



トラブルシューティング ツール

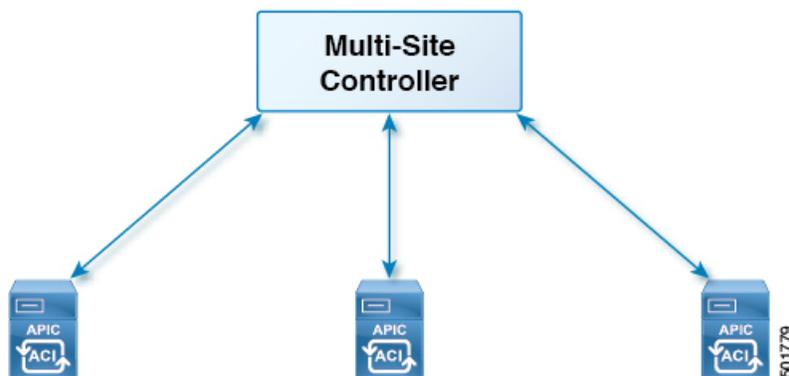
この章は、次の項で構成されています。

- [整合性チェッカーの概要](#) (1 ページ)
- [システム ログのダウンロード](#) (6 ページ)
- [Docker コンテナ情報の収集](#) (8 ページ)
- [API コール ログの生成](#) (10 ページ)
- [実行ログの読み取り](#) (11 ページ)
- [APIC サイトでのポリシー解決の確認](#) (12 ページ)

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：

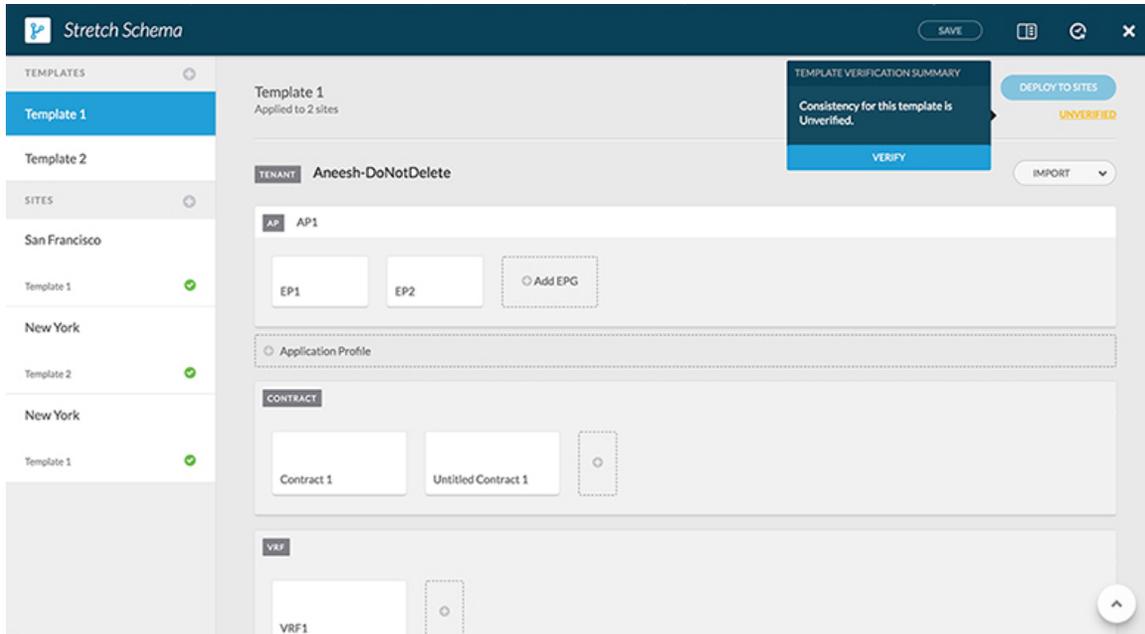
- EPG
- VRF
- BD
- 外部 EPG

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：

POLICY	VERIFICATION	NEW YORK	SAN FRANCISCO
BD1	APIC	✓	✓
	Switch	✗	✗
EP1	APIC	✓	✓
	Switch	✗	✗
EP2	APIC	✓	✓
	Switch	✗	✗
VRF1	APIC	✓	✓
	Switch	✗	✗

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。
- [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。

