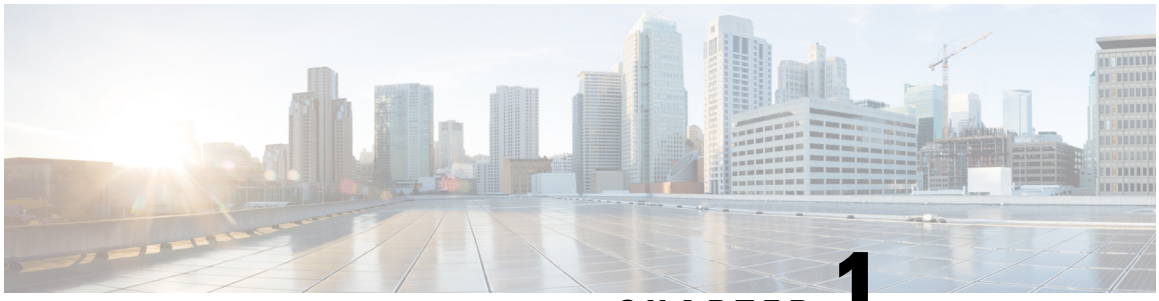
Cisco ACI Multi-Site Troubleshooting Guide, Release 3.1(x)

First Published: 2020-05-11

Last Modified: 2020-05-11

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

New and Changed Information

This chapter contains the following sections:

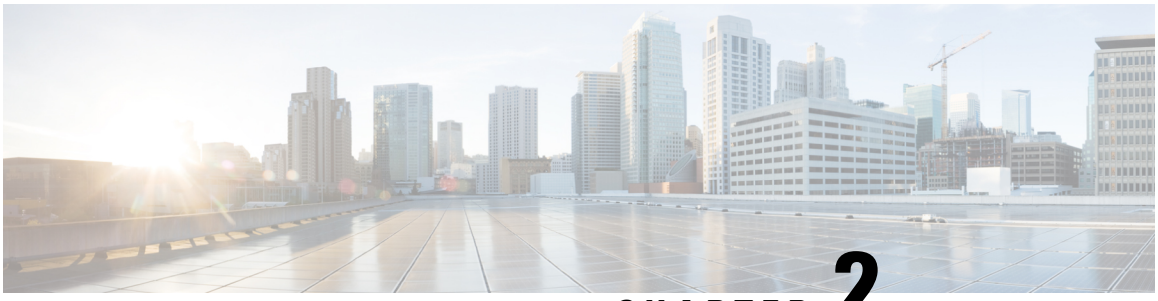
New and Changed Information

The following table provides an overview of the significant changes to the organization and features in this guide from the release the guide was first published to the current release. The table does not provide an exhaustive list of all changes made to the guide.

Table 1: Latest Updates

Release	New Feature or Update	Where Documented
3.0(1)	--	--

New and Changed Information



CHAPTER 2

Overview and Troubleshooting Basics

This chapter contains the following sections:

Topics Covered in This Guide

The chapters in this guide describe troubleshooting tools and tips that can be used to resolve common Cisco ACI Multi-Site Orchestrator issues. The topics covered in each chapter are summarized below.

Troubleshooting Tools—Provides information about the following:

- Explains how to use the Multi-Site troubleshooting tools to generate a troubleshooting report, generate API call logs, log on to a VM for data collection, read the execution log, verify that microservices are active, and verify policy resolution on Cisco APIC sites.
    - Consistency checker
    - Generating API call logs
    - Docker Container Information
    - Execute logs
    - Multi-Site microservices
    - APIC policy resolution

Installations, Upgrades, and Reboots—Provides information about the following:

- Restarting, replacing, or relocating Orchestrator nodes
- Backing up and restoring MongoDB
- Configuring NTP settings after initial installation
- Changing the Orchestrator secret and key files

Users—Provides information about troubleshooting user authentication issues.

Platform Health—Provides information about the following:

- Generating and downloading troubleshooting reports
- Inspecting Docker services
- Resolving missing Node label issues
- Inter-site traffic flow and BGP sessions
- Unicast and Multicast traffic failures

Tenants and Schemas—Provides information about the following:

- Policy deployment errors
- Tenant policy reports using REST API
- Undeploying templates and schemas

Multipod and Multi-Site—Provides information about troubleshooting Multi-Pod and Multi-Site issues.

Verifying NX-OS Hardware Tables—Provides information about the following:

- Endpoint manager learning
- BGP EVPN routing tables
- VNID, S-Class, and VTEP mappings
- Line card hardware tables

Orchestrator GUI Not Loading

In certain cases because of how typical browsers cache information, the Orchestrator GUI may not come up fully and continue to display the loading progress screen.

Step 1 Clear browser cache.

Specific steps differ based on the type of browser you're using.

Step 2 Reload the Orchestrator GUI.

Troubleshooting Basics

This section describes the first steps to take when you encounter an issue working with Multi-Site. Other chapters in this guide describes issues related to one or more specific features.

Before you begin

Become familiar with the tools listed in Troubleshooting Tools, on page 7.

Step 1 Determine if the issue is related to Multi-Site.

If you are havving issues with Multi-Site Orchestrator, first check the following things to determine if the issues are Multi-Site related. If the answer is no to one of the questions, the issue might be Multi-Site related. If all the answers are no, then it could be related to APIC, a switch, the intersite network, or the WAN.

a) Is Multi-Site accessible?

b) If traffic is not flowing...

Generate the APIC policy report as described in Generating a Tenant Policy Report Using the REST API, on page 53. Then verify the following:

- Are all the expected MOs deployed to the APIC sites?
- Do all the expected MOs have the correct property values on the APIC sites?
- Do the VRFs, BDs, EPGs, and L3InstPs have the correct mappings on all the sites?
- Do the EPGs have the correct peer context Dn?

c) If traffic is flowing, but should not...

Gather the policy resolution information as described in Verifying Policy Resolution on APIC Sites, on page 16. Then verify the following:

- Are all the MOs expected not to be there, actually not deployed to the APIC sites?
- Do all the related MOs have the correct property values on the APIC sites?

Step 2 Determine which part of Multi-Site has an issue.

a) Check that all Docker services are up and running.

For more information, see Gathering Docker Container Information, on page 12.

b) Check there are no errors in the execution log

For more information, see Reading the Execution Log, on page 15.

c) Check the APIC policy report for any issues.

For more information, see Generating a Tenant Policy Report Using the REST API, on page 53.

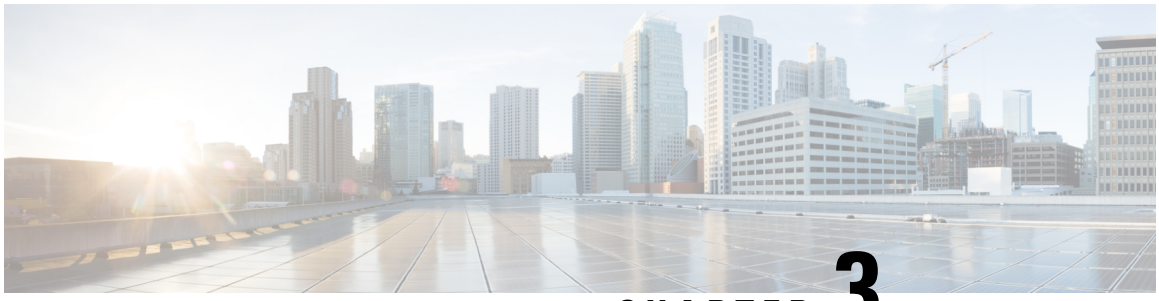
d) Check for connectivity issues

If you see connectivity issue in the Dashboard tab of the GUI, check one of the following:

- If you see the `No sites configured with BGP peering` or `BGP Session Failed` error, see Troubleshooting Inter-Site BGP Sessions, on page 45.
- If you see the `Unicast/Multicast Failure`, see Troubleshooting Unicast or Multicast Traffic Failures, on page 46 or Troubleshooting Multi-Site Multicast Functionality, on page 47

Step 3 Redeploy Schemas and Templates.

After you identify and correct any issues related to schemas or templates, undeploy and redeploy them as described in Undeploying Schemas and Templates, on page 54.



CHAPTER 3

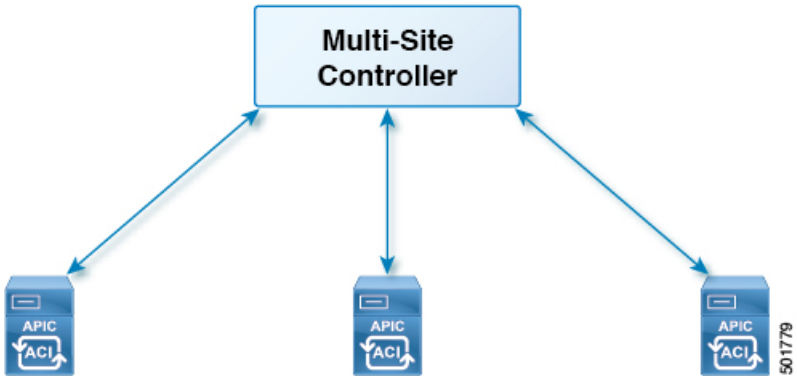
Troubleshooting Tools

This chapter contains the following sections:

Consistency Checker Overview

The Consistency Checker verifies deployments after the initial deploy operation, and integrates the results of this tool within the Cisco ACI Multi-Site user interface. This feature verifies cross mappings. Only usable on a template that has been deployed, that is stretched across at least two sites and contains at least one of the following policies:

- EPG
- VRF
- BD
- External EPG



Verifying a Template that has Been Deployed Across Sites

This section describes how to verify a template that has been deployed across sites.

