71 Jun 24 09:58 resolv.conf.hash
```

### Docker ネットワークの表示

`docker network list` コマンドを使用して、Docker が使用するネットワークのリストを表示できます。

```
# docker network list
NETWORK ID          NAME                DRIVER              SCOPE
c0ab476dfb0a        bridge              bridge              local
79f5e2d63623        docker_gwbridge     bridge              local
dee475371fcb        host                host                local
99t2hdts7et0        ingress             overlay             swarm
588qhaj3mrj1        msc_msc             overlay             swarm
a68901087366        none                null                local
```

# Stale Docker コンテナの削除

通常、Multi-Site Orchestrator のアップグレードを実行すると、アップグレード プロセスにより、新しいバージョンに置き換えられた古い Docker コンテナがアップグレードから削除されます。ただし、通常の Docker 操作中にサービスに障害が発生した場合などにコンテナが停止し、それらを置き換えるために新しいコンテナが作成される場合があります。これらの停止したコンテナは、次のアップグレードで削除されるまでシステムに残ります。

何らかの理由で古いコンテナを手動で削除する場合は、このセクションの手順を使用してコンテナ ログを収集し、コンテナを削除できます。

**ステップ 1** システムに古いコンテナがあるかどうかを確認します。

システム内に大量の古いコンテナが**ある場合にのみ**、手動でコンテナを削除することをお勧めします。停止しているコンテナが数個しかない場合は、次のアップグレード プロセスで削除されるようにしておくことをお勧めします。

`docker ps -a` コマンドを実行し、終了ステータスのコンテナをチェックすることで、古いコンテナをチェックできます。次に例を示します。

```
# docker ps -a
CONTAINER ID    IMAGE                                 COMMAND                  CREATED          STATUS
5bd87a6a6813    [...]/msc-kong:3.0.1i                 "/docker-entrypoint.…"   23 hours ago     Up 23
hours (healthy)
bf73df31ff51    [...]/msc-ui:3.0.1i                   "/nginx.sh"              25 hours ago     Up 25
hours (healthy)
77c96b515e63    [...]/msc-ui:3.0.1i                   "/nginx.sh"              25 hours ago     Exited
 (1) 25 hours ago
dfedfd82233a    [...]/msc-ui:3.0.1i                   "/nginx.sh"              25 hours ago     Exited
 (1) 25 hours ago
d70aa6262396    [...]/msc-kong:3.0.1i                 "/docker-entrypoint.…"   25 hours ago     Exited
 (143) 23 hours ago
7c15db4db6eb    [...]/msc-endpointservice:3.0.1i      "python3 main.py"        25 hours ago     Up 25
hours (healthy)
```

**ステップ 2** Orchestrator GUI を使用してシステム ログを収集します。

非アクティブなDockerコンテナを削除すると、それらに関連付けられているログもすべて削除されます。古いコンテナを手動でクリーンアップする場合は、後で必要になる場合に備えて、最初にログを収集して保存することをお勧めします。

システム ログの収集については、システム ログのダウンロード （12 ページ） で説明されています。

**ステップ 3** いずれかのノードにログインし、次のコマンドを実行して古いコンテナを削除します。

a) 終了したコンテナが停止していることを確認します。

```
# docker ps --filter "status=exited" --format '{{.ID}}' | xargs --no-run-if-empty docker container
 stop
```

b) コンテナを削除します。

```
# docker ps --filter "status=exited" --format '{{.ID}}' | xargs --no-run-if-empty docker container
 rm -f
# docker container prune -f
```

**ステップ 4** 他の 2 つのノードについても前の手順を繰り返します。

# 欠落しているノード ラベルのトラブルシューティング

マルチサイト Orchestrator GUI にログインできないが、Orchestrator ノードに引き続き SSH 経由でアクセスできる場合は、いずれかのノードのラベルが失われている可能性があります。このセクションでは、この問題を診断し、適切なノードラベルを再適用して解決する方法について説明します。

**ステップ 1** SSH 経由で マルチサイト Orchestrator ノードの 1 つにログインします。

いずれかのノードにログインできます。

**ステップ 2** MongoDB コンテナがすべてのノードで適切に複製されているかどうかを確認します。

```
# docker service ls
ID                  NAME                MODE                REPLICAS            IMAGE
[...]
jvztl0waek4c        msc_mongodb1        replicated          1/1                 msc-mongo:3.6
x1tkpwf1q1df        msc_mongodb2        replicated          1/1                 msc-mongo:3.6
zbi376btmjbg        msc_mongodb3        replicated          0/1                 msc-mongo:3.6
[...]
```

上記の出力では、MongoDB コンテナーがいずれかのノードで適切に複製されていないことがわかります。

**ステップ 3** すべてのノードのホスト名を見つけます。

```
# docker node ls
ID                            HOSTNAME    STATUS      AVAILABILITY    MANAGER STATUS  ENGINE VERSION
z3b6sqc38gfgoerte8cx1w17r     node1       Ready       Active          Reachable       18.06.1-ce
mb3hqe1g0r55oa2zoe32yyfiw *   node2       Ready       Active          Leader          18.06.1-ce
ur5vq2gli8zfc8ngafjn8plej     node3       Ready       Active          Reachable       18.06.1-ce
```

**ステップ 4** 各ノードを検査します。

ノードごとに次のコマンドを繰り返します。<*node-name*> を前の手順のノードのホスト名に置換します。

```
# docker inspect <node-name>
```

**例**：

```
# docker inspect node3
[
    {
        "ID": "ur5vq2gli8zfc8ngafjn8plej",
        "Version": {
            "Index": 317093
        },
        "CreatedAt": "2018-01-19T11:00:41.522951756Z",
        "UpdatedAt": "2019-03-17T07:38:35.487509349Z",
        "Spec": {
            "Labels": {},
            "Role": "manager",
            "Availability": "active"
        },
[...]
```

1 つ以上のノードにラベルがない場合、[ラベル（Labels）] フィールドは空になります。

**ステップ 5** ノードの欠落しているラベルを復元します。

次のコマンド：

- <*node-label*> をノードに適切なラベルに置き換えます。

  各ノードのホスト名はカスタマイズできますが、ラベルは msc-node1、msc-node2、または msc-node3 にする必要があります。

- <*node-name*> をラベルのないノードのホスト名に置き換えます。

```
# docker node update --label-add "msc-node=<node-label>" <node-name>
```

**例**：

```
# docker node update --label-add "msc-node=msc-node3" node3
```

**ステップ 6** 適切に追加されたラベルを確認します。

