Troubleshoot ACI Fabric Discovery - Initial Fabric Setup

Contents

Introduction

This document describes the steps understand and troubleshoot the initial fabric discovery process, including example issue scenarios.

Background Information

The material from this document was extracted from the Troubleshooting Cisco Application Centric Infrastructure, Second Edition book, specifically the **Fabric Discovery - Initial fabric setup** chapter.

Fabric discovery workflow

The ACI fabric discovery process follows a specific sequence of events. The basic steps are as follows:

1. Connect to the **KVM console** of the first APIC and complete the **setup script** by inputting values such as fabric name, APIC cluster size, and tunnel endpoint (TEP) address pool.

2. Once completed, the APIC1 will begin sending **LLDP** via its fabric ports. The LLDP packets contain special TLVs with information such as the **infra VLAN** and its role as an APIC (also referred to as the controller).
3. On reception of these LLDP packets from APIC1 the leaf will program the infra VLAN on all ports where an APIC is detected.
4. The leaf begins sending DHCP Discovers on the now-known infra VLAN.
5. The user logs into the **OOB IP** of APIC1 via HTTPS and registers the first leaf node in the **Fabric Membership** submenu.
6. Once the leaf is given a **Node ID**, APIC1 will respond with an IP address from the configured **TEP address pool** and the DHCP process completes.
7. The registered leaf relays DHCP Discovers from other directly connected spines which were discovered via LLDP to the APIC1.
8. The user will see those dynamically discovered spines appear in the Fabric Membership submenu and can register them.
9. Once the spines are registered, the APIC1 responds with an IP address from the TEP pool and DHCP completes for those nodes.
10. The spines relay DHCP Discovers from all other nodes of pod1. (This is assuming there is a full-mesh between spines and leaf switches as is advised and is the typical architecture).
11. Once the leaf nodes connected to the other APICs are registered, the APIC cluster can be established via TCP communication amongst themselves. Make sure to complete the setup dialog on APIC2 and APIC3.
12. Confirm all APICs have formed a cluster and are fully fit. If this is the case, fabric discovery is complete.

Beginning in 4.2, a new CLI command is available on fabric nodes to assist in the diagnosis of common discovery issues. The following sections will cover the checks performed and provide additional validation commands to assist in troubleshooting failures.

```
leaf101# show discoveryissues
Checking the platform type................LEAF!
Check01 - System state - in-service [ok]
Check02 - DHCP status [ok]
TEP IP: 10.0.72.67 Node Id: 101 Name: leaf101
Check03 - AV details check [ok]
Check04 - IP rechability to apic [ok]
Ping from switch to 10.0.0.1 passed
Check05 - infra VLAN received [ok]
infra vLAN:3967
Check06 - LLDP Adjacency [ok]
Found adjacency with SPINE
Found adjacency with APIC
Check07 - Switch version [ok]
version: n9000-14.2(1j) and apic version: 4.2(1j)
Check08 - FPGA/BIOS out of sync test [ok]
Check09 - SSL check [check]
SSL certificate details are valid
Check10 - Downloading policies [ok]
Check11 - Checking time [ok]
2019-09-11 07:15:53
Check12 - Checking modules, power and fans [ok]
```

Check01 — System state

When the leaf has been allocated a Node ID and registered to the fabric, it will begin to download its bootstrap and then transition to an **in-service** state.

```
Check01 - System state - out-of-service [FAIL]

Check01 - System state - downloading-boot-script [FAIL]
```

To validate the current state of the leaf, the user can run **moquery -c topSystem**

```
leaf101# moquery -c topSystem
Total Objects shown: 1

# top.System
address                  : 10.0.72.67
bootstrapState           : done
...
serial                   : FDO20160TPS
serverType               : unspecified
siteId                   : 1
state                    : in-service
status                   :
systemUpTime             : 00:18:17:41.000
tepPool                  : 10.0.0.0/16
unicastXrEpLearnDisable  : no
version                  : n9000-14.2(1j)
virtualMode              : no
```

Check02 — DHCP status

```
Check02 - DHCP status [FAIL]
ERROR: node Id not configured
ERROR: Ip not assigned by dhcp server
ERROR: Address assigner's IP not populated
TEP IP: unknown Node Id: unknown Name: unknown
```

The leaf needs to receive a TEP address via DHCP from the APIC1 and then establish IP connectivity to the other APICs. The **Physical TEP** (PTEP) of the leaf is assigned to loopback0. If no address is assigned, the user can validate the leaf is sending a DHCP Discover with tpcdump utility. Notice for this we will use interface kpm_inb which allows you to see all CPU inband control plane network traffic.

```
(none)# tcpdump -ni kpm_inb port 67 or 68
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on kpm_inb, link-type EN10MB (Ethernet), capture size 65535 bytes
16:40:11.041148 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from a0:36:9f:c7:a1:0c,
length 300
^C
1 packets captured
1 packets received by filter
0 packets dropped by kernel
```

The user can also validate dhcpd is running on the APIC and listening on the bond0 sub-interface. The bond interface represents the fabric facing APIC ports. We will use the format bond0.<infra VLAN>.

```
apic1# ps aux | grep dhcp
root     18929  1.3  0.2 818552 288504 ?       Ssl  Sep26  87:19 /mgmt//bin/dhcpd.bin -f -4 -cf
/data//dhcp/dhcpd.conf -lf /data//dhcp/dhcpd.lease -pf /var/run//dhcpd.pid --no-pid bond0.3967
```

```
admin   22770  0.0  0.0   9108   868 pts/0    S+   19:42   0:00 grep dhcp
```