Before you begin

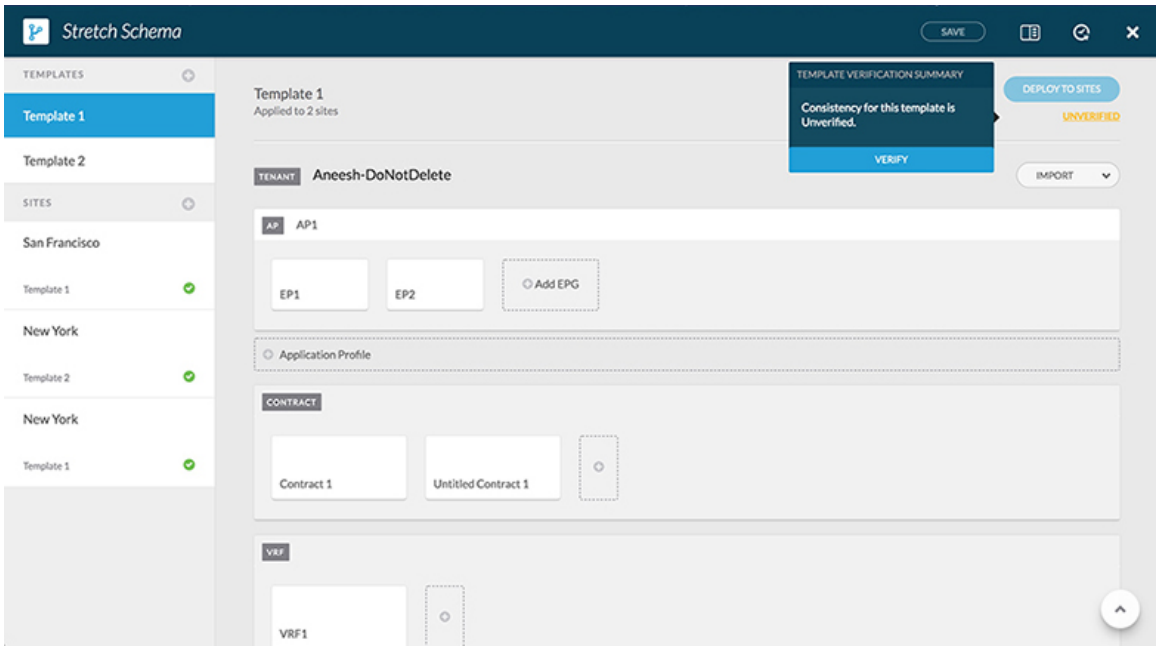
- The template that has been depolyed across at least two stretched sites and contains at least one of the following policies:
  - EPG
  - VRF
  - BD
  - External EPG

Step 1 Log in to the Multi-Site GUI.

Step 2 In the **Main Menu**, click **Schemas**, and on the Schema List page, choose the appropriate *schema_name*.

Step 3 Click on a deployed template.

Step 4 In the top right corner, click on **unverified**.



Step 5 In the **TEMPLATE VERIFICATION SUMMARY** dialog box, click **VERIFY**.

A popup message appears:

Consistency verification has been successfully triggered.

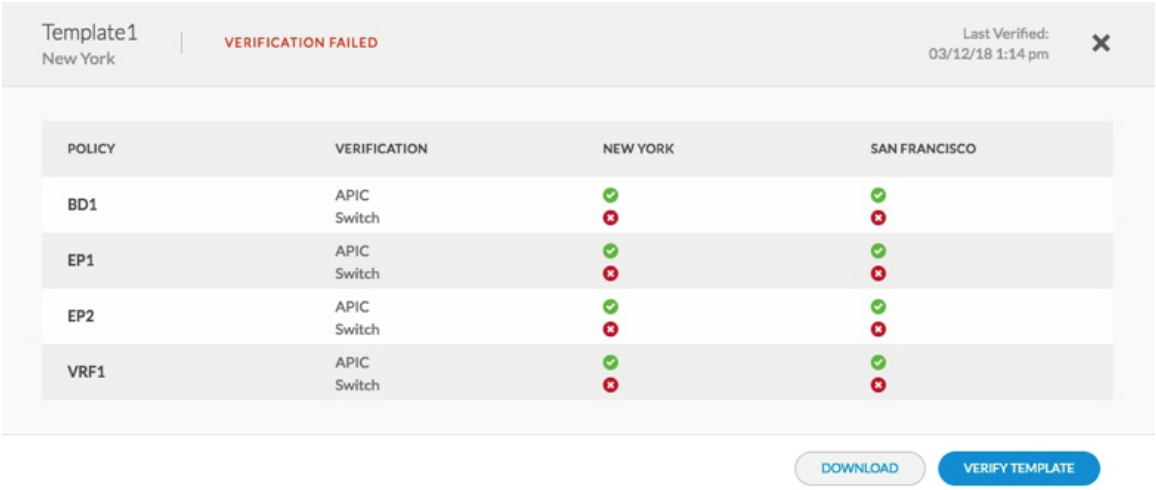
Step 6 The verification status will either be:

- **VERIFICATION SUCCESSFUL**—No action is needed.

- **VERIFICATION FAILED**—Action is needed.

a) If the verification failed, click **VERIFICATION FAILED**.
b) In the **TEMPLATE VERIFICATION SUMMARY** dialog box, for the site(s) that did fail, click on the pencil icon for a more detailed report of the template.

Example:



Hover over the red **x** for the description of the issue. The issue can either be **Not Found** (unable to locate) or **Mismatch** (misconfigured).

c) You can either click **DOWNLOAD** or **VERIFY TEMPLATE**.
   - **DOWNLOAD**—Provides you the report for only the current site.
   - **VERIFY TEMPLATE**—Provides you the verified template across all sites.

Setting Up a Scheduled Verification for Every Deployed Template

This section describes how to set up a scheduled verification for every deployed template on a per tenant basis.

Step 1 Log in to the Multi-Site GUI.
Step 2 In the **Main Menu**, click **Tenant**, and on the Tenant List page, click **Set Schedule** for the appropriate *tenant_name*.
Step 3 In the **Consistency Checker Scheduler Settings**, uncheck the **Disabler Schedule**, select the time and frequency.
a) Click **OK**.

Troubleshooting an Error

This section describes how to troubleshoot an error.

Step 1 Log in to the Multi-Site GUI.

Step 2 In the **Dashboard**, in the **SCHEMA HEALTH** section, in the view by field, click on the schema verification icon.

The small squares in a site represents the templates within the schema.

At a first glance, you can see what has passed, failed, or is unverified.

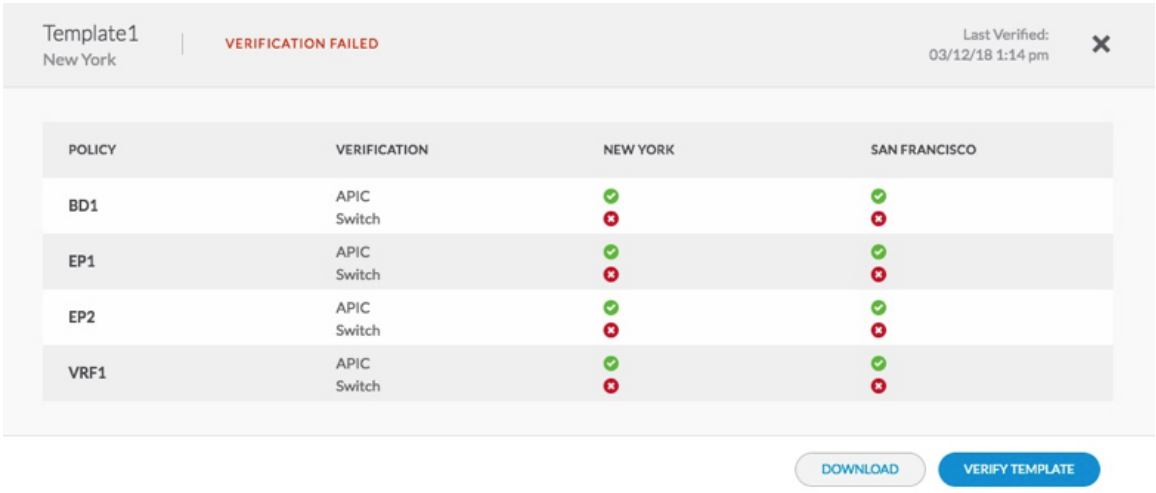
- **PASSED**—is in green.
- **FAILED**—is in red.
- **UNVERIFIED**—is in yellow.

Step 3 Expand the schema that contains a site in red to show the templates.

Step 4 If you hover over the red sites, it displays **FAILED**.

Step 5 You can click on the FAILED site and it will bring up a more detailed report.

Example:

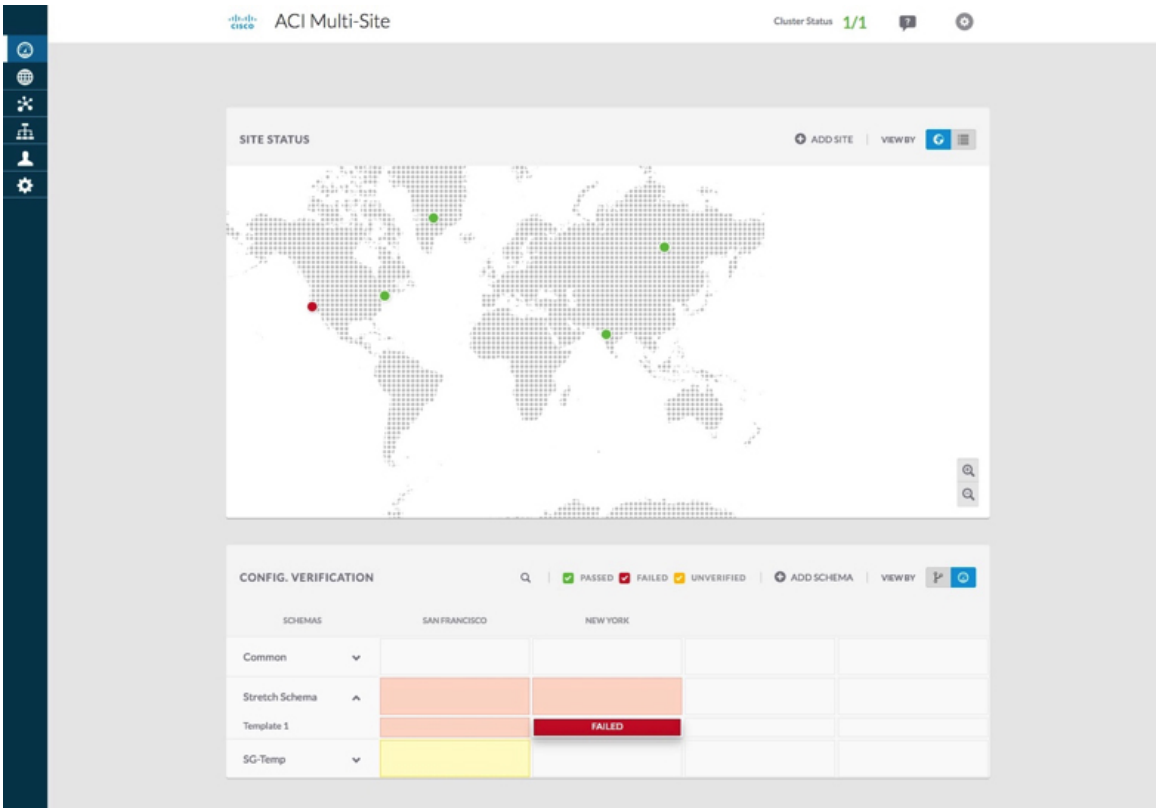


If you hover over the red **x** for the description of the issue. The issue can either be **Not Found** (unable to locate) or **Mismatch** (misconfigured).

a) You can either click **DOWNLOAD** or **VERIFY TEMPLATE**.