```
# docker inspect node3
[
    {
        "ID": "ur5vq2gli8zfc8ngafjn8plej",
        "Version": {
            "Index": 317093
        },
        "CreatedAt": "2018-01-19T11:00:41.522951756Z",
        "UpdatedAt": "2019-03-17T07:38:35.487509349Z",
        "Spec": {
            "Labels": {
                "msc-node": "msc-node3"
            },
            "Role": "manager",
            "Availability": "active"
        },
[...]
```

# ストレッチ型BDネットワークのサイト間パケットフローのトラブルシューティング

図 1 は、サイト間のレイヤ 2 ブロードキャスト拡張を使用したストレッチ ブリッジ ドメイン（BD）ネットワークを示しています。BD は、L2 不明ユニキャスト プロキシを使用して、ARP フラッドが有効になっている L3 BD です。

図 *2 :* サイト間 *ARP* フロー

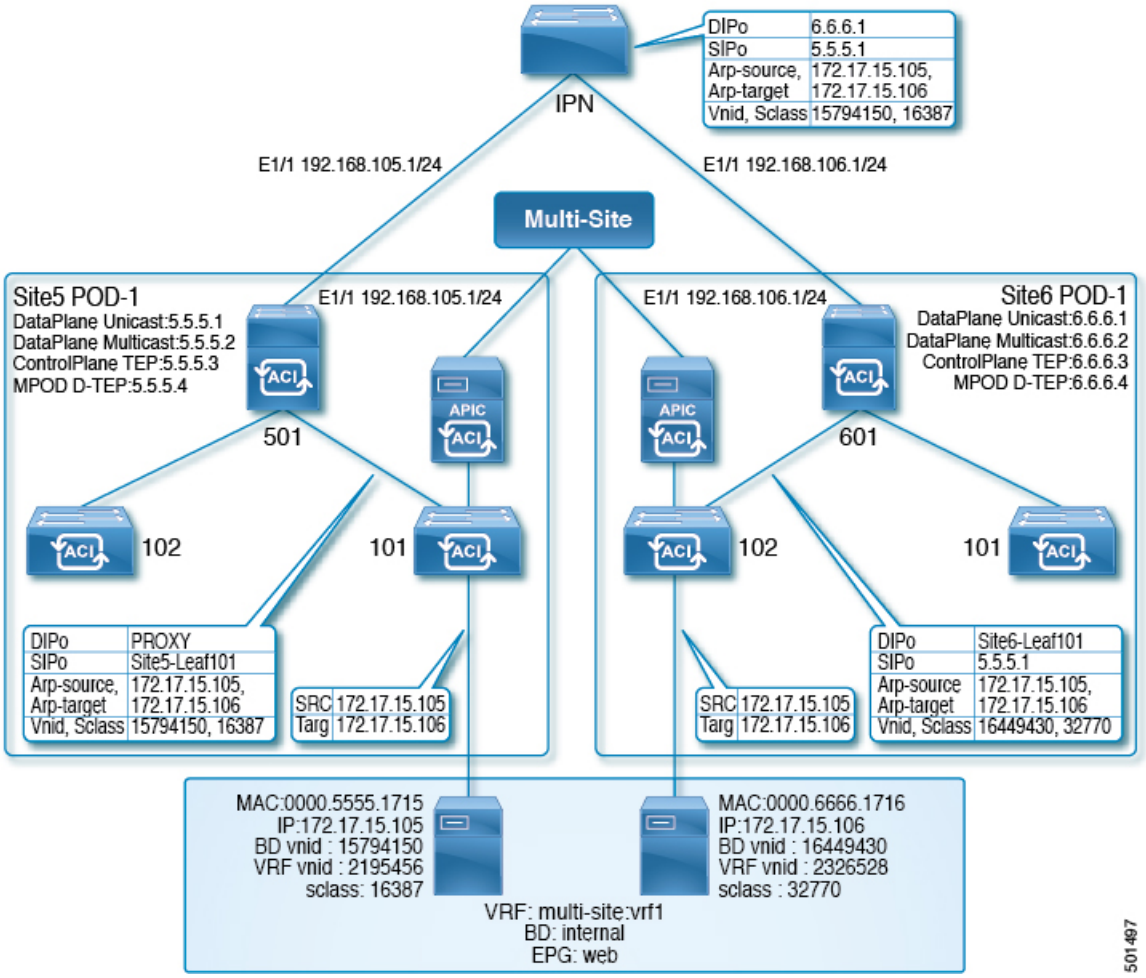
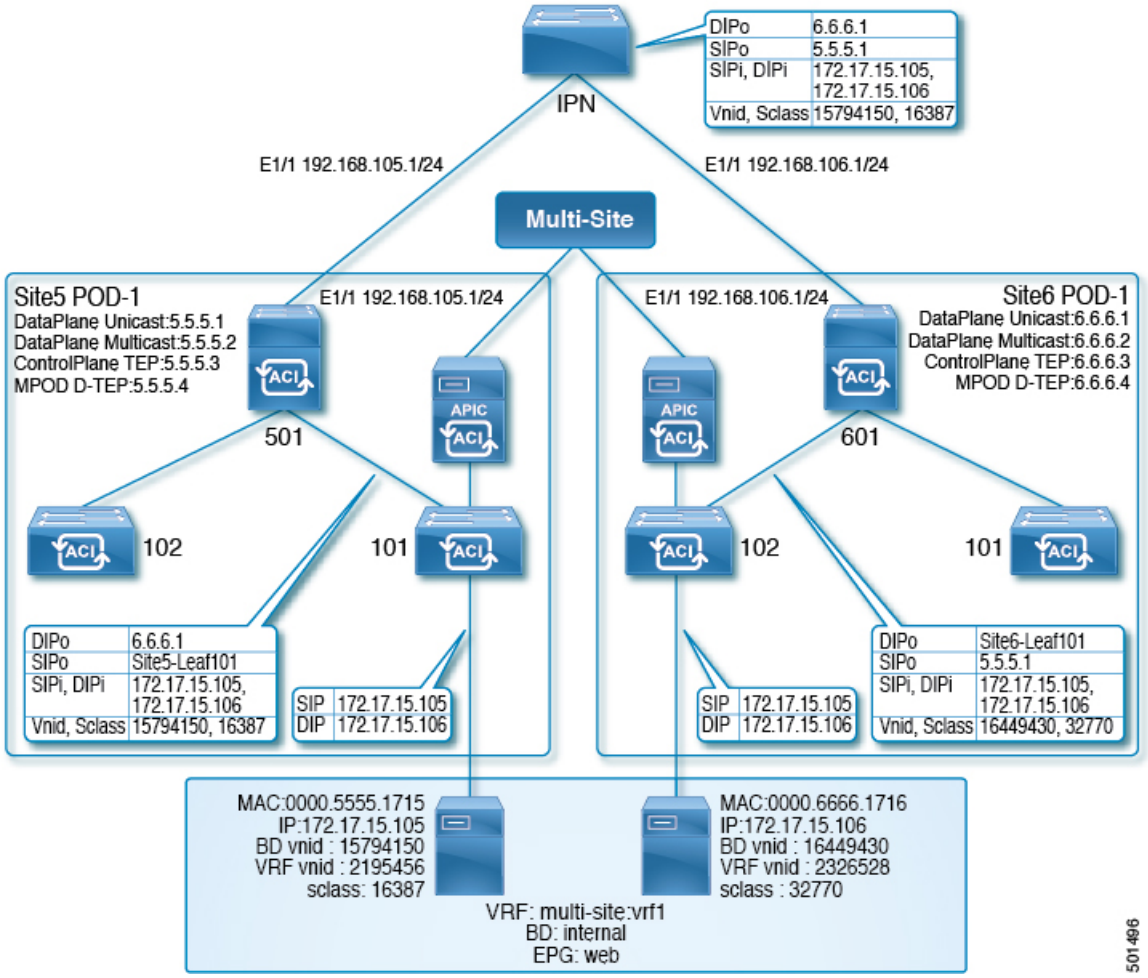



図 2 は、ユニキャスト パケット フローに焦点を当てた同じ拡張 BD ネットワークを示しています。

図 *3:* サイト間ユニキャスト フロー




172.17.15.105 の site5 ホストが IP アドレス 172.17.15.106 の site6 ホストにユニキャスト パケット（たとえば、ICMP エコー）を送信する場合、Site5-leaf101 が site6 エンドポイント（EP）、172.17.15.106 を学習したシナリオには、次のトラブルシューティング手順が適用されます。site5-leaf101 が site6 EP を学習していない場合は、BD のレイヤ 2 の不明なユニキャスト転送設定に基づいて、パケットをフラッディングするか、プロキシ用に pine501 データを送信します。

**ステップ 1** 次の例のように、site5 入力リーフ スイッチ（この場合は leaf101）で、NX-OS スタイル CLI **show endpoint mac** *mac-address* コマンドを使用して、システムが送信元 EP と宛先 EP の両方を学習したかどうかを判断します。

**例**：

```
leaf101# show endpoint mac 0000.6666.1716
Legend:
s - arp              O - peer-attached    a - local-aged       S - static
V - vpc-attached     p - peer-aged        M - span             L - local
B - bounce           H - vtep
+-----------------+---------------+-----------------+--------------+-------------+
      VLAN/         Encap           MAC Address       MAC Info/       Interface
```

```
      Domain        VLAN            IP Address        IP Info
+-----------------+---------------+-----------------+--------------+--------------+
3                    vlan-201       0000.6666.1716 L                 eth1/40
msite-site:vrf1      vlan-201       172.17.15.106 L                  eth1/40
```

**ステップ 2** ローカルおよびリモート EP の両方が学習されており、ポリシーにより EP の通信を許可する場合（EPG 内トラフィックを許可するデフォルトのコントラクトを使用）、site5-leaf101 は、ICMP パケットを次のデータでカプセル化し、ファブリック アップリンク ポートを介してスパイン スイッチにパケットを転送します。

- VXLAN ヘッダーの外部宛先 IP アドレス、6.6.6.1
- VXLAN ID (VNID)、15794150
- src-class (sclass), 16387
- VRF オーバーレイ 1 を介した site5-leaf101 TEP アドレスである送信元 IP アドレス

スパイン スイッチは、VRF オーバーレイ 1 からパケットを受信すると、宛先 IP アドレス（DIP）が MAC プロキシ アドレスに属していることを確認します。たとえば、DIP 6.6.6.1 がpine501 の MAC プロキシ アドレスに属していない場合、スパイン スイッチは、ルーティング テーブルの最長一致に基づいて、通常の IP パケットのようにパケットを転送します。この場合、DIP はリモート サイトのスパイン オーバーレイのユニキャスト TEP アドレスと一致するため、spine501 は外部送信元 IP（SIP）アドレスを、site5-leaf101 の TEP から site5 のユニキャスト オーバーレイ ユニキャスト TEP（5.5.5.1）に書き換えます。このプロセスでは、Spine501 はポッド間ネットワーク（IPN）の OSPF を介して 6.6.6.1 を学習する必要があるため、spine501 はパケットをネクスト ホップ（この場合は IPN スイッチ）に転送します。

**ステップ 3** パケットの転送に懸念がある場合は、NX-OS スタイル CLI の APIC のファブリック モードで ERSPAN を実行し、次の例のようなコマンドを使用して、アップリンク インターフェイスからの発信パケットをキャプチャします。

**例**：

この例では、テナント t1 の VRF vrf1 および BD bd1 にフォーカスし、スイッチ 101、インターフェイス eth1/1 からの発信パケットをキャプチャするようにファブリック ERSPAN を構成します。