Check03 — AV details

```
Check03 - AV details check [ok]
```

The leaf will validate if the registered APIC has an IP in a valid range for the TEP pool. If no APIC information has been recorded yet, this check will pass. The user can see the current APIC information from the leaf node's perspective via 'acidiag avread' command. Notice in the below example that when the leaf/spine prompt is showing (none)#, this is an indication the leaf/spine is not yet a member of the fabric.

```
(none)# acidiag avread
Cluster of 0 lm(t):0(zeroTime) appliances (out of targeted 0 lm(t):0(zeroTime)) with
FABRIC_DOMAIN name=Undefined Fabric Domain Name set to version= lm(t):0(zeroTime);
discoveryMode=PERMISSIVE lm(t):0(zeroTime); drrMode=OFF lm(t):0(zeroTime)
---------------------------------------------
clusterTime=<diff=0 common=2019-10-01T18:51:50.315+00:00 local=2019-10-01T18:51:50.315+00:00
pF=<displForm=1 offsSt=0 offsVlu=0 lm(t):0(zeroTime)>>
---------------------------------------------

leaf101# acidiag avread
Cluster of 3 lm(t):0(2019-09-30T18:45:10.320-04:00) appliances (out of targeted 3 lm(t):0(2019-
10-01T14:52:55.217-04:00)) with FABRIC_DOMAIN name=ACIFabric1 set to version=apic-4.2(1j)
lm(t):0(2019-10-01T14:52:55.217-04:00); discoveryMode=PERMISSIVE lm(t):0(1969-12-
31T20:00:00.003-04:00); drrMode=OFF lm(t):0(1969-12-31T20:00:00.003-04:00); kafkaMode=OFF
lm(t):0(1969-12-31T20:00:00.003-04:00)
      appliance id=1  address=10.0.0.1 lm(t):2(2019-09-27T17:32:08.669-04:00) tep
address=10.0.0.0/16 lm(t):1(2019-07-09T19:41:24.672-04:00) routable address=192.168.1.1
lm(t):2(2019-09-30T18:37:48.916-04:00) oob address=0.0.0.0 lm(t):0(zeroTime) version=4.2(1j)
lm(t):1(2019-09-30T18:37:49.011-04:00) chassisId=c67d1076-a2a2-11e9-874e-a390922be712
lm(t):1(2019-09-30T18:37:49.011-04:00) capabilities=0X3EEFFFFFFFFF--0X2020--0X1 lm(t):1(2019-09-
26T09:32:20.747-04:00) rK=(stable,absent,0) lm(t):0(zeroTime) aK=(stable,absent,0)
lm(t):0(zeroTime) oobrK=(stable,absent,0) lm(t):0(zeroTime) oobaK=(stable,absent,0)
lm(t):0(zeroTime) cntrlSbst=(APPROVED, FCH1929V153) lm(t):1(2019-10-01T12:46:44.711-04:00)
(targetMbSn= lm(t):0(zeroTime), failoverStatus=0 lm(t):0(zeroTime)) podId=1 lm(t):1(2019-09-
26T09:26:49.422-04:00) commissioned=YES lm(t):101(2019-09-30T18:45:10.320-04:00) registered=YES
lm(t):3(2019-09-05T11:42:41.371-04:00) standby=NO lm(t):0(zeroTime) DRR=NO lm(t):101(2019-09-
30T18:45:10.320-04:00) apicX=NO lm(t):0(zeroTime) virtual=NO lm(t):0(zeroTime) active=YES
      appliance id=2  address=10.0.0.2 lm(t):2(2019-09-26T09:47:34.709-04:00) tep
address=10.0.0.0/16 lm(t):2(2019-09-26T09:47:34.709-04:00) routable address=192.168.1.2
lm(t):2(2019-09-05T11:45:36.861-04:00) oob address=0.0.0.0 lm(t):0(zeroTime) version=4.2(1j)
lm(t):2(2019-09-30T18:37:48.913-04:00) chassisId=611febfe-89c1-11e8-96b1-c7a7472413f2
lm(t):2(2019-09-30T18:37:48.913-04:00) capabilities=0X3EEFFFFFFFFF--0X2020--0X7 lm(t):2(2019-09-
26T09:53:07.047-04:00) rK=(stable,absent,0) lm(t):0(zeroTime) aK=(stable,absent,0)
lm(t):0(zeroTime) oobrK=(stable,absent,0) lm(t):0(zeroTime) oobaK=(stable,absent,0)
lm(t):0(zeroTime) cntrlSbst=(APPROVED, FCH2045V1X2) lm(t):2(2019-10-01T12:46:44.710-04:00)
(targetMbSn= lm(t):0(zeroTime), failoverStatus=0 lm(t):0(zeroTime)) podId=1 lm(t):2(2019-09-
26T09:47:34.709-04:00) commissioned=YES lm(t):101(2019-09-30T18:45:10.320-04:00) registered=YES
lm(t):2(2019-09-26T09:47:34.709-04:00) standby=NO lm(t):0(zeroTime) DRR=NO lm(t):101(2019-09-
30T18:45:10.320-04:00) apicX=NO lm(t):0(zeroTime) virtual=NO lm(t):0(zeroTime) active=YES
      appliance id=3  address=10.0.0.3 lm(t):3(2019-09-26T10:12:34.114-04:00) tep
address=10.0.0.0/16 lm(t):3(2019-09-05T11:42:27.199-04:00) routable address=192.168.1.3
lm(t):2(2019-10-01T13:19:08.626-04:00) oob address=0.0.0.0 lm(t):0(zeroTime) version=4.2(1j)
lm(t):3(2019-09-30T18:37:48.904-04:00) chassisId=99bade8c-cff3-11e9-bba7-5b906a49dc39
lm(t):3(2019-09-30T18:37:48.904-04:00) capabilities=0X3EEFFFFFFFFF--0X2020--0X4 lm(t):3(2019-09-
26T10:18:13.149-04:00) rK=(stable,absent,0) lm(t):0(zeroTime) aK=(stable,absent,0)
lm(t):0(zeroTime) oobrK=(stable,absent,0) lm(t):0(zeroTime) oobaK=(stable,absent,0)
lm(t):0(zeroTime) cntrlSbst=(APPROVED, FCH1824V2VR) lm(t):3(2019-10-01T12:48:03.726-04:00)
(targetMbSn= lm(t):0(zeroTime), failoverStatus=0 lm(t):0(zeroTime)) podId=2 lm(t):3(2019-09-
26T10:12:34.114-04:00) commissioned=YES lm(t):101(2019-09-30T18:45:10.320-04:00) registered=YES
```

```
lm(t):2(2019-09-05T11:42:54.935-04:00) standby=NO lm(t):0(zeroTime) DRR=NO lm(t):101(2019-09-
30T18:45:10.320-04:00) apicX=NO lm(t):0(zeroTime) virtual=NO lm(t):0(zeroTime) active=YES
---------------------------------------------
clusterTime=<diff=15584 common=2019-10-01T14:53:01.648-04:00 local=2019-10-01T14:52:46.064-04:00
pF=<displForm=0 offsSt=0 offsVlu=-14400 lm(t):21(2019-09-26T10:40:35.412-04:00)>>
---------------------------------------------
```