- **DOWNLOAD**—Provides you the report for only the current site.
- **VERIFY TEMPLATE**—Provides you the verified template across all sites.

Step 6 You can also see which templates passed, failed or are unverified.



Step 7 (Optional) You can verify the entire schema, click on the **…** and choose **Verify Schema**.

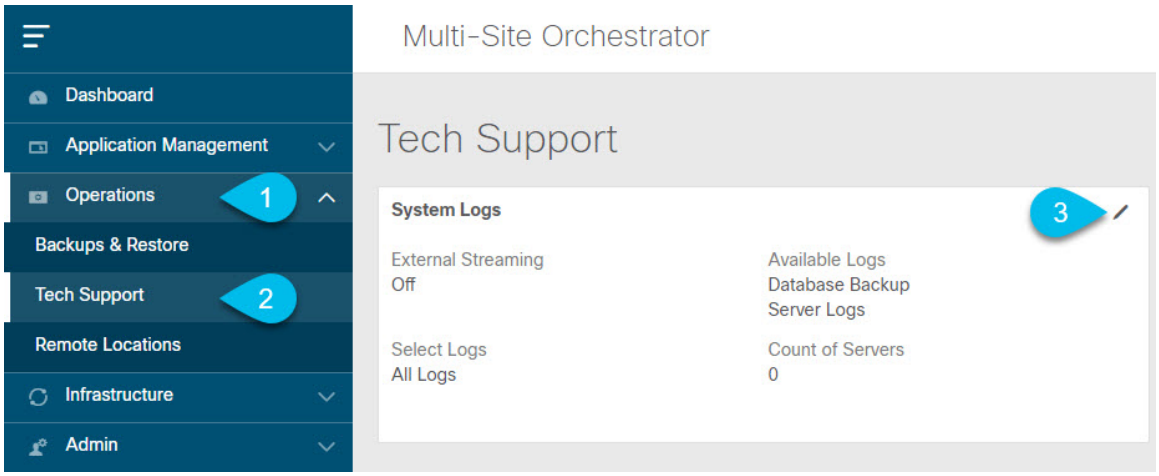
Step 8 (Optional) You can search by EPG, BD, VRF, or External EPG to find out which schema contains this policy.

Downloading System Logs

This section describes how to generate a troubleshooting report and infrastructure logs file for all the schemas, sites, tenants, and users that are managed by Cisco ACI Multi-Site Orchestrator.

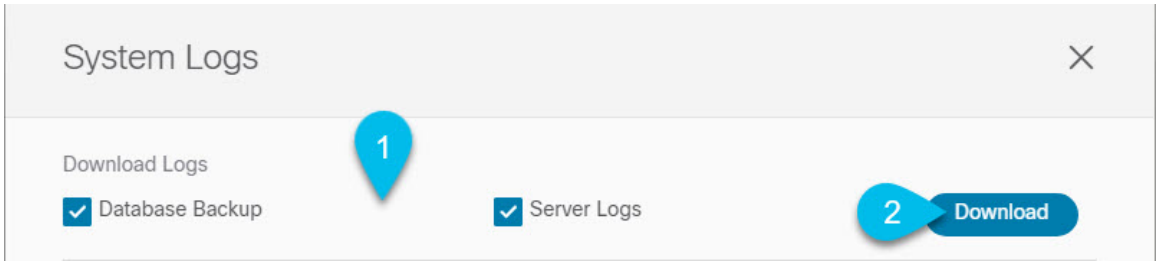
Step 1 Log in to your Multi-Site Orchestrator GUI.

Step 2 Open the **System Logs** screen.



a) In the main menu, select **Operations** > **Tech Support**.
b) In the top right corner of the **System Logs** frame, click the edit button.

Step 3 Download the logs.



a) Select which logs you want to download.
b) Click the **Download** button.

An archive of the selected items will be downloaded to your system. The report contains the following information:

- All schemas in JSON format
- All sites definitions in JSON format
- All tenants definitions in JSON format
- All users definitions in JSON format
- All logs of the containers in the `infra_logs.txt` file

Gathering Docker Container Information

You can log in to one of the Orchestrator VMs and gather information about the Docker services and its logs for specific containers. A number of useful Docker commands is available from the following cheat sheet: https://www.docker.com/sites/default/files/Docker_CheatSheet_08.09.2016_0.pdf.

Inspecting the Health of Docker Containers

To inspect the health of Docker services, you can use the `docker service ls` command. The output of the command lists the current health status of each service. All services should have all containers replicated as displayed in the `REPLICAS` column. If any one of them is down, there may be issues that need to be addressed.

```
# 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       [...]
```

Getting Container IDs

You can get the list of all running container IDs using the `docker ps` command.

```
# 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…"   [...]
```

You can get the running container ID for a specific service using the `docker ps | grep <service-name>` command.

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

To get all container IDs for a service, including the ones that are exited, you can use the `docker ps -a | grep <service-name>` command.

```
# 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
```

Viewing Container Logs

Use the `docker logs <container-id>` command to view the logs for a container. The logs for a container could be large as there are many files to be transferred, so consider your network speed when you run the command.

The sample location of the log files for a container is `/var/lib/docker/containers/<container>` There can be multiple `<container>-json.log` files.

```
# 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
```

Viewing Docker Networks

You can view the list of networks used by Docker using the `docker network list` command.

```
# 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
```

Generating the API Call Logs

You can access the Multi-Site Orchestrator API call logs through the Infra Logs in a Troubleshooting Report. For information on generating troubleshooting, see Downloading System Logs, on page 11.

You can also access the API call logs Multi-Site with the following steps:

Step 1 Locate the worker node that has the `msc-executionengine` service running, as in the following example:

Example:

```
[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
```

In this case, on `cc693965f502` the image is `msc-executionengine:latest`, find the `-json.log`, that contains the API calls from Multi-Site to the APIC controllers.

Step 2 Enter the command in the following example:

Example:

```
# 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"}
```

Reading the Execution Log

The execution log provides three different kinds of log information:

- Websocket refresh information that is printed out every 5 minutes.

```
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
```

- The schema to push and the plan being generated.
- Websocket monitoring VNID for cross VNID programming.

Note the following signs of errors:

- Log lines starting with a red error.
- Stacktrace for exceptions.

Verifying Policy Resolution on APIC Sites

In this task, use a REST API MO query on local APIC sites or switches to view the policies resolved on an APIC, for a site managed by Cisco ACI Multi-Site.

For diagrams of the managed objects (MO) relationships, see the *Cisco APIC Management Information Model Reference* (MIM). For example, in the MIM, see the diagram for `fv:FabricExtConnP`.

Step 1 To view details for the logical MOs under the Fabric External Connection Profile (fabricExtConnP), log on to the APIC CLI and enter the following MO query:

Example:

```
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
```

Step 2 To view the logical MOs for the L3Out used for Multi-Site connections, log on to the APIC CLI and enter an MO query, such as the following:

Example:

```
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
```

Step 3 To view the resolved MOs for an APIC local site, log on to the APIC CLI and enter an MO query such as the following:

Example:

```
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
```

Step 4 To view the concrete MOs on a switch for a Multi-Site site, log on to the switch and enter an MO query such as the following:

Example:

```
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
```

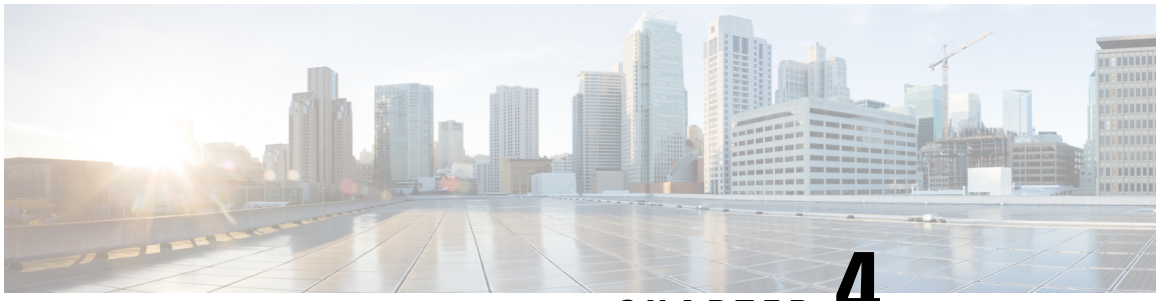
What to look for: The output displays the data translated between sites. In this example, the original data on the sites was as follows:

- `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`

Step 5 To verify the concrete MOs for a remote site, enter an MO query such as the following:

Example:

```
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
```



CHAPTER 4

Troubleshooting Installations, Upgrades, and Reboots

This chapter contains the following sections:

Increasing CPU Cycle Reservation for Orchestrator VMs

Cisco ACI Multi-Site Orchestrator VMs require a certain amount of dedicated CPU cycles. While new deployments apply CPU cycle reservation automatically, if you upgrade the Orchestrator from a release prior to Release 2.1(1), you will need to update each Orchestrator VM's settings manually.

Configuring appropriate CPU cycle reservation can resolve or help prevent a number of seemingly random issues, such as:

- Orchestrator GUI items requiring one or more retries to load.
- One or more nodes changing to `Unknown` status and then resolving back to `Ready` some time later on its own:

```
# 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
```

- Transient heartbeat misses in the Orchestrator logs (located in `/var/log/messages`) with the following example log entries:

```
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" [...]
```

To update the CPU cycle reservation setting, repeat the following steps for each Orchestrator VM:

Step 1 Log in to the vSphere client.

Step 2 Navigate to the ESX host where your Orchestrator VM is located.

Step 3 Shut down the VM.

Step 4 Right click the VM and choose **Edit Settings**

Step 5 In the **Virtual Hardware** tab, expand the **CPU** category.

Step 6 In the **Reservation** field, enter `10 GHz`.

Step 7 Click **OK** to save the changes.

Step 8 Power on the VM and wait for the Orchestrator cluster to stabilize with all nodes healthy.

Enabling NTP for Orchestrator Nodes

Not having clock synchronization configured for Orchestrator nodes may cause issues, such as random GUI session log off due to authentication token expiration.

Typically, you provide the Network Time Protocol (NTP) server details for the Orchestrator nodes during Multi-Site Orchestrator installation. However, if for any reason you have not specified NTP settings, you can configure them using the following steps.

Step 1 Log in directly to an Orchestrator VM.

Step 2 Change into the `scripts` directory.

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

Step 3 Configure the NTP settings for the node.