```
apic1# configure terminal
apic1(config)# monitor access session mySession
apic1(config-monitor-fabric)# description "This is my fabric ERSPAN session"
apic1(config-monitor-fabric)# destination tenant t1 application app1 epg epg1 destination-ip
192.0.20.123 source-ip-prefix 10.0.20.1
apic1(config-monitor-fabric-dest)# erspan-id 100
apic1(config-monitor-fabric-dest)# ip dscp 42
apic1(config-monitor-fabric-dest)# ip ttl 16
apic1(config-monitor-fabric-dest)# mtu 9216
apic1(config-monitor-fabric-dest)# exit
apic1(config-monitor-fabric)# source interface eth 1/1 switch 101
apic1(config-monitor-fabric-source)# direction tx
apic1(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
```

詳細については、『*Cisco APIC NX-OS* スタイル コマンドライン インターフェイス構成ガイド』の「*SPAN* の構成」を参照してください。

**ステップ 4** ルーティング テーブルに 6.6.6.1 の明示的なエントリが含まれているかどうかを確認するには、NX-OS スタイル コマンド、**show ip route 6.6.6.1 vrf overlay-1** を使用します。

**ステップ 5** ネクスト ホップ インターフェイスが見つかったら、**show lldp neighbor** コマンドを使用して、6.6.6.1 の学習元であるインターフェイスのネクスト ホップが予期される IPN インターフェイスであるかどうかを判断します。

ファブリック ERSPAN を使用して、spine501 がリーフ スイッチからパケットを受信したか、正しい出力インターフェイスを介してパケットを転送したことを確認します。

**ステップ 6** パケットが IPN に到着すると、これはユニキャスト パケットであるため、IPN はルーティング テーブルに基づいて IP パケットを転送します。ルーティング テーブルに正しい/予想されるネクスト ホップ インターフェイスがあることを確認するには、コマンド **show ip route 6.6.6.1** を使用します。

ファブリック ERSPAN を使用して、異なるインターフェイスからの 1 つまたは複数のパケットをキャプチャします。上記のトポロジの IPN からのネクストホップ インターフェイスは、spine601 のインターフェイスです。

パケットが site6 スイッチ、spine601 に到着すると、外部 SIP に基づいてリモート サイトの ID 5.5.5.1 を site5 にマッピングし、送信元 VNID を 15794150 にマッピングします。また、spine601 はその VNID をローカル BD の VNID、16449430 に変換し、src-class ID、16387 をローカル EP src-clas、32770 に変換します。次に、変換された VNID の範囲内で、宛先 MAC アドレスに基づいてルックアップを実行します。

**ステップ 7** site5 と site6 の間の VNID 変換を確認するには、spine601 で **show dcimgr repo vnid-maps verbose** コマンドを入力します。

**ステップ 8** site5 と site6 の間の sclass 変換を確認するには、spine601 で **show dcimgr repo sclass-maps** コマンドを入力します。

最後に、spine601 は、外部宛先を site6-leaf101 の TEP に書き換えて、そこにパケットを転送します。

**ステップ 9** パケットが正しく転送されたかどうかを判断するには、予期されるリーフ スイッチ（site6-leaf101）に移動して、ファブリック ERSPAN を実行してパケットをキャプチャします。

**ステップ 10** パケットが site6-leaf101 に到着すると、leaf101 は、VNID 16449430 の範囲内の宛先 MAC に基づいてローカル ルックアップを実行して、出力インターフェイスを決定します。出力インターフェイスを判別するには、**show endpoint mac** *mac-address* コマンドを入力します。

**ステップ 11** パケットが正しく転送されたかどうかを判断するには、アクセス SPAN を使用し次の例のようなコマンドを使用して、予想されるインターフェイスで発信パケットをキャプチャします。

**例**：

この例では、アクセス モードで SPAN を構成して、テナント t1 の EPG epg1 にフォーカスして、リーフ 101、インターフェイス eth1/2 で送信されるパケットをキャプチャします。

```
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 app1 epg epg1
apic1(config-monitor-access-source)# exit
apic1(config-monitor-access)# no shut
apic1(config-monitor-access)# show run
```

これは従来の SPAN 構成であり、アクセス リーフ ノードに対してローカルです。1 つ以上のアクセス ポートまたはポート チャネルから発信されたトラフィックをモニタリングし、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。

ACI ファブリックで、アクセス モード ERSPAN 構成を使用して、1 つ以上のリーフ ノードでアクセス ポート、ポート チャネル、vPC から発生したトラフィックをモニタできます。ERSPAN セッションの場合、宛先は常にファブリックで展開可能な EPG です。監視対象のトラフィックは、どこであれ、EPG が移動した場所である宛先に転送されます。

詳細については、『*Cisco APIC NX-OS* スタイル コマンドライン インターフェイス構成ガイド』の「*SPAN* の構成」を参照してください。

# サイト間 BGP セッションのトラブルシューティング

サイト スパイン スイッチでマルチサイト BGP セッションを確立するには、次の設定が必要です。

- 更新元には `mscp-etep` フラグが設定されている必要があります
- BGP ピア タイプは `inter-site` である必要があります
- ノードの役割は `msite-speaker` である必要があります

サイト間 BGP セッション障害のトラブルシューティングを行うには、Visore を使用して、スパイン上の次の MO を確認します。

- `fvNodeDef`
- `bgpInfraPeerDef`
- `bgpAsP`
- `fvIntersitePeeringDef`
- `l3extIntersiteLoopBackIfPDef`
- タイプがサイト間に設定されている `l3LbRtdIf`
- `LoopBackId`
- 同じループバック ID を持つ `ipv4If` では、modeExtn プロパティが `mscp-etep` に設定されています。

これらの MO のいずれかが欠落している場合、BGP セッションは起動しません。

Visore を使用してクエリを入力する方法については、『*Cisco APIC REST API* 構成ガイド』の「*REST API* の使用」の「*REST API* ツールへのアクセス」を参照してください。

（注） Visore は Firefox、Chrome、および Safari ブラウザでサポートされています。

**ステップ 1** サポートされているブラウザで、次の例のようにスパイン スイッチの URL に続けて `/visore.html` を入力します。

例：