Check04 — IP reachability to APIC

When the leaf has received an IP address, it will attempt to establish TCP sessions with the APIC and begin the process of downloading its configuration. The user can validate IP connectivity to the APIC using the 'iping' utility.

```
leaf101# iping -V overlay-1 10.0.0.1
PING 10.0.0.1 (10.0.0.1) from 10.0.0.30: 56 data bytes
64 bytes from 10.0.0.1: icmp_seq=0 ttl=64 time=0.651 ms
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.474 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.477 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.54 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.5 ms

--- 10.0.0.1 ping statistics ---  5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.474/0.528/0.651 ms
```

Check05 — Infra VLAN

```
Check05 - infra VLAN received [ok]
```

The infra VLAN check will only be successful if the node is connected to a Pod where an APIC exists. If this is not the case, the user can ignore the message because the check is expected to fail.

The leaf will determine the infra VLAN based on LLDP packets received from other ACI nodes. The first one it receives will be accepted when the switch is in discovery.

```
(none)# moquery -c lldpInst
Total Objects shown: 1

# lldp.Inst
adminSt        : enabled
childAction    :
ctrl           :
dn             : sys/lldp/inst
holdTime       : 120
infraVlan      : 3967
initDelayTime  : 2
lcOwn          : local
modTs          : 2019-09-12T07:25:33.194+00:00
monPolDn       : uni/fabric/monfab-default
name           :
operErr        :
optTlvSel      : mgmt-addr,port-desc,port-vlan,sys-cap,sys-desc,sys-name
rn             : inst
status         :
sysDesc        : topology/pod-1/node-101
txFreq         : 30

(none)# show vlan encap-id 3967
```

```
 VLAN Name                             Status    Ports
 ---- -------------------------------- --------- -------------------------------
 8    infra:default                    active    Eth1/1

 VLAN Type  Vlan-mode
 ---- ----- ----------
 8    enet  CE
```

If the infra VLAN has not been programmed on the switchport interfaces connected to the APICs, check for wiring issues detected by the leaf.

```
(none)# moquery -c lldpIf -f 'lldp.If.wiringIssues!=""'
Total Objects shown: 1

# lldp.If
id           : eth1/1
adminRxSt    : enabled
adminSt      : enabled
adminTxSt    : enabled
childAction  :
descr        :
dn           : sys/lldp/inst/if-[eth1/1]
lcOwn        : local
mac          : E0:0E:DA:A2:F2:83
modTs        : 2019-09-30T18:45:22.323+00:00
monPolDn     : uni/fabric/monfab-default
name         :
operRxSt     : enabled
operTxSt     : enabled
portDesc     :
portMode     : normal
portVlan     : unspecified
rn           : if-[eth1/1]
status       :
sysDesc      :
wiringIssues : infra-vlan-mismatch
```

Check06 — LLDP adjacency

```
Check06 - LLDP Adjacency [FAIL]
Error: leaf not connected to any spine
```

In order to determine which ports connect to other ACI devices, the leaf must receive LLDP from the other fabric nodes. To validate LLDP has been received, the user can check 'show lldp neighbors'.

```
(none)# show lldp neighbors
Capability codes:
 (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
 (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID            Local Intf      Hold-time  Capability  Port ID
apic1                 Eth1/1          120                    eth2-1
apic2                 Eth1/2          120                    eth2-1
switch                Eth1/51         120        BR          Eth2/32
switch                Eth1/54         120        BR          Eth1/25
Total entries displayed: 4
```