In the following command:

- '**-tz** *<time-zone>*' specifies the time zone you are in
- '**-ne** ' enables NTP
- '**-ns** *<ntp-server>*' specifies the NTP server

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

For example:

```
# ./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
```

Step 4 Verify NTP configuration.

You can verify that NTP is enabled using the following command:

```
# 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
```

You can also confirm that the correct date and time are set:

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

Step 5 Repeat the procedure on each Orchestrator node.

Updating DNS

This section describes how to update the DNS server address for your Multi-Site Orchestrator cluster. Note that this procedure applies to MSO OVA deployments in VMware ESX only and not Application Services Engine or Nexus Dashboard deployments.

Step 1 SSH in to one of the cluster nodes as the `root` user.

Step 2 Update the DNS configuration.

Use the `nmcli` command to update the DNS server IP address:

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

If you want to provide multiple DNS server IPs, use a space-separated list:

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

Step 3 Restart the network interface you updated.

For the changes to apply, you need to restart the `eth0` interface:

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

Step 4 Reboot the node.

Step 5 Repeat the previous steps for the other two nodes.

Restarting a Single Node of the Cluster if it Goes Down Temporarily

This section describes how to restart a single node of the cluster if it goes down temporarily.

Restart the node which was down. No additional steps are required and the cluster recovers by itself.

Restarting Two Nodes of Cluster that Go Down Temporarily

This sections describes how to restart two nodes of the cluster that go down temporarily.

Step 1 At this point due to lack of a quorum of 3 manager nodes in the Docker swarm, Multi-Site will not be available. Cisco recommends that you back up the MongoDB prior to any recovery attempts.

For more information, see Backing Up the MongoDB for Cisco ACI Multi-Site, on page 24.

Step 2 Restart the 2 nodes which were down. No additional steps are required and the cluster recovers by itself.

Backing Up the MongoDB for Cisco ACI Multi-Site

Cisco recommends that you back up the MongoDB prior to any Cisco ACI Multi-Site Orchestrator upgrades or downgrades, as described in this section.

Note You can back up the database only when the Orchestrator cluster is up, which requires at least 2 nodes to be up and running.

Step 1 Log in to the Cisco ACI Multi-Site Orchestrator virtual machine (VM).

Step 2 Execute the Cisco ACI Multi-Site Orchestrator backup script:

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

The `msc_backup_<date+%Y%m%d%H%M>.archive` file is created.

Step 3 Copy the `msc_backup_<date+%Y%m%d%H%M>.archive` file to a safe place.

Restoring the MongoDB for Cisco ACI Multi-Site

This section describes how to restore the MongoDB for Cisco ACI Multi-Site.

Step 1 Log in to the Multi-Site virtual machine (VM).

Step 2 Copy your `msc_backup_<date+%Y%m%d%H%M>.archive` file to the VM.

Step 3 Execute the Multi-Site DB restore script:

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

Step 4 Push the schemas again by executing the python script:

```
# msc_push_schemas.py
```

Custom Certificates Troubleshooting

The following sections describe how to resolve common issues when using custom SSL certificates with Multi-Site Orchestrator.

Unable to Load the Orchestrator GUI

If you are unable to load the Orchestrator GUI page after installing and activating a custom certificate, it is possible that the certificates were not copied correctly to each Orchestrator node. You can resolve this issue by recovering the default certificates and then repeating the new certificate installation procedure again.

To recover the default Orchestrator certificates:

Step 1 Log in to each Orchestrator node directly.

Step 2 Change into the certificates directory:

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

Step 3 Replace the `msc.key` and `msc.crt` files with `msc.key_backup` and `msc.crt_backup` files respectively.

Replace the `msc.key` and `msc.crt` files with `msc.key_backup` and `msc.crt_backup` files respectively.

Step 4 Restart the Orchestrator GUI service.

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

Step 5 Re-install and activate the new certificates as described in previous sections.

Adding a New Orchestrator Node to the Cluster

If you add a new node to you Multi-Site Orchestrator cluster:

Step 1 Log in to the Orchestrator GUI.

Step 2 Re-activate the key you are using as described in previous sections.

Unable to Install a New Keyring after the default Keyring Expired

If you are unable to install a new Keyring after the default Keyring expired, it is possible that custom Keyring is not installed on the cluster nodes.

You can resolve this issue by deleting the old default Keyring and creating a new Keyring using steps mentioned below:

Step 1 Execute the following commands on all the nodes in the cluster:

```
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
```

Step 2 Execute the following command to force update of msc_ui service:

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

Step 3 Once the update completes, check if all the replicas of msc_ui are healthy.

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

Step 4 Log in to any of the MSO nodes using any browser. If the browser is stuck in a loop while accepting the certificate details or displays a white screen, refresh the page one or two times until the GUI is displayed normally again.

Step 5 Log in using the username and password and activate the Keyring by following the steps described in the *'Activating Custom Keyring'* section.

Replacing a Single Node of the Cluster with a New Node

This section describes how to replace a single node of the cluster with a new node.

In this scenario node 1 goes down and you want to replace node 1 with a new node.

Step 1 On any existing node, get the ID of the node that is down (node1). Execute the following command:

```
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
```

Step 2 You must demote node1, execute the following command:

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

<*node ID*> is where you received the node ID from step 1.

Step 3 Remove node1 which is down before adding the new node, execute the following command:

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

Step 4 On any existing node, change to the /opt/cisco/msc/builds/<*build_number*>/prodha directory:

Example:

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

Step 5 Take note of the token. On any existing node, execute the following command:

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

Step 6 Taken note of IP address of the new leader. On any existing node, enter the following command:

Example:

```
[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
```

Step 7 On the leader node (node2), take note of the IP address:

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

Step 8 Prepare the new third node. Set the correct hostname for the new node:

Example:

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

Step 9 Change to the /opt/cisco/msc/builds/<*build_number*>/prodha directory:

Example:

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

Step 10 Join the new node to the swarm:

Example:

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

<*token*> is where you received the token information from step 5.

<*address of leader*> is where you received the leader IP address in step 7.

Step 11 On any node, change to the /opt/cisco/msc/builds/<*build_number*>/prodha directory:

Example:

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

Step 12 On any node, execute the following command:

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

At this point all service should be up and database replicated.

Replacing Two Existing Nodes of the Cluster with New Nodes

This section describes how to replace two existing nodes of the cluster with new nodes. The instructions here are for the Orchestrator deployments in ESX VMware VMs, which use Docker swarm for the cluster.

At this point due to lack of a quorum of 3 manager nodes in the Docker swarm, Multi-Site Orchestrator will not be available. We recommend that you back up the DB prior to any recovery attempts. For more information, see Backing Up the MongoDB for Cisco ACI Multi-Site, on page 24.

Step 1 Bring up two new nodes and set appropriate node names for each new node.

After you bring up a new node, you can assign its node name using the following command:

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

Keep in mind that all 3 node names must be unique within the cluster.

Step 2 On the only live node which was previously part of swarm, remove the other nodes that are down.

a) SSH into the only live node that was part of the swarm.

b) View the status of all nodes.

```
# 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) Delete any nodes with the **Down** status.

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

For example:

```
# docker node rm g3mebdulaed2n0cyywjrtum31
```

Step 3 Re-initiate the Docker swarm.

While still logged into the only live node, perform the following steps.

a) Leave the existing swarm.

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

b) Change into the Orchestrator scripts directory.

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

c) Re-initiate a new swarm.

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

The command will return a `token` and IP address you will need to use on the two new nodes to join them into the new cluster.

Step 4 Join the two new nodes into the cluster.

On each of the new nodes, perform the following steps.

a) SSH in to the node.
b) Change into the Orchestrator scripts directory.

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

c) Join the node into the cluster.

In the following command,

- Replace <*token*>, with the token you received from the **`./msc_cfg_init.py`** command when you re-initiated the Docker swarm in the previous step.
- Replace <*ip-address*> with the IP address of the first node, which you also received in the previous step.

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

Step 5 Deploy the new configuration.

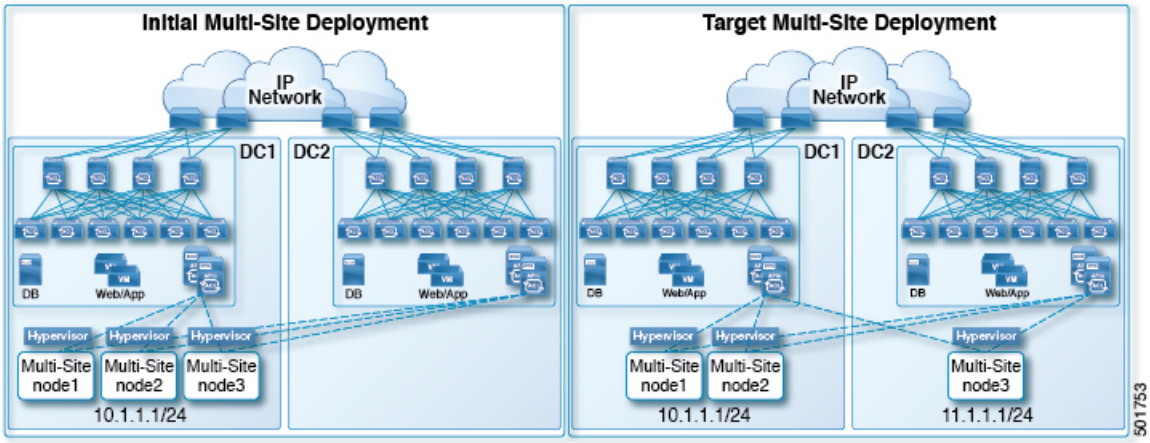
You can run the following command from any node in the new cluster. The script is located in the same `/opt/cisco/msc/builds/<build-number>/prodha` scripts directory.

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

Relocating Multi-Site Nodes to a Different Subnet

This section describes how to relocate one or more Multi-Site nodes from one subnet to another. This is a common task when Multi-Site is deployed within a single data center and the goal is to spread the nodes across one or more data centers. It is important to move a single node at a time to maintain redundancy during the migration.

Figure 1: Cisco ACI Multi-Site Deployments



The example procedure below shows the relocation of Multi-Site node3 from data center 1 where the management subnet uses a 10.1.1.1/24 subnet, to data center 2 where the management subnet uses a 11.1.1.1/24 subnet.

Step 1 On node1, demote node3:

Example:

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

Step 2 Power down node3 virtual machine (VM).

Step 3 Remove node3 from the cluster:

Example:

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

Step 4 Deploy the new Multi-Site VM (same version as node1 and node2) to the data center. Configure with the new IP details and ensure the hostname ‘node3’ gets assigned.

Step 5 Power up node3 in data center 2 and test the connectivity to node1 and node2:

Example:

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

Step 6 On node1, get the join token from node1 to join node3 to the cluster:

Example:

```
[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~]#
```

Step 7 On node3, join swarm using the join token from step 6.

Example:

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

Step 8 On any node, make sure the nodes are heathly. Verify that the STATUS is Ready, the AVAILABILITY is Active for each node, and the MANAGER STATUS is Reachable except for only one showing Leader:

Example:

```
[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 ~]#
```

Step 9 Update the swarm label for node3:

Example:

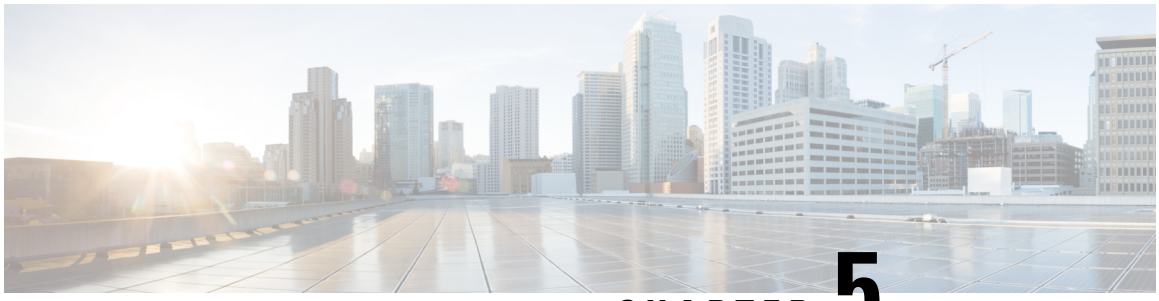
```
[root@node1 prodha]# docker node update node3 --label-add msc-node=msc-node3
```

Step 10 On any node, check the status of all the docker services. For example, make sure it states 1/1 (1 out of 1) or 3/3 (3 out of 3). This may take up to 15 minutes to sync up.

Example:

```
[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 ~]#
```

Step 11 Delete the original node3 VM that you powered down in data center 1.



CHAPTER 5

Troubleshooting Users

This chapter contains the following sections:

Resetting Local Admin Password

This section describes how to reset the local `admin` password for your Multi-Site Orchestrator cluster. Note that this procedure applies to MSO OVA deployments in VMware ESX only and not Application Services Engine or Nexus Dashboard deployments.

Step 1 SSH in to any one of the cluster nodes as the `root` user.

Step 2 Delete the `admin` credentials.

Use the following script to delete the `admin` credentials:

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

Step 3 Restart the `msc_userservice` service.

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

This will reset the `admin` user's password to the default password. Note that the default password depends on the specific version of the Multi-Site Orchestrator you are running, consult the *Cisco Multi-Site Installation and Upgrade Guide* for your version.

Troubleshooting Cisco ACI Multi-Site External User Authentication

Use the following tips to troubleshoot external user authentication problems.

Step 1 To investigate the error `Authentication method failed`, verify the following:

- The key given in the Provider configuration is correct
- The Multi-Site (client) IP address is registered in the remote Cisco ACS server

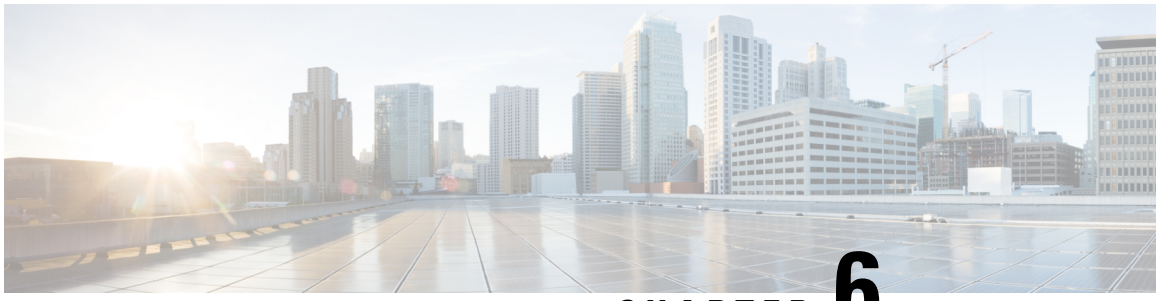
Step 2 To investigate the error `Invalid user credentials`, verify the following:

- The username entered on the Multi-Site login screen is correct and matches one that is configured on the Cisco ACS server
- The password entered on the Multi-Site login screen is correct and matches one that is configured on the Cisco ACS server

Step 3 If the user sees a Loading icon, followed by the errors `Loading ...` and `Authentication method failed`, verify the following:

- The IP address in the Provider configuration is correct
- The IP addresses for the Provider and Cisco ACS are reachable
- The port and protocol in the Provider configuration is correct
- The correct authentication method (TACACS+ or RADIUS) is selected on the remote ACS server under ...**Network Devices and AAA Clients** > **Authentication Options**
- The correct shared secret is provided in the remote ACS server user configuration, and it is not empty

Step 4 If the user is able to login, but is not able to see anything or is not able to see any tabs on the Multi-Site GUI, verify that the Cisco AV Pair and the roles are configured correctly for that user, on the remote ACS server.



CHAPTER 6

Troubleshooting Platform Health Issues

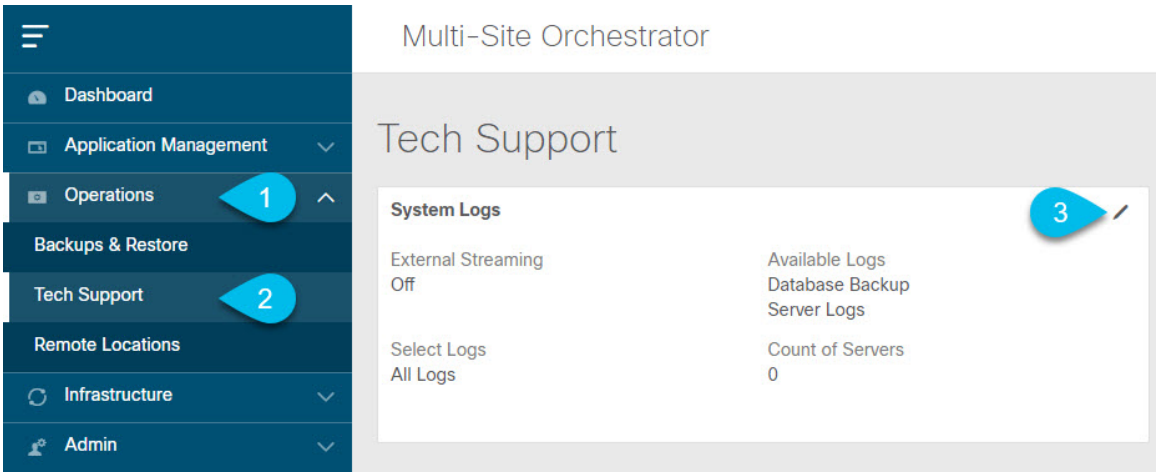
This chapter contains the following sections:

Downloading System Logs

This section describes how to generate a troubleshooting report and infrastructure logs file for all the schemas, sites, tenants, and users that are managed by Cisco ACI Multi-Site Orchestrator.

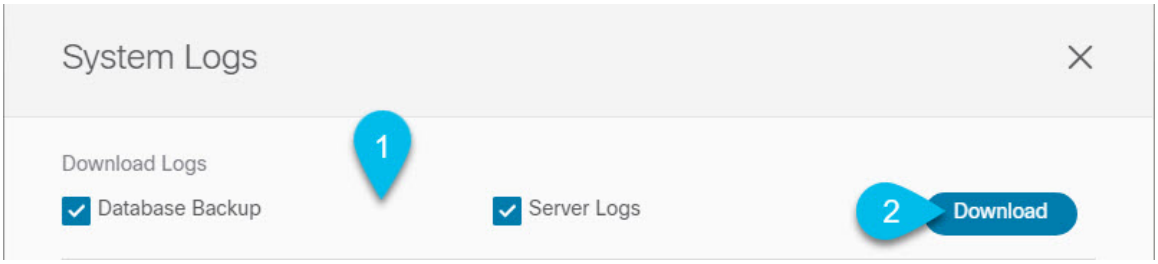
Step 1 Log in to your Multi-Site Orchestrator GUI.

Step 2 Open the **System Logs** screen.



a) In the main menu, select **Operations** > **Tech Support**.
b) In the top right corner of the **System Logs** frame, click the edit button.

Step 3 Download the logs.



a) Select which logs you want to download.
b) Click the **Download** button.

An archive of the selected items will be downloaded to your system. The report contains the following information:

- All schemas in JSON format
- All sites definitions in JSON format
- All tenants definitions in JSON format
- All users definitions in JSON format
- All logs of the containers in the `infra_logs.txt` file

Gathering Docker Container Information

You can log in to one of the Orchestrator VMs and gather information about the Docker services and its logs for specific containers. A number of useful Docker commands is available from the following cheat sheet: https://www.docker.com/sites/default/files/Docker_CheatSheet_08.09.2016_0.pdf.

Inspecting the Health of Docker Containers

To inspect the health of Docker services, you can use the `docker service ls` command. The output of the command lists the current health status of each service. All services should have all containers replicated as displayed in the `REPLICAS` column. If any one of them is down, there may be issues that need to be addressed.

```
# 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       [...]
```

Getting Container IDs

You can get the list of all running container IDs using the `docker ps` command.

```
# 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…"   [...]
```

You can get the running container ID for a specific service using the `docker ps | grep <service-name>` command.

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

To get all container IDs for a service, including the ones that are exited, you can use the `docker ps -a | grep <service-name>` command.

```
# 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
```

Viewing Container Logs

Use the `docker logs <container-id>` command to view the logs for a container. The logs for a container could be large as there are many files to be transferred, so consider your network speed when you run the command.

The sample location of the log files for a container is `/var/lib/docker/containers/<container>` There can be multiple `<container>-json.log` files.

```
# 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
```

Viewing Docker Networks

You can view the list of networks used by Docker using the `docker network list` command.