```
https://spine-ip-address/visore.html
```

**ステップ 2** プロンプトが表示されたら、スパイン CLI インターフェイスにログインする際に使用したのと同じログイン情報を使用してログインします。

**ステップ 3** `l3LbRtdIf` のクエリを入力して、タイプが `inter-site` であることを確認します。

**ステップ 4** `Ipv4IF` のクエリを入力して、モードが `cp-etep` であり、`modeExtn` が `mscp-etep` であることを確認します。

**ステップ 5** サイト間 `BgpPeers` のクエリを入力して、サイト間タイプで作成されたことを確認し、CP-TEP ループバック アドレスを送信元インターフェイスとして使用します。

**ステップ 6** これらの値のいずれかが正しくない場合は、サイトの APIC にアクセスして値を修正してください。マルチサイト の **[サイト（Sites）]** タブに戻り、[インフラストラクチャの構成（CONFIGURE INFRA）] をクリックしてから、**[適用（Apply）]** をクリックします。

# スパインスイッチの再稼働後のBGP接続損失からの回復

APIC から削除せずに、ファブリックの 1 つでマルチサイト スパイン スイッチをデコミッションおよび再コミッションすると、外部 BGP ピアリングがオフになり、スイッチで無効のままになる場合があります。この BGP ピアリングは、サイト間通信のために Multi-Site Orchestrator に必要です。

このセクションでは、最新のサイト接続情報をロードし、必要に応じてインフラ構成をサイトに再展開することにより、BGP ピアリングを再確立する方法について説明します。

**ステップ 1** Cisco ACI マルチサイト Orchestrator GUI にログインします。

**ステップ 2** サイト接続情報を更新します。

a) **[メイン メニュー (Main menu)]** で、**[インフラストラクチャ (Infrastructure)]** > **[インフラの設定 (Infra Configuration)]**を選択します。

b) 右上にある **[インフラの構成 (Infra Configuration)]** ビューで、**[インフラの設定 (Configure Infra)]** ボタンをクリックします。

c) 左側のウィンドウの **[サイト（Sites）]** で、スパイン スイッチが再委託されたサイトを選択します。

d) メイン ウィンドウで、**[サイト データのリロード（Reload Site Data）]** ボタンをクリックし、APIC からファブリック情報をプルします。

e) **[確認（Confirmation）]** ダイアログで、**[デコミッションされたスパインノードの構成を削除（Remove config for decommissioned spine nodes）]** チェックボックスがオンになっていることを確認します。

このチェックボックスを選択すると、廃止されたスパイン スイッチの古い構成情報が Multi-Site Orchestrator データベースから削除されます。

f) 最後に、**[はい (Yes)]** をクリックして確認し、接続情報をロードします。

これにより、APICから再インポートすることにより、再コミッションされたスイッチを含むサイト接続情報が更新されます。

**ステップ 3** スパイン スイッチの構成を確認します。

前の手順で更新したサイトでスパイン スイッチを選択し、すべての構成が正しいことを確認します。

情報を更新する必要がある場合は、『*Cisco ACI Multi-Site* 構成ガイド』の「インフラストラクチャ管理」の章で、スパイン スイッチのインフラ構成に関する詳細な手順を参照できます。

**ステップ 4** メインの **[ファブリック接続インフラ（Fabric Connectivity Infra）]** ビューの右上で、**[展開（Deploy）]** ボタンをクリックします。

Multi-Site 展開にクラウドサイトがある場合は、ここで複数のオプションを使用できます。再コミッションされたスイッチのオンプレミス サイトのみを更新するため、**[展開（Reload Site Data）]** をクリックするだけで、そのサイトにインフラ構成を展開できます。

# ユニキャストまたはマルチキャスト トラフィック障害のトラブルシューティング

Visore で次の手順を使用して、サイト間のユニキャストおよびマルチキャスト トラフィックの障害をトラブルシューティングします。

Visore を使用してクエリを入力する方法については、『*Cisco APIC REST API* 構成ガイド』の「*REST API* の使用」の「*REST API* ツールへのアクセス」を参照してください。

（注） Visore は Firefox、Chrome、および Safari ブラウザでサポートされています。

**ステップ 1** スパイン スイッチの Visore ページに移動します。

`https://<spine-ip-address>/visore.html`

**ステップ 2** スパイン CLI インターフェイスにログインする際に使用したのと同じログイン情報を使用してログインします。

**ステップ 3** クエリを入力して、`fvIntersiteConnPDef` および `fvIntersiteMcastConnPDef` MO が `fvSiteConnPDef` の下にあることを確認します。

これらは、リモート サイトのユニキャストおよびマルチキャスト DP TEP です。

**ステップ 4** クエリを入力して、`tunnelIf` MO がタイプ `dci-ucast` または `dci-mcast-hrep` で作成され、宛先がリモートサイトの DP TEP と同じであることを確認します。

**ステップ 5** ローカル サイトのユニキャストおよびマルチキャスト DP TEP を確認します。`fvPodConnPDef` の下の `fvIntersiteConnPDef` および `fvFabricExtConnPDef` の下の `fvIntersiteMcastConnPDef` のクエリを入力します。

**ステップ 6** `SiteLocal ipv4If` MO のクエリを入力して、それらがモード `dci-ucast` および `dci-mcast-hrep` で作成されたこと、および `ipv4Addr` MO が DP TEP と同じアドレスを使用してその MO の下に構成されていることを確認します。

**ステップ 7** これらの値のいずれかが正しくない場合は、サイトの APIC にアクセスして値を修正してください。マルチサイト の **[サイト（Sites）]** タブに戻り、[インフラストラクチャの構成（CONFIGURE INFRA）] をクリックしてから、**[適用（Apply）]** をクリックします。

# Multi-Site マルチキャスト機能のトラブルシューティング

このタスクでは、ストレッチブリッジドメイン（BD）の使用例でマルチサイトマルチキャスト機能をトラブルシューティングするための手順を示します。このトピックでは、ストレッチ BD で `L2STRETCH` および `INTERSITEBUMTRAFFICALLOW` オプションが有効になっていることを前提としています。

マルチキャスト トラフィックは、次のプロセスでサイト間を流れます。

- **ローカル サイトからリモート サイトへの TX（送信）**

  ローカル サイトからのグループ IP 外部アドレス（GIPo）トラフィック（レイヤ 2 ブロードキャストの一部、不明なユニキャスト、マルチキャスト トラフィック）は、スパイン スイッチから各リモート サイトへのヘッドエンド複製（HREP）です。外部ヘッダー（DIPo）の宛先 IP アドレスは、リモート サイトのマルチキャスト HREP TEP IP（マルチキャスト DP-TEP IP とも呼ばれる）と呼ばれるユニキャスト アドレスに書き換えられます。外部ヘッダー（SIPo）の送信元 IP アドレスは、ユニキャスト ETEP IP で書き換えられます。

- **ローカル サイトからのリモートによる RX（受信）**

  ローカル サイト宛ての着信トラフィック マルチキャスト HREP TEP IP アドレスが変換されます。サイトの APIC は、そのデータからローカル サイト BD-GIPo を取得し、それ以降は通常の GIPo ルックアップ パスに従います。

このプロセスの問題をトラブルシューティングするには、スパイン スイッチの CLI にログオンし、次の手順を実行します。

**ステップ 1** ローカルに構成された Multi-Site TEP IP アドレスを確認するには、スーパーバイザ モジュールにログオンし、次の例のようなコマンドを入力します。

**例：**

```
swmp11-spine6# show ip interface vrf overlay-1
loopback11, Interface status: protocol-up/link-up/admin-up, iod: 126, mode: dci-ucast, vrf_vnid:
```

```
 16777199
  IP address: 33.20.1.1, IP subnet: 33.20.1.1/32
  IP primary address route-preference: 1, tag: 0
loopback12, Interface status: protocol-up/link-up/admin-up, iod: 127, mode: mcast-hrep, vrf_vnid:
 16777199
  IP address: 33.30.1.1, IP subnet: 33.30.1.1/32
```

**ステップ 2** スパイン スイッチの MFDM を確認するには、スーパーバイザ モジュールにログオンし、次の例のようなコマンドを入力します。

**例**：

```
swmp11-spine6# show forwarding distribution multicast hrep
MFDM HREP NODE TABLE
--------------------
IP Address: 0xb1e0101
Table Id: 2
Flags: 0x0
IfIndex: 0x18010009
Internal BD 0x1001
Internal encap 0xb54
Nexthop Information: (num: 5)
Address               Ifindex                  Dvif
0x14950a02        0x1a018019        0x1eb (Selected)  <== Selected NH to reach the HREP TEP IP
0x14950602        0x1a00e00f        0x0
0x14950802        0x1a010011        0x0
0x14950902        0x1a011012        0x0
0x14950b02        0x1a01901a        0x0
```

**ステップ 3** HREP TEP IP アドレスの到達可能性を確認するには、スーパーバイザ モジュールにログオンし、次の例のようなコマンドを入力します。

**例**：

```
swmp11-spine6# show ip route 11.30.1.1 vrf overlay-1
11.30.1.1/32, ubest/mbest: 5/0
    *via 20.149.6.2, Eth1/15.15, [110/9], 1d21h, ospf-default, intra
    *via 20.149.8.2, Eth1/17.17, [110/9], 1d21h, ospf-default, intra
    *via 20.149.9.2, Eth1/18.18, [110/9], 1d21h, ospf-default, intra
    *via 20.149.10.2, Eth1/25.25, [110/9], 1d21h, ospf-default, intra
    *via 20.149.11.2, Eth1/26.26, [110/9], 1d21h, ospf-default, intra
     via 10.0.112.95, Eth2/21.77, [115/65], 1d21h, isis-isis_infra, L1
     via 10.0.112.95, Eth1/24.35, [115/65], 1d21h, isis-isis_infra, L1
     via 10.0.112.92, Eth2/19.76, [115/65], 1d21h, isis-isis_infra, L1
     via 10.0.112.92, Eth1/21.36, [115/65], 1d21h, isis-isis_infra, L1
     via 10.0.112.90, Eth2/17.75, [115/65], 1d21h, isis-isis_infra, L1
     via 10.0.112.90, Eth1/23.33, [115/65], 1d21h, isis-isis_infra, L1
```

**ステップ 4** スパイン スイッチ ラインカード モジュールの MFIB を確認するには、ルートとしてモジュールにログオンし、次の例のような（vsh_lc）コマンドを使用します。

**例**：

```
root@module-1# show forwarding multicast hrep tep_routes

****HREP TEP ROUTES****
-------------------------------------------------------------------------------------------------------
| Tep Ip    |   Tep If    |    NH Ip     |       NH If     |     NH dmac    |    NH dvif | Vlan
Id | Bd Id |
-------------------------------------------------------------------------------------------------------
|22.30.1.1    |0x1801000b   |20.149.11.2    |0x1a01901a     |00c8.8bba.54bc |490  |2901   |4098
  |
|11.30.1.1    |0x18010009   |20.149.10.2  |0x1a018019   |00c8.8bba.54bc |491  |2900   |4097   |
```

**ステップ 5** リモート サイトのマルチキャスト HREP TEP の詳細を確認するには、次の例のようなコマンドを入力して、スパイン スイッチ ラインカード モジュールにログオンして SDK を調査します。

**例**：