Check07 — Switch version

```
Check07 - Switch version [ok]
version: n9000-14.2(1j) and apic version: 4.2(1j)
```

If the APIC and leaf versions are not the same, fabric discovery could fail. To validate the version running on the leaf, use 'show version' or 'vsh -c 'show version''.

```
(none)# show version
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Documents:
http://www.cisco.com/en/US/products/ps9372/tsd_products_support_series_home.htmlCopyright (c)
2002-2014, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php

Software
 BIOS:      version 07.66
 kickstart: version 14.2(1j) [build 14.2(1j)]
   system:    version 14.2(1j) [build 14.2(1j)]
   PE:        version 4.2(1j)
 BIOS compile time:       06/11/2019
 kickstart image file is: /bootflash/aci-n9000-dk9.14.2.1j.bin
 kickstart compile time:  09/19/2019 07:57:41 [09/19/2019 07:57:41]
 system image file is:    /bootflash/auto-s
 system compile time:     09/19/2019 07:57:41 [09/19/2019 07:57:41]
...
```

The same command will also work on the APICs.

```
apic1# show version
Role        Pod         Node        Name                      Version
----------  ----------  ----------  ------------------------  --------------------
controller  1           1           apic1                     4.2(1j)
controller  1           2           apic2                     4.2(1j)
controller  2           3           apic3                     4.2(1j)
leaf        1           101         leaf101                   n9000-14.2(1j)
leaf        1           102         leaf102                   n9000-14.2(1j)
leaf        1           103         leaf103                   n9000-14.2(1j)
spine       1           1001        spine1                    n9000-14.2(1j)
spine       1           1002        spine2                    n9000-14.2(1j)
```

Check08 — FPGA/EPLD/BIOS out of sync

The FPGA, EPLD and BIOS versions could affect the leaf node's ability to bring up the modules as expected. If these are too far out of date, the interfaces of the switch could fail to come up. The user can validate the running and expected versions of FPGA, EPLD, and BIOS with the following moquery commands.

```
(none)# moquery -c firmwareCardRunning
Total Objects shown: 2

# firmware.CardRunning
```

```
biosVer        : v07.66(06/11/2019)
childAction    :
descr          :
dn             : sys/ch/supslot-1/sup/running
expectedVer    : v07.65(09/04/2018)  interimVer     : 14.2(1j)
internalLabel  :
modTs          : never
mode           : normal
monPolDn       : uni/fabric/monfab-default
operSt         : ok
rn             : running
status         :
ts             : 1970-01-01T00:00:00.000+00:00
type           : switch
version        : 14.2(1j)

# firmware.CardRunning
biosVer        : v07.66(06/11/2019)
childAction    :
descr          :
dn             : sys/ch/lcslot-1/lc/running
expectedVer    : v07.65(09/04/2018)  interimVer     : 14.2(1j)
internalLabel  :
modTs          : never
mode           : normal
monPolDn       : uni/fabric/monfab-default
operSt         : ok
rn             : running
status         :
ts             : 1970-01-01T00:00:00.000+00:00
type           : switch
version        : 14.2(1j)


(none)# moquery -c firmwareCompRunning
Total Objects shown: 2

# firmware.CompRunning
childAction    :
descr          :
dn             : sys/ch/supslot-1/sup/fpga-1/running
expectedVer    : 0x14  internalLabel  :
modTs          : never
mode           : normal
monPolDn       : uni/fabric/monfab-default
operSt         : ok
rn             : running
status         :
ts             : 1970-01-01T00:00:00.000+00:00
type           : controller
version        : 0x14

# firmware.CompRunning
childAction    :
descr          :
dn             : sys/ch/supslot-1/sup/fpga-2/runnin
expectedVer    : 0x4
internalLabel  :
modTs          : never
mode           : normal
monPolDn       : uni/fabric/monfab-default
operSt         : ok
rn             : running
status         :
```

```
ts              : 1970-01-01T00:00:00.000+00:00
type            : controller
version         : 0x4
```

If the running FPGA version does not match the expected FPGA version, it can be updated with the steps found in the chapter "Fabric discovery", section "Device replacement" under scenario "Leaf/Spine EPLD/FPGA not correct, F1582".

Check09 — SSL check

```
Check09 - SSL check [check]
SSL certificate details are valid
```

SSL communication is used between all fabric nodes to ensure encryption of control plane traffic. The SSL certificate used is installed during manufacturing and is generated based on the serial number of the chassis. The format of the subject should be as follows:

subject= /serialNumber=PID:N9K-C93xxxxx SN:FDOxxxxxxxx/CN=FDOxxxxxxxx

To validate SSL certificate during the discovery of a switch, use the following command.

```
(none)# cd /securedata/ssl && openssl x509 -noout -subject -in server.crt
subject= /serialNumber=PID:N9K-C93180YC-EX SN:FDO20432LH1/CN=FDO20432LH1
```

Note that the above will only work as non-root user if the switch node is still in discovery.

