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

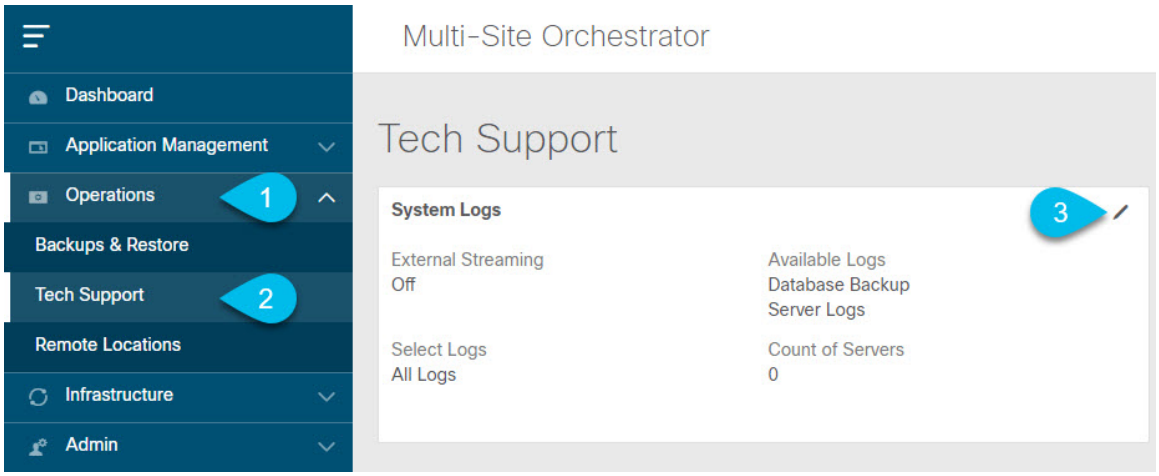
この章は、次の項で構成されています。

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。

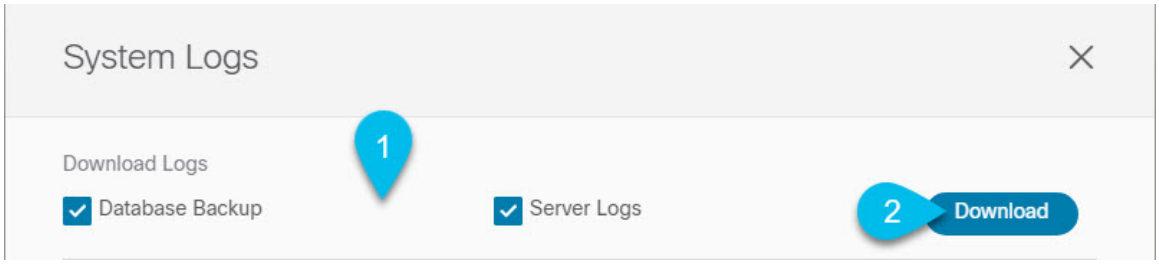
ステップ 2 **[システムログ (System Logs)]** 画面を開きます。



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 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 *1*: サイト間 *ARP* フロー