```
root@module-1# show platform internal hal objects mcast hreptep
## Get Objects for mcast hreptep for Asic 0
  OBJECT 1:
Handle                                                  : 52303
tepifindex                                              : 0x18010009
tepipaddr                                               : 11.30.1.1/0
intbdid                                                 : 0x1001
intvlanid                                               : 0xb54
nexthopipaddr                                           : 20.149.10.2/0
nexthopifindex                                          : 0x1a018019
nexthopmacaddr                                          : 00:c8:8b:ba:54:bc
```

**ステップ 6** リモート サイトの HREP トンネルを持つ GIPo ルートを確認するには、スパイン スイッチのスーパーバイザ モジュールにログオンし、次の例のようなコマンドを使用して、スパイン スイッチの IS-IS 詳細を調べます。

**例**：

```
swmp11-spine6# show isis internal mcast routes gipo
GIPo: 225.0.6.176 [TRANSIT]
    OIF List:
      Ethernet1/21.36
      Ethernet1/23.33
      Ethernet1/24.35
      Tunnel9         <== Multicast HREP tunnel for Remote Site 1
      Tunnel1         <== Multicast HREP tunnel for Remote Site 2
      Ethernet2/17.75
      Ethernet2/19.76
      Ethernet2/21.77
```

**ステップ 7** リモート サイトの HREP トンネルを持つ GIPo ルートを確認するには、次の例のようなコマンドを使用して、スパイン スイッチの MRIB を調べます。

**例**：

```
swmp11-spine6# show ip mroute 225.0.6.176 vrf overlay-1
IP Multicast Routing Table for VRF "overlay-1"
(*, 225.0.6.176/32), uptime: 1d02h, isis
  Incoming interface: Null, RPF nbr: 0.0.0.0
  Outgoing interface list: (count: 8)
    Tunnel9, uptime: 1d01h
    Tunnel11, uptime: 1d02h
    Ethernet2/21.77, uptime: 1d02h
    Ethernet2/19.76, uptime: 1d02h
    Ethernet2/17.75, uptime: 1d02h
    Ethernet1/24.35, uptime: 1d02h
    Ethernet1/23.33, uptime: 1d02h
    Ethernet1/21.36, uptime: 1d02h
```

**ステップ 8** FC の MFIB を確認するには、root としてモジュールにログオンし、次の例のようなコマンドを使用します。

**例**：

```
root@module-24# show forwarding multicast route group 225.0.6.176 vrf all
  (*, 225.0.6.176/32), RPF Interface: NULL, flags: Dc
    Received Packets: 0 Bytes: 0
    Number of Outgoing Interfaces: 8
```

```
    Outgoing Interface List Index: 484
      Ethernet1/21.36 Outgoing Packets:N/A Bytes:N/A
      Ethernet1/23.33 Outgoing Packets:N/A Bytes:N/A
      Ethernet1/24.35 Outgoing Packets:N/A Bytes:N/A
      Tunnel9 Outgoing Packets:0 Bytes:0
      Tunnel11 Outgoing Packets:0 Bytes:0
      Ethernet2/17.75 Outgoing Packets:N/A Bytes:N/A
      Ethernet2/19.76 Outgoing Packets:N/A Bytes:N/A
      Ethernet2/21.77 Outgoing Packets:N/A Bytes:N/A
```

**ステップ 9** リモート サイト用の HREP トンネルを持つ GIPo ルートを確認するには、次の例のようなコマンドを使用して、FC 上の SDL を調べます。

**例**：

```
root@module-24# show platform internal hal objects mcast l3mcastroute groupaddr 225.0.6.176/32
extensions
## Get Extended Objects for mcast l3mcastroute for Asic 0
  OBJECT 0:
Handle                                                    : 78705
groupaddr                                                 : 225.0.6.176/32
grpprefixlen                                              : 0x20
sourceaddr                                                : 0.0.0.0/32
ispimbidir                                                : Enabled
ctrlflags                                                 : UseMetFlag,
rtflags                                                   : none, UseMetEntry,
acirtpolicy                                               : none
 - - - - - - - - -
Relation Object repllistnextobj :
  rel-repllistnextobj-mcast-mcast_mcast_repl_list-handle  : 78702
  rel-repllistnextobj-mcast-mcast_mcast_repl_list-id      : 0x600001e4
```

**ステップ 10** リモート サイトへの GIPo ルートを確認するには、次の例のようなコマンドを使用して、最後の手順で確認した複製リストを調べます。

**例**：

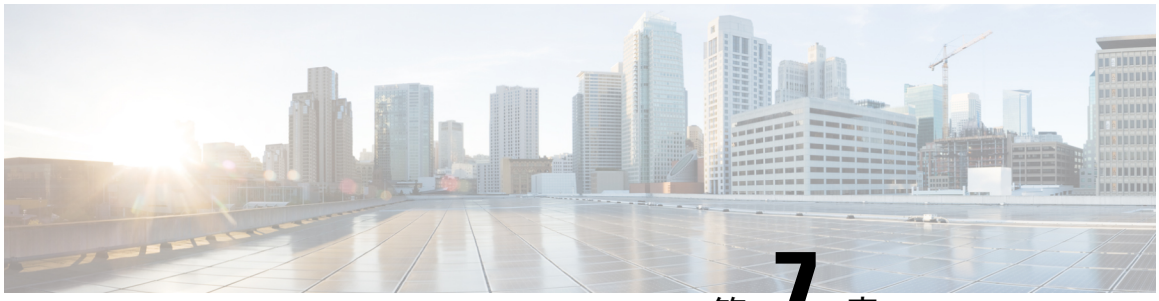
```
root@module-24# show platform internal hal objects mcast mcastrepllist id 0x600001e4
## Get Objects for mcast mcastrepllist for Asic 0
- - - - - - - - - -
Repl-List Asicpd Debug :
Entry-Num 0
Repl Entry Id:      0x1e5 Hw Epg Id:    4050 Hw Bd Id:   4050
Mc Id:                484 Met Id:        485 Encap Id:     -1
Sh Grp:                 0 Next Met Id:   749
Entry-Num 1
Repl Entry Id:      0x2ed Hw Epg Id:    4098 Hw Bd Id:   4098
Mc Id:                490 Met Id:        749 Encap Id:     -1
Sh Grp:                 0 Next Met Id:  1191
Entry-Num 2
Repl Entry Id:      0x3a4 Hw Epg Id:    4097 Hw Bd Id:   4097
Mc Id:                491 Met Id:       1191 Encap Id:     -1
Sh Grp:                 0 Next Met Id:     0
```

**ステップ 11** ローカル（TX）サイトとリモート（RX）サイトで VNID と GIPo のマッピングを確認するには、次の例のようなコマンドを入力します。

**例**：

```
root@module-2# show platform internal hal objects dci vnidmap extensions | grep -B 5 -A 5 225.1.148.0

  OBJECT 182:
Handle                                                    : 26456
```

```
isbdvnid                                                : Enabled
localvnid                                               : 0xe78007
localgipo                                               : 225.1.148.0/32
remotevnid                                              : 0xe1000c
remotevrfvnid                                           : 0x208019
islocalbdctrl                                           : Enabled
siteid                                                  : 0x3

 OBJECT 1285:
Handle                                                  : 29468
isbdvnid                                                : Enabled
localvnid                                               : 0xe78007
localgipo                                               : 225.1.148.0/32
remotevnid                                              : 0xee7fa8
remotevrfvnid                                           : 0x2e000e
islocalbdctrl                                           : Enabled
siteid                                                  : 0x2
```



第 7 章

# テナントとスキーマのトラブルシューティング

この章は、次の内容で構成されています。



## APIC からの展開エラーのトラブルシューティング

Cisco ACI マルチサイト スキーマで構成されているテナント ポリシーを展開するとき、エラーを受け取るか、問題が発生する可能性があります。これらのエラーと問題のトラブルシューティングを行うには、次の手順に従います。

**ステップ 1** **[サイトへの展開（DEPLOY TO SITES）]** をクリックした後に APIC エラーが発生した場合は、問題を修正し、スキーマ/テンプレートを再度展開してみてください。たとえば、エラーには次の 2 種類があります。

- 設定ミス：たとえば、BD の VRF の選択を忘れるなど、必要な関連付けが定義されていません。
- サイトの問題—ネットワーク障害、通信障害、またはインフラ設定の問題などの問題がある可能性があります。スキーマを保存し、サイトの問題に対処してから、スキーマの展開に戻ります。

**ステップ 2** スキーマが正常に展開されてもトラフィックが流れない場合は、次の手順を実行します。

a) トラブルシューティング レポートを生成し、エラーを調べます。システム ログのダウンロード （12 ページ） を参照してください

b) ポリシー レポートを生成し、テナント ポリシー構成を調べます。REST API を使用したテナント ポリシー レポートの生成 （58 ページ） を参照してください

c) Multi-Site VM にログオンし、実行ログを生成してエラーを見つけます。Docker コンテナ情報の収集 （14 ページ） および 実行ログの読み取り （17 ページ） を参照してください。

**ステップ 3** エラーが見つからない場合は、APIC、スイッチ、IPN、または WAN に問題がある可能性があります。

# REST API を使用したテナント ポリシー レポートの生成

テナント ポリシー レポートを生成するには、マルチサイト REST API を使用して、次の例のようなクエリを入力します。

クエリを受信すると、マルチサイト ではテナントで定義されているすべてのポリシーの APIC をクエリし、マルチサイトと APIC の間のトラフィックを生成します。これは、メンテナンス期間中に行うことができます。

問題のあるテナントを一覧表示するには、次の例のようなクエリを入力し、出力をコピーします。

**例**：