The chassis serial number can be found with the following command.

```
(none)# show inventory
NAME: "Chassis",  DESCR: "Nexus C93180YC-EX Chassis"
PID: N9K-C93180YC-EX     ,  VID: V00  ,  SN: FDO20160TPS
...
```

Additionally, the certificate must be valid at the current time. To view the valid dates of the certificate, use the '-dates' flag in the openssl command.

```
(none)# cd /securedata/ssl && openssl x509 -noout -dates -in server.crt
notBefore=Nov 28 17:17:05 2016 GMT
notAfter=Nov 28 17:27:05 2026 GMT
```

Check10 — Download policy

```
Check10 - Downloading policies [FAIL]
Registration to all PM shards is not complete
Policy download is not complete
```

Once the leaf has IP reachability to the APIC, it will download its configuration from the APIC and the APIC will acknowledge that the download is complete. The status of this process can be viewed with the following command.

```
(none)# moquery -c pconsBootStrap
Total Objects shown: 1

# pcons.BootStrap
allLeaderAcked         : no
```

```
allPortsInService      : yes
allResponsesFromLeader : yes
canBringPortInService  : no
childAction            :
completedPolRes        : no
dn                     : rescont/bootstrap
lcOwn                  : local
modTs                  : 2019-09-27T22:52:48.729+00:00
rn                     : bootstrap
state                  : completed
status                 :
timerTicks             : 360
try                    : 0
worstCaseTaskTry       : 0
```

Check11 — Time

```
Check11 - Checking time [ok]
2019-10-01 17:02:34
```

This check shows the user the current time. If there is too much delta between APIC and switch time, discovery could fail. On the APIC, the time can be checked with the date command.

```
apic1# date
Tue Oct  1 14:35:38 UTC 2019
```

Check12 — Module, PSU, fan check

For the switch to have connectivity to other devices, the modules need to be up and online. This can be validated via 'show module' and 'show environment' commands.

```
(none)# show module

Mod  Ports  Module-Type                         Model              Status
---  -----  ----------------------------------- ------------------ ----------
1    54     48x10/25G+6x40/100G Switch          N9K-C93180YC-EX    ok

Mod  Sw              Hw
---  --------------  ------
1    14.2(1j)        0.3050

Mod  MAC-Address(es)                         Serial-Num
---  --------------------------------------  ----------
1    e0-0e-da-a2-f2-83 to e0-0e-da-a2-f2-cb  FDO20160TPS

Mod  Online Diag Status
---  ------------------
1    pass

(none)# show environment
Power Supply:
Voltage: 12.0 Volts

Power                              Actual        Total
Supply    Model                    Output     Capacity    Status
                               (Watts )     (Watts )
-------  ------------------  -----------  -----------  --------------
1        NXA-PAC-650W-PI            0 W        650 W       shut
2        NXA-PAC-650W-PI          171 W        650 W        ok
```

```
                                  Actual        Power
Module     Model                     Draw    Allocated    Status
                                (Watts )     (Watts )
--------  -------------------  -----------  -----------  --------------
1         N9K-C93180YC-EX            171 W        492 W    Powered-Up
fan1      NXA-FAN-30CFM-B              N/A          N/A    Powered-Up
fan2      NXA-FAN-30CFM-B              N/A          N/A    Powered-Up
fan3      NXA-FAN-30CFM-B              N/A          N/A    Powered-Up
fan4      NXA-FAN-30CFM-B              N/A          N/A    Powered-Up

N/A - Per module power not available

Power Usage Summary:
--------------------
Power Supply redundancy mode (configured)                Non-Redundant(combined)
Power Supply redundancy mode (operational)               Non-Redundant(combined)

Total Power Capacity (based on configured mode)               650  W
Total Power of all Inputs (cumulative)                        650  W
Total Power Output (actual draw)                              171  W
Total Power Allocated (budget)                                N/A
Total Power Available for additional modules                  N/A

Fan:
------------------------------------------------------
Fan             Model                Hw         Status
------------------------------------------------------
Fan1(sys_fan1)  NXA-FAN-30CFM-B      --         ok
Fan2(sys_fan2)  NXA-FAN-30CFM-B      --         ok
Fan3(sys_fan3)  NXA-FAN-30CFM-B      --         ok
Fan4(sys_fan4)  NXA-FAN-30CFM-B      --         ok
Fan_in_PS1      --                   --         unknown
Fan_in_PS2      --                   --         ok
Fan Speed: Zone 1:  0x7f
Fan Air Filter : Absent

Temperature:
-----------------------------------------------------------------------------------
Module   Sensor                       MajorThresh   MinorThres   CurTemp     Status
                                    (Celsius)     (Celsius)    (Celsius)
-----------------------------------------------------------------------------------
1        Inlet(1)                      70            42           35         normal
1        outlet(2)                     80            70           37         normal
1        x86 processor(3)              90            80           38         normal
1        Sugarbowl(4)                  110           90           60         normal
1        Sugarbowl vrm(5)              120           110          50         normal
```

If a module is not coming online, reseat the module and check for FPGA, EPLD, or BIOS mismatches.

Example Broken Scenarios

Scenario 1 - First leaf does not appear in Fabric Membership

In this scenario, the user logs into APIC1 after completing the setup script and no switches have appeared in Fabric Membership. For the discovery of first leaf to occur successfully, the APIC should receive a DHCP Discover from the leaf in discovery phase.