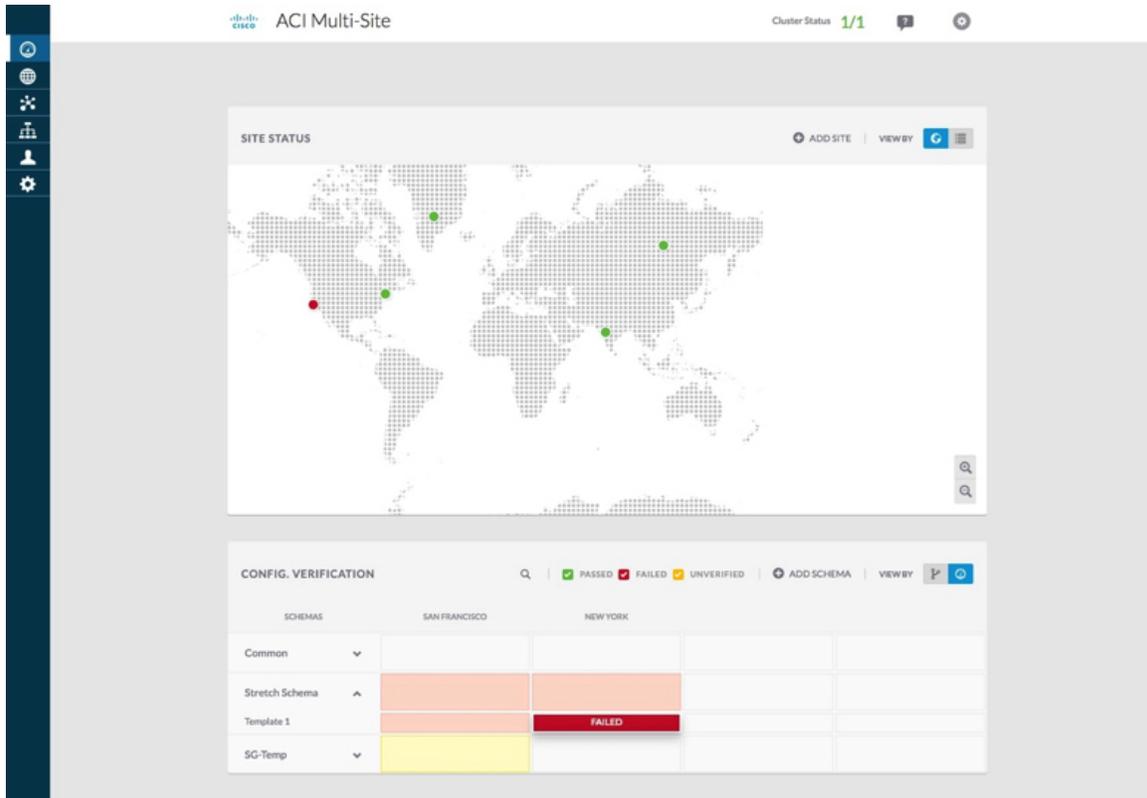
例：

Template1 New York		VERIFICATION FAILED		Last Verified: 03/12/18 1:14 pm	
POLICY	VERIFICATION	NEW YORK		SAN FRANCISCO	
BD1	APIC	✓		✓	
	Switch	✗		✗	
EP1	APIC	✓		✓	
	Switch	✗		✗	
EP2	APIC	✓		✓	
	Switch	✗		✗	
VRF1	APIC	✓		✓	
	Switch	✗		✗	

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。

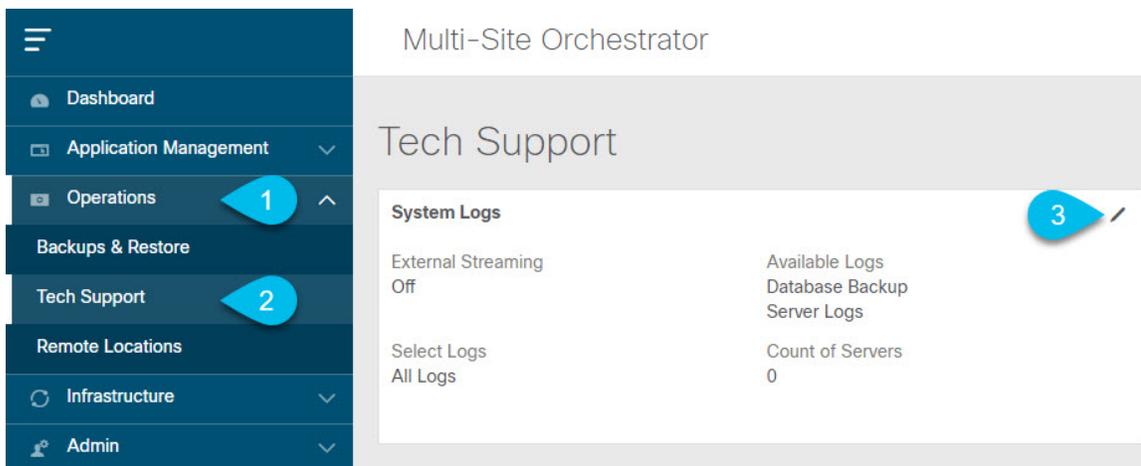
ステップ8 (オプション) EPG、BD、VRF、または外部EPGで検索して、このポリシーが含まれているスキーマを見つけることができます。