```
GET https://multi-site-ip-address/api/v1/policy-report?
tenants=tenant1,tenant2&validate=true
```

# スキーマおよびテンプレートの展開解除

トラブルシューティングで、一部のテナントポリシーが正しく構成されていないことがわかった場合は、テンプレートまたはスキーマを展開解除して、後で再作成することができます。テンプレートとスキーマを展開解除するには、次の手順に従います。

**ステップ 1** 1 つのサイトに展開されているテンプレートを展開解除します。

a) [スキーマ（Schema）] タブで、サイト固有のテンプレートの 3 つのドットをクリックします。
b) **[はい（YES）]** をクリックし、確認します。

**ステップ 2** 複数のサイトに展開されている 1 つのサイトでテンプレートを展開解除します。

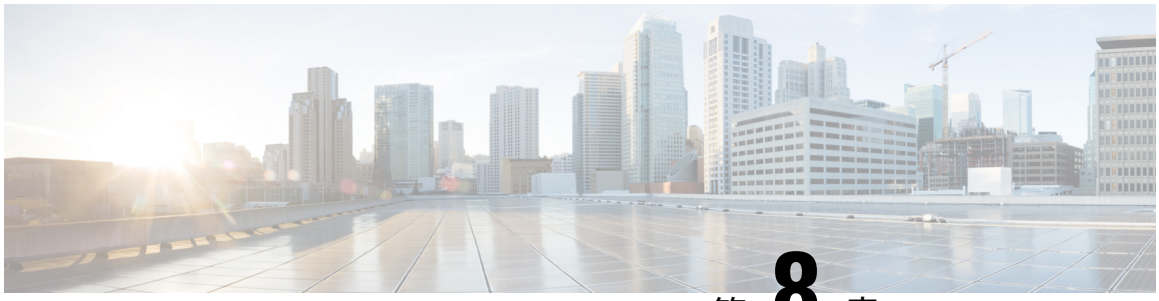
a) [スキーマ（Schema）] タブで、テンプレートをクリックします。
b) **[+]** をクリックし、サイト選択パネルを開きます。
c) サイトの行で、テンプレートの **[X]** をクリックします。
d) **[保存 (SAVE)]** をクリックします。
e) 残りのテンプレートをサイトに再展開します。

**ステップ 3** すべてのサイトからスキーマを削除します。

a) [スキーマ（Schema）] タブで、スキーマをクリックします。
b) [アクション（Actions）] をクリックし、[削除（Delete）] を選択します。
c) スキーマを展開解除することを確認し、**[はい（YES）]** をクリックします。

### 次のタスク

スキーマまたはテンプレートを修正してから、再展開してください。



第 8 章

# マルチポッドおよびマルチキャストの問題のトラブルシューティング

この章は、次の項で構成されています。



## マルチサイトとマルチポッドのトラブルシューティング

このセクションでは、マルチサイトおよびマルチポッドをトラブルシューティングする方法を説明します。

### エラー：400

次のエラーが表示される場合

```
Error:400 - Invalid Configuration Following Intersite Spines are not configured as Mpod
 Spines: 1202
```

既存のすべてのスパインに対してファブリック外部接続を有効にする必要があります。新しいスパインを追加する場合は、**Setup Multipod** GUI ウィザードを使用します。

この問題を解決するには 2 つの方法があります。

- 外部ルーティング ネットワークの下ですべてのスパインを有効にします。
  - APIC GUI のメニュー バーで、**[テナント（Teant）]** > **[インフラ（infra）]** をクリックします。
  - **[Navigation（ナビゲーション）]** ペインで、**[ネットワーキング（Networking）]** > **[外部ルーテッド ネットワーク（External Routed Networks）]**を展開し、外部ルーテッド ネットワークを右クリックして、**[ファブリック外部接続を有効にする（Enable Fabric External Connectivity）]** を選択します。
- 外部ルーテッド ネットワークの下に新しいスパインを追加します。
  - APIC GUI のメニュー バーで、**[ファブリック（Fabric）]** をクリックします。

- **[ナビゲーション（Navigation）]** ペインで、**[クイック スタート（Quick Start）]** > **[ノードまたはポッド セットアップ（Node or Pod Setup）]** > **[マルチポッドのセットアップ（Setup Multipod）]** を展開し、マルチポッド セットアップを完了します。

# リモート リーフ構成の確認

リモート リーフ スイッチの直接通信を有効にしたら、次の手順を使用して構成を確認できます。

**ステップ 1** スイッチに SSH 接続します。

**ステップ 2** 直接通信が有効になっていることを確認します。

次の出力で、`rldirectMode` が *yes* に設定されていることを確認します。

```
remote-leaf-switch# cat /mit/sys/summary
# System
[...]
remoteNetworkId          : 0
remoteNode               : no
rlOperPodId              : 1
rlRoutableMode           : yes
rldirectMode             : yes
[...]
```

**ステップ 3** リモート リーフ スイッチが完全ルーティング可能モードであり、Cisco APIC のパブリック IP アドレスと通信していることを確認します。

a) `rlRoutableMode` が *yes* に設定されていることを確認します。

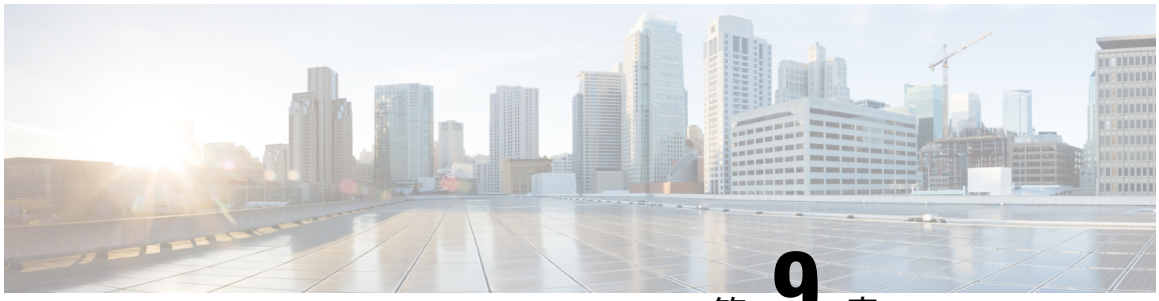
```
remote-leaf-switch# moquery -c topSystem | grep rlRoutableMode
rlRoutableMode           : yes
```

b) リモート リーフ スイッチから Cisco APIC ルーティング可能な IP アドレスに ping できることを確認します。

```
remote-leaf-switch# iping -V overlay-1 110.0.0.225

PING 110.0.0.225 (110.0.0.225) from 193.0.3.20: 56 data bytes

64 bytes from 110.0.0.225: icmp_seq=0 ttl=61 time=0.401 ms
```

c) リモート リーフ スイッチの `dhcpRespMo` が APIC のルーティング可能な IP アドレスに設定されていることを確認します。

```
remote-leaf-switch# moquery -c dhcpResp

serverId     : 110.0.0.225
siAddr       : 110.0.0.225
status       :
subnetMask   : 255.255.255.255
yiAddr       : 191.2.0.72
```



第 9 章

# NXOS ハードウェア テーブルの確認

この章は、次の項で構成されています。



## End Point Manager 学習の確認

次のコマンドを使用して、End Point Manager（EPM）の学習を確認します。

次の例では、コマンドを使用しすることで、ソース EP 172.17.15.105 が site5、leaf101 で検出されていることを確認できます。この出力は、EP 172.17.15.105 の場合、BD-VNID が 15794150、VRF-VNID が 2195456、pcTag または sclass が 16387 であることを示しています。