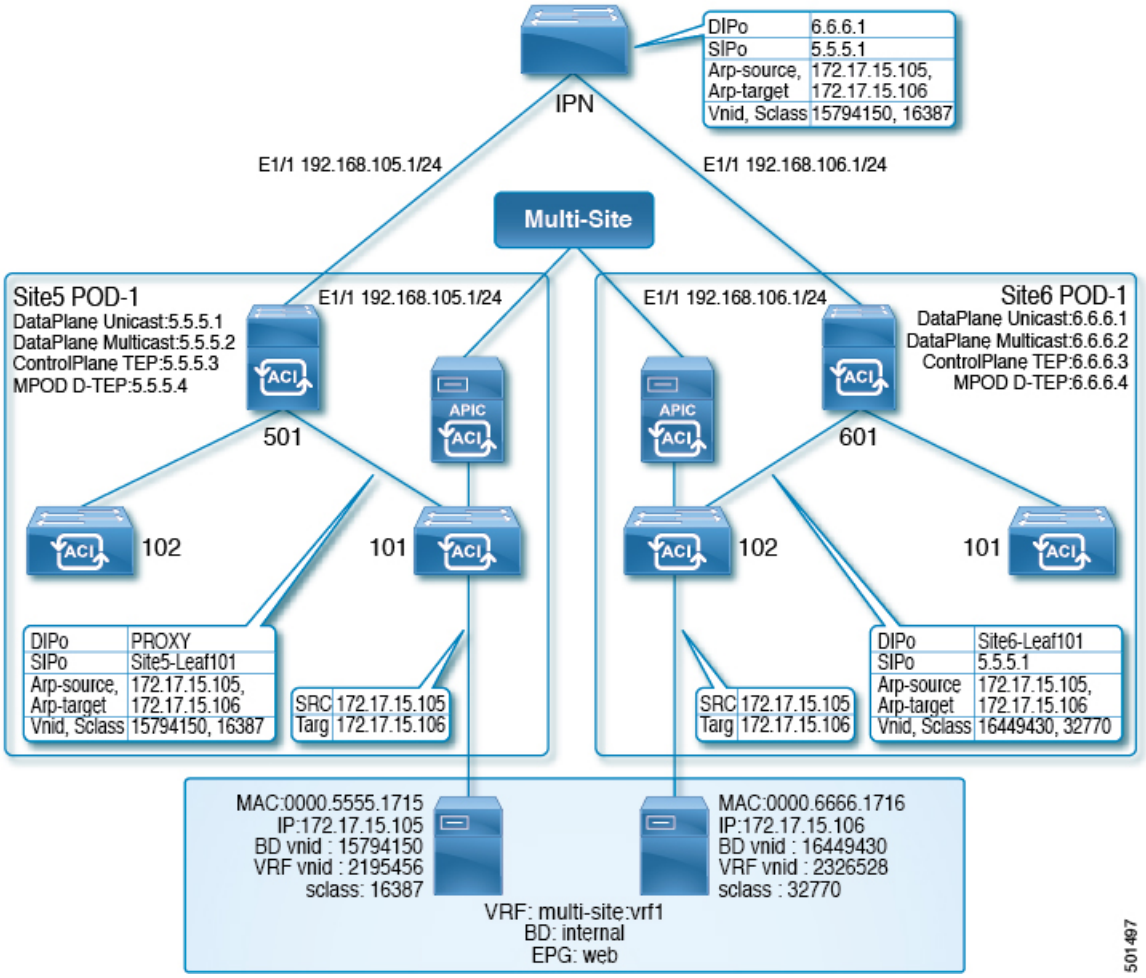
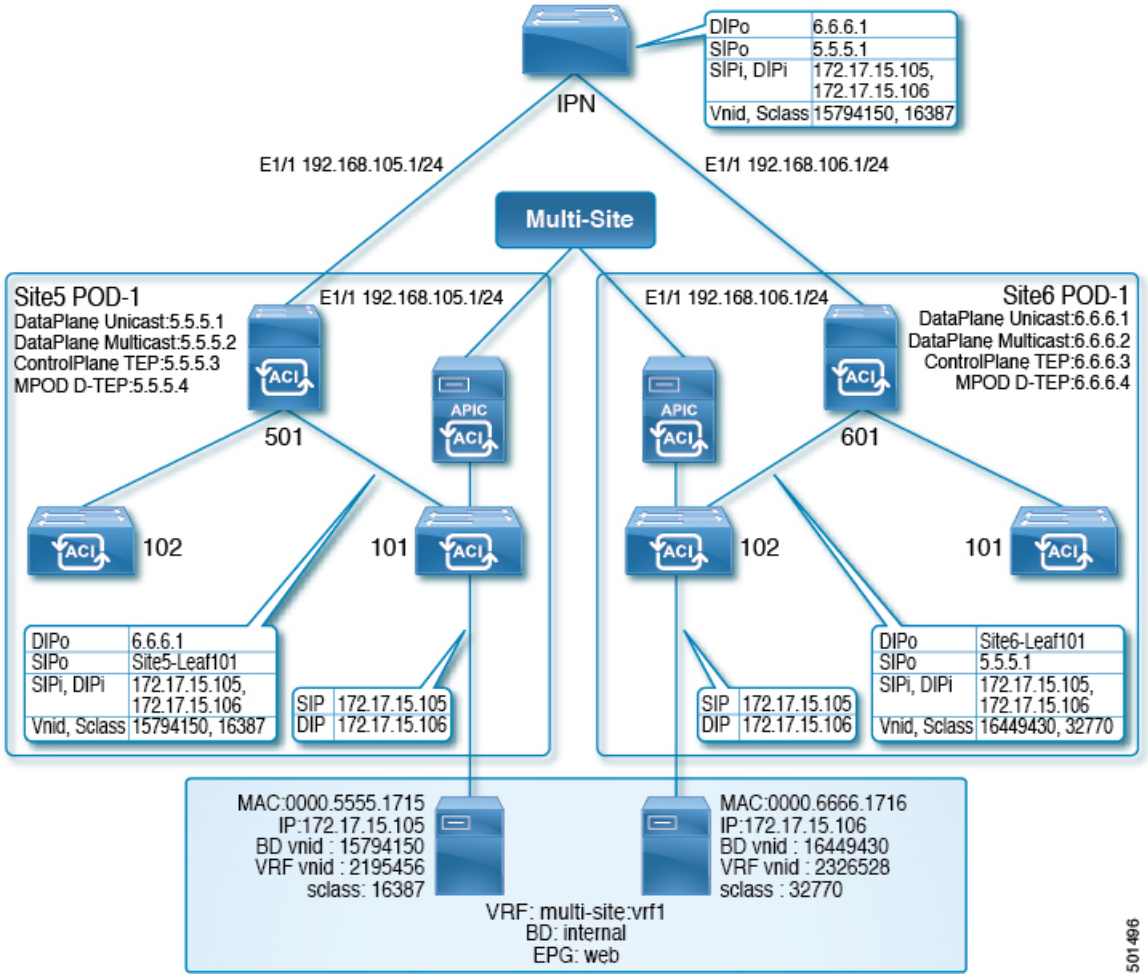


図 2 は、ユニキャスト パケット フローに焦点を当てた同じ拡張 BD ネットワークを示しています。

図 *2:* 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
```

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。