```
# 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
```

Removing Stale Docker Containers

Typically, when you perform a Multi-Site Orchestrator upgrade, the upgrade process removes any old docker containers that are replaced by a newer version from the upgrade. However, during normal Docker operation containers may be stopped, for example if a service fails, and new containers may be created to replace them. These stopped containers remain in the system until they are removed by the next upgrade.

If for any reason you want to manually remove any stale containers, you can use the instructions in this section to collect the container logs and remove the containers.

Step 1 Check if there are any stale containers in the system.

We recommend doing manual container removal **only** if you see an excessive number of stale containers in the system. If there are only a few or no stopped containers, we recommend leaving them to be removed by the next upgrade process.

You can check for any stale containers by running the `docker ps -a` command and checking for any containers with `Exited` status, for example:

```
# 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)
```

Step 2 Collect system logs using the Orchestrator GUI.

Removing inactive Docker containers also removes any logs associated with them. If you do choose to manually clean up old containers, we recommend first collecting and saving the logs in case you may need them later.

Collecting system logs is described in Downloading System Logs, on page 11.

Step 3 Log in to one of the nodes and execute the following commands to remove stale containers.

a) Ensure that the `exited` containers are stopped.

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

b) Remove the containers.

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

Step 4 Repeat the previous step for the other two nodes.

Troubleshooting Missing Node Labels

If you cannot log in to your Multi-Site Orchestrator GUI, but the Orchestrator nodes are still reachable via SSH, one of the nodes may have lost its label. This section describes how to diagnose this issue and resolve it by re-applying the proper node label.

Step 1 Log in to one of the Multi-Site Orchestrator nodes via SSH.

You can log in to any one of the nodes.

Step 2 Check if the MongoDB containers are properly replicated on all nodes.

```
# 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
[...]
```

In the above output, you can see that the MongoDB container is not properly replicated on one of the nodes.

Step 3 Find the hostnames of all the nodes.

```
# 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
```

Step 4 Inspect each node.

Repeat the following command for each node, replacing <*node-name*> with the hostname of the node from the previous step.

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

Example:

```
# 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"
        },
[...]
```

If one or more nodes are missing a label, the `Labels` field will be empty.

Step 5 Restore the missing label for the node.

In the following command:

- Replace <*node-label*> with the label appropriate for the node.

  While the hostname of each node can be customized, the labels must be `msc-node1`, `msc-node2`, or `msc-node3`.

- Replace <*node-name*> with the hostname of the node that is missing a label.

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

Example:

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

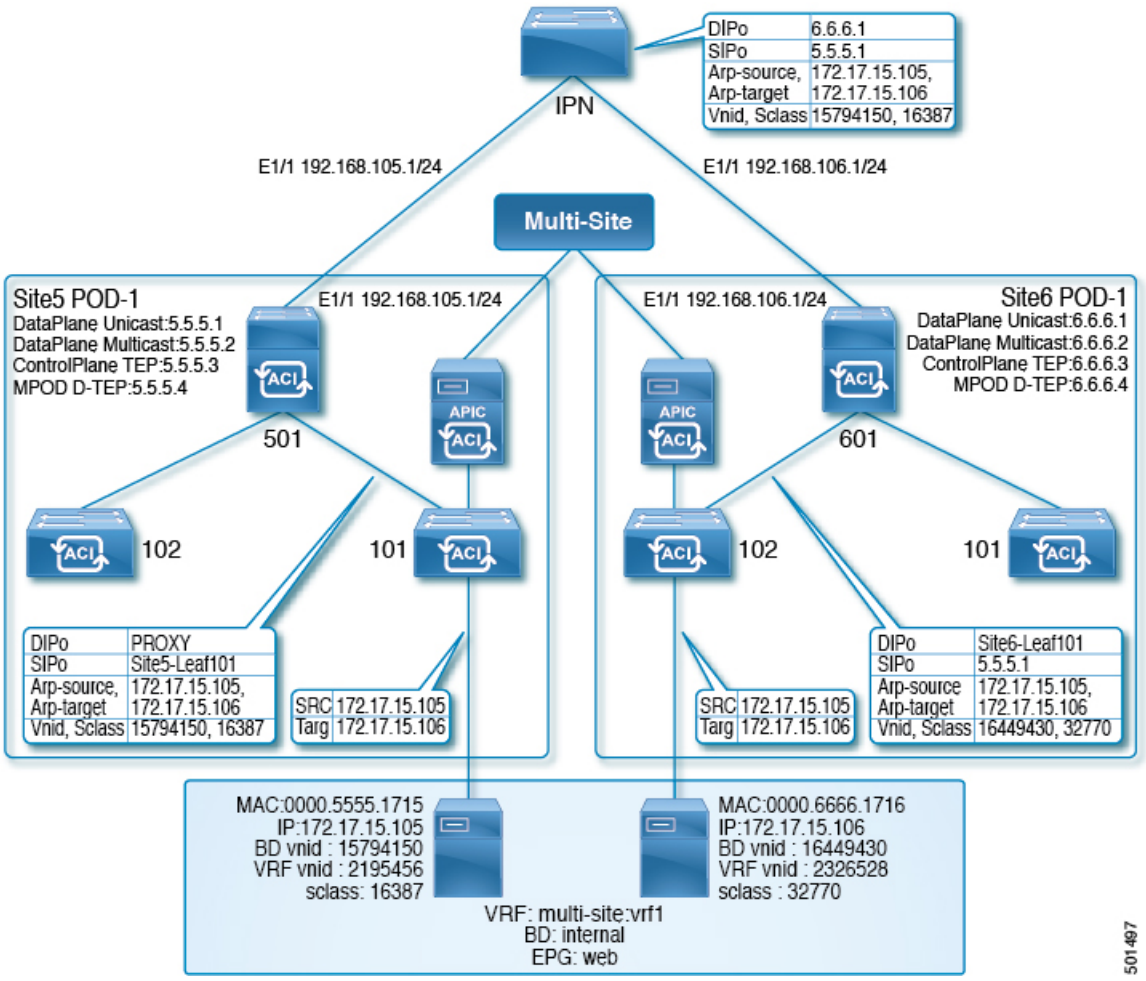
Step 6 Verify that the label was added correctly.

```
# 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"
        },
[...]
```

Troubleshooting Intersite Packet Flow in a Stretched BD Network

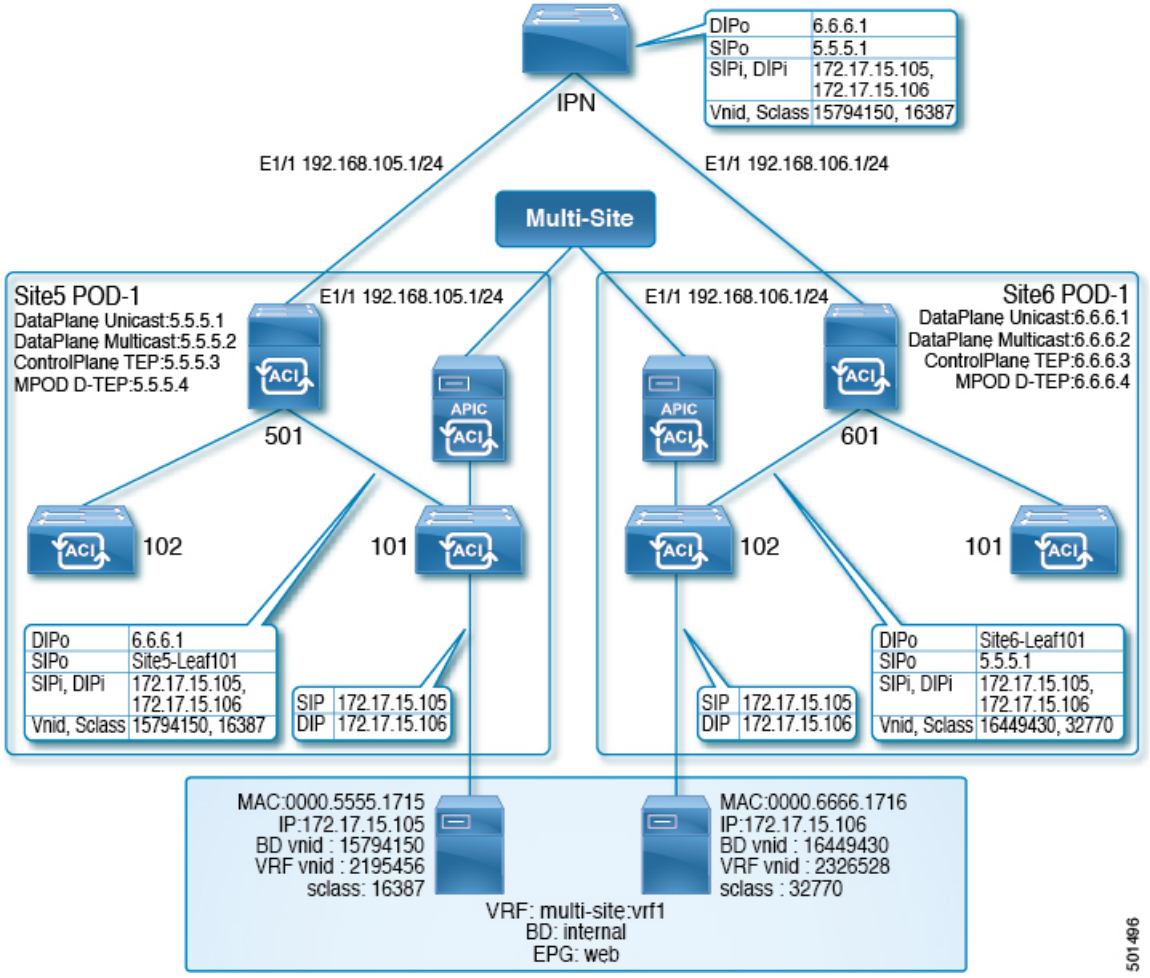
Figure 1 shows a stretched bridge domain (BD) network with Layer 2 Broadcast extension between sites. The BD is an L3 BD with ARP flood enabled, using L2 Unknown Unicast Proxy.

Figure 2: Intersite ARP Flow



Figrue 2 shows the same stretched BD network with focus on the unicast packet flow.

Figure 3: Intersite Unicast Flow



When the site5 host at 172.17.15.105 sends a unicast packet (for example, an ICMP echo) towards the site6 host with IP address 172.17.15.106, the following troubleshooting steps apply to the scenario, in which Site5-leaf101 has learned the site6 endpoint (EP), 172.17.15.106. If site5-leaf101 has not learned the site6 EP, it either floods the packet or sends spine501 data for proxy, based on the BD's Layer 2 unknown unicast forward settings.

Step 1 On the site5 ingress leaf switch (leaf101 in this case), use the NX-OS style CLI **show endpoint mac** *mac-address* command to determine whether the system has learned both the source and destination EP, as in the following example:

Example:

```
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
```

Step 2 If both the local and remote EPs have been learned, and the policy permits the EPs to communicate (using the default contract permitting intra-EPG traffic), site5-leaf101 encapsulates the ICMP packet with the following data, and forwards the packet out through the fabric uplink port towards the spine switch:

- VXLAN header's outer destination IP address, 6.6.6.1
- VXLAN ID (VNID), 15794150
- src-class (sclass), 16387
- Source IP address, which is the site5-leaf101 TEP address through the VRF overlay-1

When the spine switch receives a packet from the VRF overlay-1, it verifies that the destination IP address (DIP) belongs to the MAC proxy address. For example, if the DIP 6.6.6.1 does not belong to the MAC proxy address of spine501, the spine switch forwards the packet like a normal IP packet, based on the longest match in the routing table. In this case, since the DIP matches a remote site's spine overlay unciast TEP address, spine501 rewrites the outer source IP (SIP) address from site5-leaf101's TEP to site5's unicast overlay unicast TEP (5.5.5.1). In this process, Spine501 should have learnt 6.6.6.1 through OSPF in the interpod network (IPN), so spine501 forwards the packet to the next hop,which is the IPN switch in this case.

Step 3 If there is a concern about the packet being forwarded, run ERSPAN in fabric mode on the APIC, in the NX-OS style CLI, to capture the outgoing packet from the uplink interface, using commands such as in the following example.

Example:

This example configures Fabric ERSPAN to capture outgoing packets from switch 101, interface eth1/1, with focus on VRF vrf1, and BD bd1, in tenant t1.