```
leaf101# show sys int epm end mac 0000.5555.1715

MAC : 0000.5555.1715 ::: Num IPs : 1
IP# 0 : 172.17.15.105 ::: IP# 0 flags :
Vlan id : 18 ::: Vlan vnid : 8393 ::: VRF name : msite-tenant-welkin:dev
BD vnid : 15794150 ::: VRF vnid : 2195456
Phy If : 0x1a000000 ::: Tunnel If : 0
Interface : Ethernet1/1
Flags : 0x80004c04 ::: sclass : 16387 ::: Ref count : 5
EP Create Timestamp : 07/30/2017 07:28:40.535135
EP Update Timestamp : 07/30/2017 08:05:56.769126
EP Flags : local|IP|MAC|sclass|timer|
::::


leaf101# show sys int epm end ip 172.17.15.106

MAC : 0000.6666.1716 ::: Num IPs : 1
IP# 0 : 172.17.15.106 ::: IP# 0 flags :
Vlan id : 9 ::: Vlan vnid : 8193 ::: VRF name : msite-tenant-welkin:dev
BD vnid : 16449430 ::: VRF vnid : 2326528
Phy If : 0x1a027000 ::: Tunnel If : 0
Interface : Ethernet1/40
Flags : 0x80005c04 ::: sclass : 16386 ::: Ref count : 5
EP Create Timestamp : 07/31/2017 05:15:24.179330
```

```
EP Update Timestamp : 08/01/2017 10:45:06.108770
EP Flags : local|IP|MAC|host-tracked|sclass|timer|
::::
```

# BGP EVPN ルーティング テーブルの確認

次のコマンドを使用して、BGP EVPN ルーティング テーブルを確認します。

この例では、leaf101 から発見されたエンド ポイント 172.17.15.105 が EPM（Endpoint manager）によって COOP を介してspine501 に公開されます。スパインで COOP プロセスを実行し、EP を L2vpn EVPN に同期します。コマンド出力は、EP 172.17.15.105 がサイト 5 に対してローカルであり、BGP EVPN によってサイト 6 にアドバタイズされていることを示しています。

```
spine501# show bgp l2vpn evpn 172.17.15.105 vrf overlay-1
Route Distinguisher: 1:99680230 (L2VNI 15794150)
BGP routing table entry for
[2]:[0]:[15794150]:[48]:[0000.5555.1715]:[32]:[172.17.15.105]/272, version 719 dest ptr
 0xab0a63de
MSITE RD: 1:99680230 (L2VNI 15794150)
Local Route Distinguisher: 5.5.5.4:65005 (L2VNI 1)
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
 5.5.5.4 (metric 0) from 0.0.0.0 (5.5.5.3)
 Origin IGP, MED not set, localpref 100, weight 32768
 Received label 15794150 2195456
 Extcommunity:
 RT:5:5

 Path-id 1 advertised to peers:
 6.6.6.3

Route Distinguisher: 5.5.5.4:65005 (L2VNI 1)
BGP routing table entry for
[2]:[0]:[15794150]:[48]:[0000.5555.1715]:[32]:[172.17.15.105]/272, version 719 dest ptr
 0xab0a63de
MSITE RD: 1:99680230 (L2VNI 15794150)
Local Route Distinguisher: 5.5.5.4:65005 (L2VNI 1)
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
 5.5.5.4 (metric 0) from 0.0.0.0 (5.5.5.3)
 Origin IGP, MED not set, localpref 100, weight 32768
 Received label 15794150 2195456
 Extcommunity:
 RT:5:5

 Path-id 1 advertised to peers:
 6.6.6.3
```

```
spine501# show bgp internal evi 15794150

**************************************************
Global EVI : 1
Number of EVI : 1
L2RIB bound / VNI Req to L2RIB : Yes / 1
VNI Adds / Dels from L2RIB : 9 / 6
Topo global/mpod/wan/avs/msite reg pending: 0/0/0/0/0
Topo global/mpod/wan/avs/msite registered: 1/0/0/0/1
L2RIB is up/registered/local-req: 1/1
L2RIB down: in-prg/up-defer: 0/0
L2RIB register/failures: 1/0
L2RIB deregister/failures: 0/0
L2RIB flow control (#enabled/#disabled): Disabled (0/0)
**************************************************
L2RIB Emulation Library Info
----------------------------
L2RIB Service BGP state UP BIND PEERBIND
Global EVI 134217729, MPOD SHARD shard [0, 0]
Global EVI 134217729, MSITE SHARD shard [0, 4294967295] --- --- The global EVI is same
for the identical across multi-sites.
Global EVI 134217729, GOLF SHARD shard [0, 0]
Global EVI 134217729, EXT_SRC SHARD shard [0, 0]
 MTS: total 1 bufs 1 free 0 full 0 working
 MTS TX: 408 (Fail 0) RX: 229
 MTS PAUSE: 0 (Flush Fail 0)
Peer service COOP state UP BIND PEERBIND
 BIND TX: 61 RX: 0
 REGISTER TX: 61 RX: 0
 TOPO TX: 0 RX: 21
 MAC TX: 395 RX: 208
 IP TX: 19 RX: 39
 IMET TX: 0 RX: 0
 SMAD TX: 0 RX: 0
Peer service ISIS state DOWN UNBIND PEERUNBIND
 BIND TX: 0 RX: 0
 REGISTER TX: 0 RX: 0
 TOPO TX: 0 RX: 0
 MAC TX: 0 RX: 0
 IP TX: 0 RX: 0
 IMET TX: 0 RX: 0
 SMAD TX: 0 RX: 0
**************************************************
BGP L2VPN/EVPN RD Information for 1:99680230
 L2VNI ID : 15794150 (vni_15794150)
 #Prefixes Local/BRIB : 2 / 4
 #Paths L3VPN->EVPN/EVPN->L3VPN : 0 / 0
**************************************************
==============================================
BGP Configured VNI Information:
 VNI ID (Index) : 15794150 (0)
 RD : 1:99680230
 Export RTs : 1
 Export RT cfg list: 65005:99680230(refcount:1
 Import RTs : 1
 Import RT cfg list: 65006:117112726(refcount:1
 Topo Id : 15794150
 VTEP IP : 0.0.0.0
 VTEP VPC IP : 0.0.0.0
 Enabled : Yes
 Delete Pending : No
 RD/Import RT/Export RT : Yes/Yes/Yes
 Type : 3
```

```
 Usage : 2
 L2 stretch enabled : 1
 VRF Vnid : 2195456
 Refcount : 00000003
 Encap : VxLAN

==========================================
++++++++++++++++++++++++++++++++++++++++++
BGP VNI Information for vni_15794150
 L2VNI ID : 15794150 (vni_15794150)
 RD : 1:99680230
 VRF Vnid : 2195456
 Prefixes (local/total) : 2/4
 VNID registered with COOP : Yes
 Enabled : Yes
 Delete pending : 0
 Stale : No
 Import pending : 0
 Import in progress : 0
 Encap : VxLAN
 Topo Id : 15794150
 VTEP IP : 0.0.0.0
 VTEP VPC IP : 0.0.0.0
 Active Export RTs : 1
 Active Export RT list : 65005:99680230
 Config Export RTs : 1
 Export RT cfg list: 65005:99680230(refcount:1
 Export RT chg/chg-pending : 0/0
 Active Import RTs : 1
 Active Import RT list : 65006:117112726
 Config Import RTs : 1
 Import RT cfg list: 65006:117112726(refcount:1
 Import RT chg/chg-pending : 0/0
 IMET Reg/Unreg from L2RIB : 1/0
 MAC Reg/Unreg from L2RIB : 1/0
 MAC IP Reg/Unreg from L2RIB : 1/0
 IP-only Reg/Unreg from L2RIB : 0/0
 SMAD Reg/Unreg from L2RIB : 1/0
 IMET Add/Del from L2RIB : 0/0
 MAC Add/Del from L2RIB : 97/96
 MAC IP Add/Del from L2RIB : 3/2
 SMAD Add/Del from L2RIB : 0/0
 IMET Dnld/Wdraw to L2RIB : 0/0
 IMET Dnld/Wdraw to L2RIB failures : 0/0
 MAC Dnld/Wdraw to L2RIB : 190/189
 MAC Dnld/Wdraw to L2RIB failures : 0/0
 SMAD Dnld/Wdraw to L2RIB : 0/0
 SMAD Dnld/Wdraw to L2RIB failures : 0/0
 MAC-IP/SMAD Msite-RD routes : 4
 MAC-IP WAN-RD routes : 0
 MAC-IP network host routes : 0
 Type : 3
```

# VNID、S クラス、および VTEP マッピングの確認

次のコマンドを使用して、リモート サイト ID を確認します。

サイト間の VNID および pcTag または S クラス変換の場合、変換は宛先サイトから検証する必要があります。たとえば、パケットが site5 から site6 に送信されている場合、変換は site6 のス

パインによって行われます。翻訳が site5 にプッシュされているかどうかを確認するには、次のコマンドを使用します。

```
spine501# show dcimgr repo eteps

Remote site=6 :
Rem Etep=6.6.6.1/32, is_ucast=yes
Rem Etep=6.6.6.2/32, is_ucast=no
```

次のコマンドを使用して、リモート サイトとローカル サイト間の sclass-map を確認します。

```
spine501# show dcimgr repo sclass-maps
------------------------------------------------------------
     Remote            |        Local
site  Vrf       PcTag  |  Vrf      PcTag     Rel-state
------------------------------------------------------------
  6   2326528   32770  |  2195456   49154   [formed]
  6   2326528   16386  |  2195456   16387   [formed]
```

次のコマンドを使用して、リモート サイトとローカル サイト間の VRF または BD VNID マップを確認します。