Check that APIC1 is sending LLDP TLVs matching the parameters set in the setup script.

```
apic1# acidiag run lldptool out eth2-1
Chassis ID TLV
      MAC: e8:65:49:54:88:a1
Port ID TLV
      MAC: e8:65:49:54:88:a1
Time to Live TLV
      120
Port Description TLV
      eth2-1
System Name TLV
      apic1
System Description TLV
      topology/pod-1/node-1
Management Address TLV
      IPv4: 10.0.0.1
      Ifindex: 4
Cisco Port State TLV
      1
Cisco Node Role TLV
      0
Cisco Node ID TLV
      1
Cisco POD ID TLV
      1
Cisco Fabric Name TLV
      ACIFabric1
Cisco Appliance Vector TLV
      Id: 1
      IPv4: 10.0.0.1
      UUID: c67d1076-a2a2-11e9-874e-a390922be712
Cisco Node IP TLV
      IPv4:10.0.0.1
Cisco Port Role TLV
      2
Cisco Infra VLAN TLV
      3967
Cisco Serial Number TLV
      FCH1929V153
Cisco Authentication Cookie TLV
      1372058352
Cisco Standby APIC TLV
      0
End of LLDPDU TLV
```

Also validate that APIC1 is receiving LLDP from the directly connected leaf node.

```
apic1# acidiag run lldptool in eth2-1
Chassis ID TLV
      MAC: e0:0e:da:a2:f2:83
Port ID TLV
      Local: Eth1/1
Time to Live TLV
      120
Port Description TLV
      Ethernet1/1
System Name TLV
      switch
System Description TLV
      Cisco Nexus Operating System (NX-OS) Software 14.2(1j)
TAC support: http://www.cisco.com/tacCopyright (c) 2002-2020, Cisco Systems, Inc. All rights
reserved.
System Capabilities TLV
      System capabilities:  Bridge, Router
      Enabled capabilities: Bridge, Router
Management Address TLV
      MAC: e0:0e:da:a2:f2:83
      Ifindex: 83886080
Cisco 4-wire Power-via-MDI TLV
      4-Pair PoE supported
      Spare pair Detection/Classification not required
      PD Spare pair Desired State: Disabled
      PSE Spare pair Operational State: Disabled
Cisco Port Mode TLV
      0
Cisco Port State TLV
      1
Cisco Serial Number TLV
      FDO20160TPS
Cisco Model TLV
      N9K-C93180YC-EX
Cisco Firmware Version TLV
      n9000-14.2(1j)
Cisco Node Role TLV
      1
Cisco Infra VLAN TLV
      3967
Cisco Node ID TLV
      0
End of LLDPDU TLV
```

If APIC1 is receiving LLDP from the directly connected leaf node, the leaf should program the infra VLAN on the ports connected to the APIC. This VLAN programming can be validated via the 'show vlan encap-id <x>' command where 'x' is the configured infra VLAN.

```
(none)# show vlan encap-id 3967
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
8    infra:default                    active    Eth1/1

VLAN Type  Vlan-mode
---- ----- ----------
8    enet  CE
```

If the infra VLAN has not been programmed, check for wiring issues detected by the leaf node.

```
(none)# moquery -c lldpIf -f 'lldp.If.wiringIssues!=""'
Total Objects shown: 1

# lldp.If
id           : eth1/1
adminRxSt    : enabled
adminSt      : enabled
adminTxSt    : enabled
childAction  :
descr        :
dn           : sys/lldp/inst/if-[eth1/1]
lcOwn        : local
mac          : E0:0E:DA:A2:F2:83
modTs        : 2019-09-30T18:45:22.323+00:00
monPolDn     : uni/fabric/monfab-default
name         :
operRxSt     : enabled
operTxSt     : enabled
portDesc     :
portMode     : normal
```

```
portVlan     : unspecified
rn           : if-[eth1/1]
status       :
sysDesc      :
wiringIssues : infra-vlan-mismatch
```

When wiring issues attribute is set to 'infra-vlan-mismatch', the indication is that the leaf has learned of a different infra VLAN than the value which the APIC is sending (the APIC sent value can be verified using the command 'moquery -c lldpInst'). This scenario can occur if the leaf receives LLDP from a node that was once a part of another fabric. Essentially, a node in discovery will accept the first infra VLAN received via LLDP. To resolve this, remove the connections between this leaf and the other ACI nodes, except for the APIC, then clean reload the switch with 'acidiag touch clean' and 'reload' commands. Once the switch has booted, verify the correct infra VLAN is programmed. If this is true, connections can be restored to the other nodes and the user can proceed further with the ACI fabric set up.

Scenario 2 - Other APICs do not join the cluster

In this scenario, all fabric nodes have been discovered but APIC2 and 3 have not yet joined the APIC cluster.