```
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
```

For more information, see *Configuring SPAN* in *Cisco APIC NX-OS Style Command-Line Interface Configuration Guide*.

Step 4 To verify whether the routing table contains an explicit entry for 6.6.6.1, use the NX-OS style command, **show ip route 6.6.6.1 vrf overlay-1**.

Step 5 When the next-hop interface has been found, use the **show lldp neighbor** command to determine whether the next hop of the interface , where 6.6.6.1 is learned from, is the expected IPN interface.

Use the Fabric ERSPAN to confirm that spine501 has recieved the packet from the leaf switch or forwarded it through the correct egress interface.

Step 6 When the packet arrives at IPN, because it is a unicast packet, IPN forwards the IP packet based on the routing table. To confirm the routing table has the correct/expected next-hop interface, use the command **show ip route 6.6.6.1**.

Use Fabric ERSPAN to capture one packet or a multiple packets from different interfaces. The next-hop interface from IPN in the topology above is spine601's interface.

If the packet arrvies at the site6 switch, spine601, it maps the remote site's ID based on the outer SIP, 5.5.5.1 to site5, together with the source VNID to 15794150. Also, spine601 translates that VNID to the VNID of the local BD,16449430, and translates the src-class ID, 16387, to the local EP src-clas, 32770. Then it performs a look up based on the destination-MAC address, within the scope of the translated VNID.

Step 7 To verify the VNID translation bewteen site5 and site6, on spine601 enter the **show dcimgr repo vnid-maps verbose** command.

Step 8 To verify the sclass translation between site5 and site6, on spine601 enter the **show dcimgr repo sclass-maps** command.

Finally, spine601 rewrites the outer destination to the TEP of site6-leaf101 and forwards the packet there.

Step 9 To determine if the packet was forwarded properly, go to the expected leaf switch (site6-leaf101) run Fabric ERSPAN to capture the packet.

Step 10 If the packet arrives at site6-leaf101, leaf101 performs a local lookup, based on the destination MAC within the scope of VNID, 16449430, to determine the egress interface. To determine the egress interface, enter the **show endpoint mac** *mac-address* command.

Step 11 To determine if the packet was forwarded properly, use access SPAN to capture the outgoing packet on the expected interface, using commands such as in the following example.

Example:

This example configures SPAN in access mode to capture packets being sent on leaf 101, interface eth1/2, with focus on EPG epg1 in tenant t1.

```
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
```

This is the traditional SPAN configuration, local to an Access leaf node. Traffic originating from one or more access ports or port-channels can be monitored and sent to a destination port local to the same leaf node.

In the ACI fabric, you can also use an access mode ERSPAN configuration to monitor traffic originating from access ports, port-channels, and vPCs in one or more leaf nodes. For an ERSPAN session, the destination is always an EPG which can be deployed anywhere in the fabric. The monitored traffic is forwarded to the destination wherever the EPG is moved.

For more information, see *Configuring SPAN* in *Cisco APIC NX-OS Style Command-Line Interface Configuration Guide*.

Troubleshooting Inter-Site BGP Sessions

For Multi-Site BGP sessions to be established on site spine switches, the following settings are required:

- The update source should have the `mscp-etep` flag set
- The BGP peer type should be `inter-site`
- The node role should be `msite-speaker`

To troubleshoot inter-site BGP session failures, verify the following MOs on the spines, using Visore:

- `fvNodeDef`
- `bgpInfraPeerDef`
- `bgpAsP`
- `fvIntersitePeeringDef`
- `l3extIntersiteLoopBackIfPDef`
- `l3LbRtdIf` with type set to `inter-site`
- `LoopBackId`
- `ipv4If` with the same loopback ID has the modeExtn property set to `mscp-etep`

If any of these MOs are missing, the BGP sessions do not come up.

For information about entering queries using Visore, see *Accessing REST API Tools* in *Using the REST API* in the *Cisco APIC REST API Configuration Guide*.

Note Visore is supported on the Firefox, Chrome, and Safari browsers.

Step 1 In a supported browser, enter the URL of the spine switch followed by `/visore.html`, as in the following example:

Example:

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

Step 2 When prompted, log in using the same credentials you would use to log in to the spine CLI interface.

Step 3 Enter a query for `l3LbRtdIf` to verify that the type is `inter-site`.

Step 4 Enter a query for `Ipv4IF` to verify that the `mode` is `cp-etep` and the `modeExtn` is `mscp-etep`.

Step 5 Enter a query for Intersite `BgpPeers` to verify it was created with the `inter-site` type and uses the CP-TEP loopback address as the source interface.

Step 6 If any of these values is incorrect, go to the APIC for the site and correct the values. Return to the Sites tab in Multi-Site and click **CONFIGURE INFRA**, and then click **Apply**.

Recovering from BGP Connectivity Loss After Spine Switch Recommission

When you decommission and recommission a multi-site spine switch in one of the fabrics without removing it from the APIC, the external BGP peering is turned off and may remain disabled on the switch. This BGP peering is required by the Multi-Site Orchestrator for inter-site communication.

This section describes how to re-establish BGP peering by loading the latest site connectivity information and re-deploying the Infra configuration to the site in case it is required..

Step 1 Log in to the Cisco ACI Multi-Site Orchestrator GUI.

Step 2 Refresh site connectivity information.

a) In the **Main menu**, select **Infrastructure** > **Infra Configuration**.
b) In the top right of the main **Infra Configuration** view, click the **Configure Infra** button.
c) In the left pane, under **Sites**, select the site with the recommissioned spine switch.
d) In the main window, click the **Reload Site Data** button to pull fabric information from the APIC.
e) In the **Confirmation** dialog, ensure that you check the **Remove config for decommissioned spine nodes** checkbox.

When you select the checkbox, the old configuration information for the decommissioned spine switch will be removed from the Multi-Site Orchestrator database.

f) Finally, click **Yes** to confirm and load the connectivity information.

This will update the site connectivity information including the recommissioned switch by re-importing it from the APIC.

Step 3 Verify spine switches configuration.

Select the spine switches in the site you refreshed in the previous step and verify that all configuration is correct.

If you need to update any information, detailed instructions on spine switch Infra configuration are available in the "Infrastructure Management" chapter of the *Cisco ACI Multi-Site Configuration Guide*.

Step 4 In the top right of the main **Fabric Connectivity Infra** view, click the **Deploy** button.

If you have cloud sites in your Multi-Site deployment, you will have multiple options available here. Since you are updating on-premises sites only for the recommissioned switch, simply click **Deploy** to deploy the Infra configuration to that site.

Troubleshooting Unicast or Multicast Traffic Failures

Use these steps in Visore to troubleshoot inter-site Unicast and Multicast traffic failures.

For information about entering queries using Visore, see *Accessing REST API Tools* in *Using the REST API* in the *Cisco APIC REST API Configuration Guide*.

Note Visore is supported on the Firefox, Chrome, and Safari browsers.

Step 1 Browse to the spine switch's Visore page.

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

Step 2 Log in using the same credentials you would use to log in to the spine CLI interface.

Step 3 Enter a query to verify the `fvIntersiteConnPDef` and `fvIntersiteMcastConnPDef` MOs are under `fvSiteConnPDef`.

These are the remote site unicast and multicast DP TEPs.

Step 4 Enter a query to verify that the `tunnelIf` MO was created with the type `dci-ucast` or `dci-mcast-hrep`, and the destination is the same as the remote site DP TEPs.

Step 5 Check the local site unicast and multicast DP TEPs. Enter a query for `fvIntersiteConnPDef` under `fvPodConnPDef` and `fvIntersiteMcastConnPDef` under `fvFabricExtConnPDef`.

Step 6 Enter a query for the `SiteLocal ipv4If` MOs to verify they were created with the mode `dci-ucast` and `dci-mcast-hrep` and the `ipv4Addr` MO was configured under it with the same address as the DP TEP's.

Step 7 If any of these values is incorrect, go to the APIC for the site and correct the values. Return to the Sites tab in Multi-Site and click **CONFIGURE INFRA**, and then click **Apply**.

Troubleshooting Multi-Site Multicast Functionality

This task provides the steps for troubleshooting the Multi-Site multicast functionality in a stretched bridge domain (BD) use case. This topic assumes that the `L2STRETCH` and `INTERSITEBUMTRAFFICALLOW` options are enabled in the stretched BD.

Multicast traffic flows between sites in the following process:

- **TX (Sending) from local to remote site**

  The Group IP Outer address (GIPo) traffic (part of Layer 2 Broadcast, Unknown Unicast, Multicast traffic) from the local site is Head-End Replicated (HREP) to each remote site from the Spine switch. The Destination IP address of the outer header (DIPo) is rewritten to a unicast address called as Multicast HREP TEP IP (also called Multicast DP-TEP IP) of the remote site. The Source IP address of the outer header (SIPo) is rewritten with the Unicast ETEP IP.

- **RX (Receiving) by remote from local site**

  Incoming traffic destined to the local site Multicast HREP TEP IP address is translated. The APIC on the site derives the local site BD-GIPo from that data, and follows the regular GIPo lookup path from then on.

To troubleshoot problems in this process, log on to the spine switch CLIs and use the following steps:

Step 1 To verify the locally configured Multi-Site TEP IP addresses, log on to the Supervisor module, and enter a command such as the following example:

Example:

```
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
```

Step 2 To confirm the MFDM on the spine switch, log on to the Supervisor module and enter a command such as the following example:

Example:

```
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
```

Step 3 To verify HREP TEP IP address reachability, log on to the Supervisor module, and enter a command such as the following example:

Example:

```
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
```

Step 4 To verify the MFIB on the spine switch line card module, log on to the module as root and use an (`vsh_lc`) command such as the following example:

Example:

```
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   |
```

Step 5 To verify the multicast HREP TEP details for remote sites, log on to the spine switch line card module to investigate the SDK, by entering a command such as the following example:

Example:

```
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
```

Step 6 To verify the GIPo route having HREP tunnels for the remote sites, log on to the spine switch Supervisor module and examine IS-IS details on the spine switch with a command, such as the following example:

Example:

```
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
```

Step 7 To verify the GIPo route having HREP tunnels for the remote sites, examine the MRIB on the spine switches using a command, such as the following example:

Example:

```
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
```

Step 8 To verify the MFIB on an FC, log on to the module as root, and use a command such as the following example:

Example:

```
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
```

Step 9 To verify the GIPo route having HREP tunnels for the remote sites, examine the SDL on FC, using a command such as the following example:

Example:

```
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
```

Step 10 To verify the GIPo route to the remote sites, examine the replication list identified in the last step, using a command such as the following example:

Example:

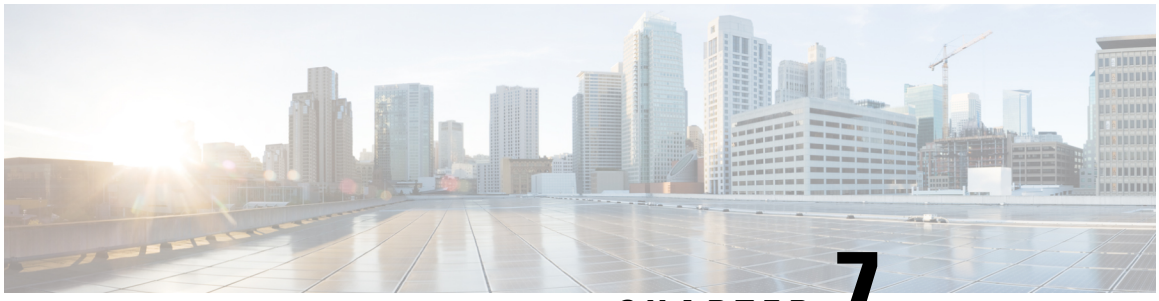
```
root@module-24# show platform internal hal objects mcast mcastrepllist id 0x600001e4
## Get Objects for mcast mcastrepllist for Asic 0
- - - - - - - - - -
Repl-List Asicpd Debug :
Entry-Num 0
Repl Entry Id:      0x1e5 Hw Epg Id:    4050 Hw Bd Id:   4050
Mc Id:                484 Met Id:        485 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
```

Step 11 To verify the VNID and GIPo mappings on the local (TX) and remote (RX) sites, enter a command such as the following example:

Example:

```
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
```



CHAPTER 7

Troubleshooting Tenants and Schemas

This chapter contains the following sections:

Troubleshooting Deployment Errors From APIC

When deploying tenant policies you have configured in a Cisco ACI Multi-Site schema, you may receive errors or issues may occur. To troubleshoot these errors and issues follow these steps.

Step 1 If you receive an APIC error after clicking **DEPLOY TO SITES**, correct any problems and try to deploy the schema/template again. For example, errors can be of two kinds:

- Misconfiguration—for example, a required association has not been defined, such as forgetting to choose the VRF for a BD.
- Site problem—there may be problems such as network failures, communication failures, or problems with the Infra settings. Save the schema, address the site problem, then return to deploy the schema again.

Step 2 If the schema is deployed successfully, but traffic does not flow, perform the following steps:

a) Generate a Troubleshooting Report and examine it for errors; see Downloading System Logs, on page 11

b) Generate a policy report and examine the tenant policy configuration; see Generating a Tenant Policy Report Using the REST API, on page 53

c) Log on to the Multi-Site VM and generate the execution log to find errors; see Gathering Docker Container Information, on page 12 and Reading the Execution Log, on page 15.

Step 3 If you find no errors, the problem may be with APIC, switches, the IPN or the WAN.

Generating a Tenant Policy Report Using the REST API

To generate a tenant policy report, use the Multi-Site REST API, to enter a query such as the following example.

When receiving the query, Multi-Site queries APIC for all policies defined in the tenant, generating traffic between Multi-Site and APIC. You may want to do this during a maintenance window.

To list the tenants with issues, enter a query such as the following example and copy the output:

Example:

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

Undeploying Schemas and Templates

In troubleshooting, if you find that some tenant policies are incorrectly configured, you may want to undeploy a template or schema and later recreate it. To undeploy templates and schemas follow these steps.

Step 1 Undeploy a template that is deployed to one site:

a) In the Schema tab click the three dots on a site-specific template.
b) Click **YES** to confirm.

Step 2 Undeploy a template on one site, that is deployed to multiple sites:

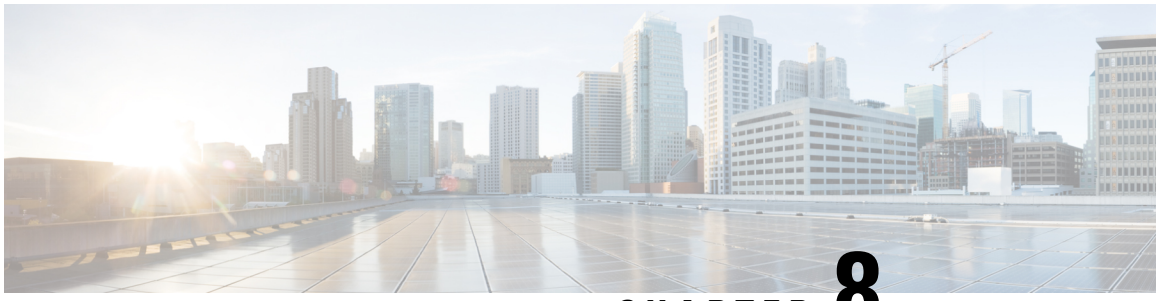
a) In the Schema tab click the template.
b) Click **+** to open the sites-selection panel.
c) On the row for the site, click **X** on the template.
d) Click **SAVE**.
e) Redeploy the remaining templates to the site.

Step 3 Remove a schema from all sites:

a) In the Schemas tab, click the schema.
b) Click **Actions** and choose **Delete**.
c) Confirm you want to undeploy the schema and click **YES**.

What to do next

Correct the schema or template then redeploy.



CHAPTER 8

Troubleshooting Multipod and Multi-Site Issues

This chapter contains the following sections:

Troubleshooting Multi-Site and Multi-Pod

This section describes how to troubleshoot Multi-Site and Multi-Pod.

Error:400

If you receive the following error:

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

You must enable the fabric external connectivity for all the existing spines and if you are trying to add new spines use the **Setup Multipod** GUI wizard.

There are two ways to resolve this issue.

- Enable all the spines under the external routed network:
  - In the APIC GUI, on the menu bar, click **Tenant** > **infra**.
  - In the **Navigation** pane, expand **Networking** > **External Routed Networks**, right-click on the external routed network and choose **Enable Fabric External Connectivity**.
- Add new spines under the external routed network:
  - In the APIC GUI, on the menu bar, click **Fabric**.
  - In the **Navigation** pane, expand **Quick Start** > **Node or Pod Setup** > **Setup Multipod** and complete the Multipod setup.

Verifying Remote Leaf Configuration

After you enable direct communication for Remote Leaf switches, you can verify the configuration using the following steps.

Step 1 SSH in to the switch.

Step 2 Verify that direct communication is enabled.

In the following output, verify that `rldirectMode` is set to *yes*:

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

Step 3 Verify that the remote leaf switches are in complete routable mode and are talking to Cisco APIC's public IP address.

a) Verify that `rlRoutableMode` is set to *yes*.

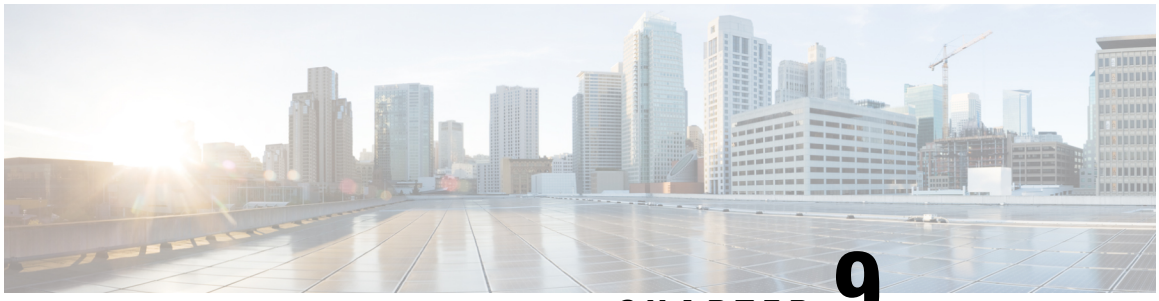
```
remote-leaf-switch# moquery -c topSystem | grep rlRoutableMode
rlRoutableMode           : yes
```

b) Verify that you can ping the Cisco APIC routable IP address from the remote leaf switch.

```
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) Verify that `dhcpRespMo` in the remote leaf switch is set to the APIC's routable IP address.

```
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
```



CHAPTER 9

Verifying NXOS Hardware Tables

This chapter contains the following sections:

Verifying End Point Manager Learning

Use the following commands to verify End Point Manager (EPM) learning.

In the following example, you can use the command to verify that source EP 172.17.15.105 is discovered in site5, leaf101. This output shows that for EP 172.17.15.105, the BD-VNID is 15794150, VRF-VNID is 2195456, and the pcTag or sclass is 16387.

```
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|
::::
```

Verifying BGP EVPN Routing Table

Use the following commands to verify the BGP EVPN routing table.

In this example, the end point 172.17.15.105 discovered from leaf101 is published to spine501 via COOP by EPM (Endpoint manager). COOP process on the spine and then syncs the EP to L2vpn EVPN. The command output show us that EP 172.17.15.105 is local to site 5 and being advertised to site 6 by BGP EVPN.

```
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
```

Verifying VNID, S-Class, and VTEP Mappings

Use the following command to verify remote site ID.

For VNID and pcTag or s-class translation between sites, the translation should be verified from the destination site. For example, if the packet is sending from site5 to site6, the translation is done by the site6's spine. To verify if the translation is pushed to site5, use the following command.

```
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
```

Use the following command to verify sclass-map between the remote and local site.

```
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]
```

Use the following commands to verify VRF or BD VNID map between remote and local site.

```
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
```

Use the following commands to verify DCI HAL objects.

```
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
```

Use the following commands to verify VXLAN objects.

```
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
```

Verifying LC Hardware Tables

Use the following commands to verify LC hardware tables.

```
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
```