```
spine501# show dcimgr repo vnid-maps detail
------------------------------------------------------------------
     Remote             |           Local
site  Vrf       Bd      |    Vrf      Bd       Rel-state
------------------------------------------------------------------
   6  2326528           |  2195456            [formed]
      0x238000          |  0x218000
------------------------------------------------------------------
   6  2326528  16449430 |  2195456  15794150  [formed]
      0x238000 0xfaff96 |  0x218000 0xf0ffe6
------------------------------------------------------------------

spine501# show dcimgr repo vnid-maps verbose

Local site=5  Remote site=6:
Loc vrfvnid=2195456 Rem vrfvnid=2326528 rel-state=formed
  BD Vnids:
  Loc vnid=15794150 Rem vnid=16449430 rel-state=formed
```

次のコマンドを使用して、DCI HAL オブジェクトを確認します。

```
module-1# show plat int hal objects dci all
Dumping dci objects
## Get Objects for dci remotesite for Asic 0

  OBJECT 0:
Handle                                                  : 23967
siteid                                                  : 0x6
iswan                                                   : Disabled

## Get Objects for dci remotesiteetep for Asic 0

  OBJECT 0:
Handle                                                  : 23970
ucastetep                                               : 6.6.6.1/32
siteid                                                  : 0x6
```

```
## Get Objects for dci vnidmap for Asic 0

  OBJECT 0:
Handle                                                  : 23977
isbdvnid                                                : Disabled
localvnid                                               : 0x218000
localgipo                                               : 0.0.0.0/32
remotevnid                                              : 0x238000
remotevrfvnid                                           : 0x238000
islocalbdctrl                                           : Disabled
siteid                                                  : 0x6


  OBJECT 1:
Handle                                                  : 23980
isbdvnid                                                : Enabled
localvnid                                               : 0xf0ffe6
localgipo                                               : 225.0.225.160/32
remotevnid                                              : 0xfaff96
remotevrfvnid                                           : 0x238000
islocalbdctrl                                           : Enabled
siteid                                                  : 0x6

## Get Objects for dci remotevrfvnid for Asic 0

  OBJECT 0:
Handle                                                  : 23972
remotevnid                                              : 0x238000
siteid                                                  : 0x6

## Get Objects for dci sclassmap for Asic 0

  OBJECT 0:
Handle                                                  : 23986
localsclass                                             : 0x4003  //18387
remotesclass                                            : 0x4002  //16386
remotevnid                                              : 0x238000
siteid                                                  : 0x6


  OBJECT 1:
Handle                                                  : 23974
localsclass                                             : 0x1
remotesclass                                            : 0x1
remotevnid                                              : 0x238000
siteid                                                  : 0x6


  OBJECT 2:
Handle                                                  : 23983
localsclass                                             : 0xc002  //49154
remotesclass                                            : 0x8002  //32770
remotevnid                                              : 0x238000
siteid                                                  : 0x6
```

次のコマンドを使用して、VXLAN オブジェクトを確認します。

```
module-1# show plat int hal objects vxlan mytep  | egrep -B 13 -A 8 5.5.5.1


 OBJECT 11:
Handle                                                  : 23964
useforvpc                                               : Disabled
```

```
usefornonvpc                                       : Disabled
useforvteps                                        : Disabled
proxyforv4                                         : Disabled
proxyforv6                                         : Disabled
isdciucastetep                                     : Enabled
isdcimcastetep                                     : Disabled
isdcieteplocal                                     : Enabled
proxyformac                                        : Disabled
outerbdid                                          : 0x2
rmac                                               : 00:0d:0d:0d:0d:0d
csouterbdid                                        : 0x1
address                                            : 5.5.5.1/32
encaptype                                          : iVxlan
id                                                 : 0x1400000b
lid                                                : 0x0
iftype                                             : none
ifname                                             : Lo11
Relation Object ipteptovrf :
  rel-ipteptovrf-l3-l3_vrf-handle                  : 6983
  rel-ipteptovrf-l3-l3_vrf-id                      : 0x2
```

```
module-1# show plat int hal objects vxlan remotetep  | egrep -B 28 -A 8
6.6.6.1

  OBJECT 1:
Handle                                             : 25810
operst                                             : up
enablelearning                                     : Disabled
enablebindlearn                                    : Disabled
enablesclasslearn                                  : Disabled
drop                                               : Disabled
isvpcpeer                                          : Disabled
islocal                                            : Disabled
proxyforv4                                         : Disabled
proxyforv6                                         : Disabled
proxyformac                                        : Disabled
unicastreplication                                 : Disabled
splithorizongroupid                                : 0x0
trustqosmarking                                    : Disabled
trustsclass                                        : Disabled
trustlb                                            : Disabled
trustdl                                            : 0x1
istepscale                                         : Disabled
usedfhash                                          : Disabled
dismark                                            : Disabled
hwencapidx                                         : 0x5
srcnat                                             : Disabled
isingressonly                                      : Disabled
isipn                                              : Disabled
isdciucastetep                                     : Enabled
isdcimcastetep                                     : Disabled
address                                            : 6.6.6.1/32
encaptype                                          : iVxlan
id                                                 : 0x18010004
lid                                                : 0x0
iftype                                             : none
ifname                                             : Tunnel4
Relation Object ipteptovrf :
  rel-ipteptovrf-l3-l3_vrf-handle                  : 6983
  rel-ipteptovrf-l3-l3_vrf-id                      : 0x2
```

# LC ハードウェア テーブルの確認

次のコマンドを使用して、LC ハードウェア テーブルを確認します。

```
module-1# show platform internal hal dci sclassmap
Non-Sandbox Mode

Sandbox_ID: 0  Asic Bitmap: 0x0

                      --- DCI Sclass table ---
Site  Remote   Local   Remote    Remote   Local
 ID    Vnid    Sclass  Sclass    Sclass   Sclass  Scope
----+--------+--------+--------+--------+-------
6    2326528     16387     16386      16386       16387      1
6    2326528     1         1          1           1          1
6    2326528     49154     32770      32770       49154      1
next asic
```

```
module-1# show platform internal hal dci vnidmap

Non-Sandbox Mode

Sandbox_ID: 0  Asic Bitmap: 0x0

Site  POD  isBD  Local Remote  ----EPG table----    BD State Table
 ID   ID         vnid   vnid   idx     Localvnid     idx     isBD
--------+-+-+-+----+--------+--------+----+-+-+-+-+----+--+-+----
6   1     0   2195456    2326528  15360     2195456       15360     0
6   1     1   15794150   16449430  15361     15794150       15361     1
```

```
module-1# show platform internal sug tile-table dci-sclass

================================================================================================================

                                    SLICE : 0                                FP : 6
                  TILE : 0
================================================================================================================
ENTRY[001208] = tile_entry_dci_sclass_entry_0_key_valid=0x1
                tile_entry_dci_sclass_entry_0_key_scope=0x1
                tile_entry_dci_sclass_entry_0_key_sclass_in=0x1
                tile_entry_dci_sclass_entry_0_data_sclass_out=0x1
ENTRY[001645] = tile_entry_dci_sclass_entry_0_key_valid=0x1
                tile_entry_dci_sclass_entry_0_key_scope=0x1
                tile_entry_dci_sclass_entry_0_key_sclass_in=0x4002
                tile_entry_dci_sclass_entry_0_data_sclass_out=0x4003
ENTRY[001713] = tile_entry_dci_sclass_entry_0_key_valid=0x1
                tile_entry_dci_sclass_entry_0_key_scope=0x1
                tile_entry_dci_sclass_entry_0_key_sclass_in=0x8002
                tile_entry_dci_sclass_entry_0_data_sclass_out=0xc002

================================================================================================================

                                    SLICE : 1                                FP : 6
                  TILE : 0
================================================================================================================
ENTRY[001208] = tile_entry_dci_sclass_entry_0_key_valid=0x1
                tile_entry_dci_sclass_entry_0_key_scope=0x1
                tile_entry_dci_sclass_entry_0_key_sclass_in=0x1
```

```
                tile_entry_dci_sclass_entry_0_data_sclass_out=0x1
ENTRY[001645] = tile_entry_dci_sclass_entry_0_key_valid=0x1
                tile_entry_dci_sclass_entry_0_key_scope=0x1
                tile_entry_dci_sclass_entry_0_key_sclass_in=0x4002
                tile_entry_dci_sclass_entry_0_data_sclass_out=0x4003
ENTRY[001713] = tile_entry_dci_sclass_entry_0_key_valid=0x1
                tile_entry_dci_sclass_entry_0_key_scope=0x1
                tile_entry_dci_sclass_entry_0_key_sclass_in=0x8002
                tile_entry_dci_sclass_entry_0_data_sclass_out=0xc002
```

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。