Validate the setup script values across APICs. Values that must match are:

- Fabric domain
- Fabric ID
- TEP pool
- Infra VLAN
- GIPo
- Cluster size
- Firmware version

```
apic1# cat /data/data_admin/sam_exported.config
Setup for Active and Standby APIC

fabricDomain = ACIFabric1
fabricID = 1
systemName =apic1
controllerID = 1
tepPool = 10.0.0.0/16
infraVlan = 3967
GIPo = 225.0.0.0/15
clusterSize = 3
standbyApic = NO
enableIPv4 = Y
enableIPv6 = N
firmwareVersion = 4.2(1j)
ifcIpAddr = 10.0.0.1
apicX = NO
podId = 1
oobIpAddr = 10.48.22.69/24
```

Verify common issues with 'acidiag cluster' command on all 3 APICs.

```
apic1# acidiag cluster
Admin password:
```

```
Product-name = APIC-SERVER-M1
Serial-number = FCH1906V1XV
Running...

Checking Core Generation: OK
Checking Wiring and UUID: OK
Checking AD Processes: Running
Checking All Apics in Commission State: OK
Checking All Apics in Active State: OK
Checking Fabric Nodes: OK
Checking Apic Fully-Fit: OK
Checking Shard Convergence: OK
Checking Leadership Degration: Optimal leader for all shards
Ping OOB IPs:
APIC-1: 10.48.22.69 - OK
APIC-2: 10.48.22.70 - OK
APIC-3: 10.48.22.71 - OK
Ping Infra IPs:
APIC-1: 10.0.0.1 - OK
APIC-2: 10.0.0.2 - OK
APIC-3: 10.0.0.3 - OK
Checking APIC Versions: Same (4.2(1j))
Checking SSL: OK

Done!
```

Finally, use 'avread' to validate if these settings match across all APICs. Note that this is a different command from the typical 'acidiag avread' which shows similar output, but it is parsed for easier consumption.

```
apic1# avread
Cluster:
-------------------------------------------------------------------------
fabricDomainName        ACIFabric1
discoveryMode           PERMISSIVE
clusterSize             3
version                 4.2(1j)
drrMode                 OFF
operSize                3

APICs:
-------------------------------------------------------------------------
                APIC 1                  APIC 2                  APIC 3
version           4.2(1j)                 4.2(1j)                 4.2(1j)
address           10.0.0.1                10.0.0.2                10.0.0.3
oobAddress        10.48.22.69/24          10.48.22.70/24          10.48.22.71/24
routableAddress   0.0.0.0                 0.0.0.0                 0.0.0.0
tepAddress        10.0.0.0/16             10.0.0.0/16             10.0.0.0/16
podId             1                       1                       1
chassisId         3c9e5024-.-5a78727f     573e12c0-.-6b8da0e5     44c4bf18-.-20b4f52&
cntrlSbst_serial  (APPROVED,FCH1906V1XV)  (APPROVED,FCH1921V1Q9)  (APPROVED,FCH1906V1PW)
active            YES                     YES                     YES
flags             cra-                    cra-                    cra-
health            255                     255                     255
apic1#
```

Scenario 3 - Spine does not appear in Fabric Membership

In this scenario, the first leaf has been discovered in the fabric but no spines have appeared for discovery under the Fabric Membership submenu.

Validate physical connectivity from leaf to spine. In the example below, the leaf switch is connected to a spine via interface e1/49.

```
leaf101# show int eth1/49
Ethernet1/49 is up
admin state is up, Dedicated Interface
 Hardware: 1000/10000/100000/40000 Ethernet, address: 0000.0000.0000 (bia e00e.daa2.f3f3)
 MTU 9366 bytes, BW 100000000 Kbit, DLY 1 usec
 reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation ARPA, medium is broadcast
 Port mode is routed
 full-duplex, 100 Gb/s
...
```

If the port is in an **out-of-service** status, check on the spine that LLDP has been received from the directly connected leaf.

```
(none)# show lldp neighbors
Capability codes:
 (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
 (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID            Local Intf      Hold-time  Capability  Port ID
leaf102               Eth2/27         120        BR          Eth1/53
leaf103               Eth2/29         120        BR          Eth1/49
leaf101               Eth2/32         120        BR          Eth1/51
Total entries displayed: 3
```

Another validation is to verify that there is no version difference between leaf and spine. If there is, remediate the situation by copying the newer version to /bootflash of the spine. Then, configure the switch to boot to the software with the following commands:

```
(none)# ls -alh /bootflash
total 3.0G
drwxrwxr-x  3 root admin 4.0K Oct  1 20:21 .
drwxr-xr-x 50 root root  1.3K Oct  1 00:22 ..
-rw-r--r--  1 root root  3.5M Sep 30 21:24 CpuUsage.Log
-rw-rw-rw-  1 root root  1.7G Sep 27 14:50 aci-n9000-dk9.14.2.1j.bin
-rw-r--r--  1 root root  1.4G Sep 27 21:20 auto-s
-rw-rw-rw-  1 root root     2 Sep 27 21:25 diag_bootup
-rw-r--r--  1 root root    54 Oct  1 20:20 disk_log.txt
-rw-rw-rw-  1 root root   693 Sep 27 21:23 libmon.logs
drwxr-xr-x  4 root root  4.0K Sep 26 15:24 lxc
-rw-r--r--  1 root root  384K Oct  1 20:20 mem_log.txt
-rw-r--r--  1 root root  915K Sep 27 21:10 mem_log.txt.old.gz
-rw-rw-rw-  1 root root   12K Sep 27 21:17 urib_api_log.txt
```

```
(none)# setup-bootvars.sh aci-n9000-dk9.14.2.1j.bin
In progress
In progress
In progress
In progress
Done
```

If the new image is continuously removed from bootflash, ensure that the folder is less than half full by removing older images or auto-s file; check the space utilization by using 'df -h' on the switch.

After setting the boot variable, reload the switch and it should boot to the new version.