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。

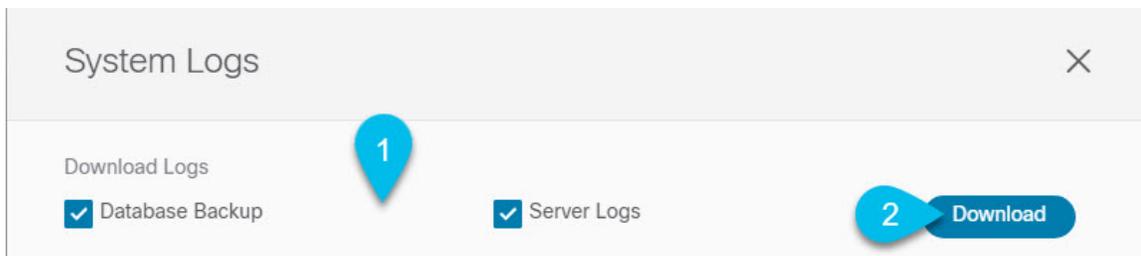
ステップ1 マルチサイト Orchestrator GUI にログインします。

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  [...]
ve5m91wb1qc4                     msc_auditsevice                   replicated  1/1        [...]
bl0op2eli7bp                      msc_authyldapsevice               replicated  1/1        [...]
uxc6pgzficls                      msc_authytacacssevice             replicated  1/1        [...]
qcws6ta7abwo                      msc_backupsevice                  global     3/3        [...]
r4p3opyf5dkm                      msc_cloudsevice                   replicated  1/1        [...]
xrm0c9vof3r8                     msc_consistencysevice              replicated  1/1        [...]
le4gy9kov7ey                      msc_endpointsevice                replicated  1/1        [...]
micd93h5gj97                     msc_executionengine                replicated  1/1        [...]
6wxh4mgnnfi9                     msc_jobschedulersevice             replicated  1/1        [...]
lrjl764xw91g                      msc_kong                           global     3/3        [...]
n351htjnks75                     msc_kongdb                         replicated  1/1        [...]
xcikdp9o3i6                      msc_mongoddb1                     replicated  1/1        [...]
u9b9ihxxnzt                      msc_mongoddb2                     replicated  1/1        [...]
m0byoou6zuv5                     msc_mongoddb3                     replicated  1/1        [...]
logqawe8k3cg                     msc_platformsevice                global     3/3        [...]
m3sxf06odn74                     msc_schemasevice                  global     3/3        [...]
3wd4zrqf6kbb                     msc_sitesevice                    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-cloudsevice:2.1.2g              "bin/cloudsevice"                       [...]
685f54b70a0d   msc-executionengine:2.1.2g         "bin/executionengine"                  [...]
0c719107adce   msc-schemasevice:2.1.2g             "bin/schemasevice"                     [...]
f2e3d144738c   msc-userservice:2.1.2g              "bin/userservice"                       [...]
edd0d4604e27   msc-syncengine:2.1.2g               "bin/syncengine"                       [...]
001616674a00   msc-sitesevice:2.1.2g               "bin/sitesevice"                       [...]
7b30c61f8aa7   msc-platformsevice:2.1.2g          "bin/platformsevice"                   [...]
d02923992d77   msc-backupsevice:2.1.2g             "bin/backupsevice"                     [...]
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 \(6 ページ\)](#) 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.nv4j5uj5786yj621wjxsxvxml
00f627c6804c  msc-platformservice:latest        "bin/platformservice"    2 weeks ago   Up 2 weeks (healthy)
9050/tcp      msc_platformservice.1.fw58j62dfcme4noh67am0s73
```

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.LightArrayRevolverScheduler$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"
# 12.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]]
# 12.LocalBdSubstitute
dn: sys/inst-overlay-1/localSite-5/localCtxSubstitute-[vxlan-2195456]/localBdSubstitute-
[vxlan-15794150]
# 12.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]
# 12.LocalPcTagSubstitute
dn: sys/inst-overlay-1/localSite-5/localCtxSubstitute-[vxlan-2195456]/localPcTagSubstitute-
49154# 12.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]
# 12.LocalPcTagSubstitute
dn: sys/inst-overlay-1/localSite-5/localCtxSubstitute-[vxlan-2195456]/localPcTagSubstitute-
16387# 13.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]]
# 13.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
# 12.RsToLocalBdSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remoteBdSubstitute-
[vxlan-16449430]/rsToLocalBdSubstitute
# 12.RemoteBdSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remoteBdSubstitute-
```

```
[vxlan-16449430]
# 12.RsToLocalPcTagSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remotePcTagSubstitute-32770/rsToLocalPcTagSubstitute
# 12.RemotePcTagSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remotePcTagSubstitute-32770# 12.RsToLocalPcTagSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remotePcTagSubstitute-16386/rsToLocalPcTagSubstitute
# 12.RemotePcTagSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/remotePcTagSubstitute-16386# 13.RsToLocalCtxSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]/rsToLocalCtxSubstitute
# 13.RemoteCtxSubstitute
dn: sys/inst-overlay-1/remoteSite-6/remoteCtxSubstitute-[vxlan-2326528]
# dci.RemoteSite
dn: sys/inst-overlay-1/remoteSite-6
```

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。