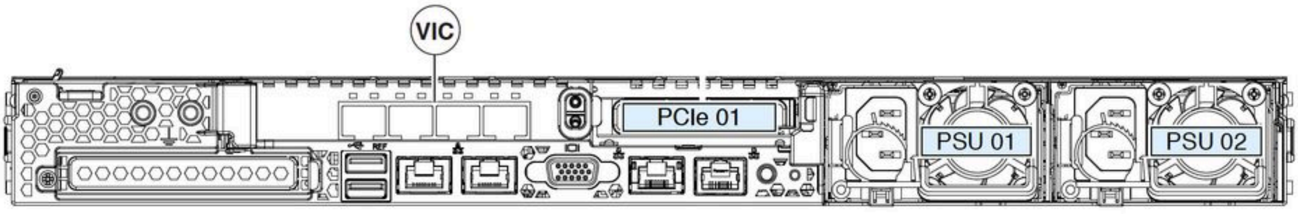
FPGA, EPLD, and BIOS validation might be required after the reload. Please refer to the sub-section "Leaf/Spine EPLD/FPGA not correct, F1582" for further troubleshooting on this matter.

Scenario 4 - After initial fabric discovery, cluster is flapping between fully-fit and degraded

If this is happening after a new fabric setup, it can be caused by incorrect cabling of the APIC-M3 or APIC-L3 connecting into the fabric. You can confirm such incorrect cabling by executing "show lldp neighbors" on both leaf switches connected to the APIC. You will notice after executing this multiple times that both leaf switches are seeing the same APIC interface.

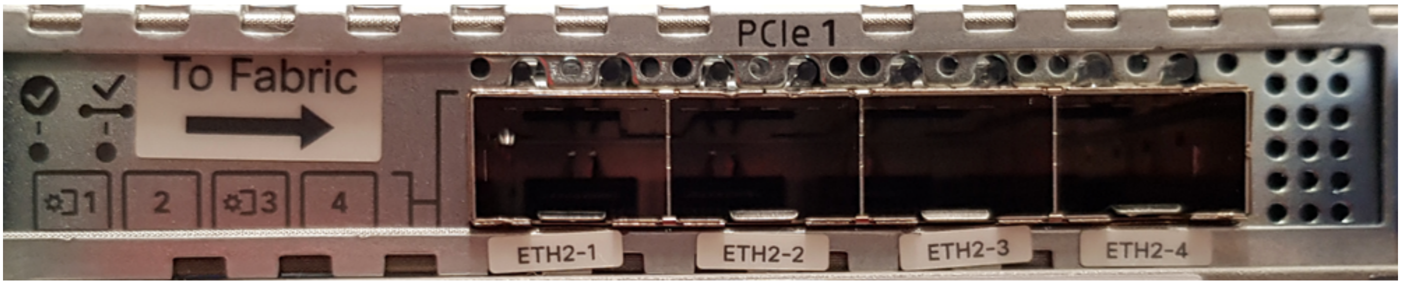
The back of an APIC-M3/L3 server looks like the following:

Rear-view of APIC-M3/L3 server



Note that for an APIC-M3/L3, the VIC card has 4 ports: ETH2-1, ETH2-2, ETH2-3, and ETH2-4, as seen below:

View of APIC VIC 1455 with labels

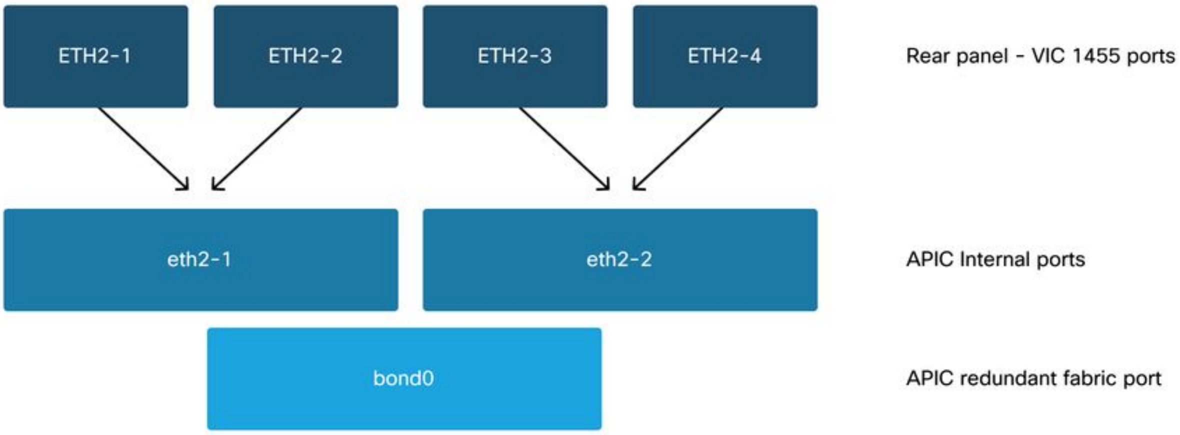


The rules to connect the APIC server to leaf switches are as follows:

- All ports must have the same speed, either 10-Gigabit or 25-Gigabit.
- ETH2-1 and ETH2-2 is one port-channel pair, corresponding to eth2-1 ('ifconfig' output) from the APIC OS.
- ETH2-3 and ETH2-4 is the other port-channel pair, corresponding to eth2-2 ('ifconfig' output) on APIC OS.
- Only one connection is allowed per port-channel pair. For example, connect one cable to either ETH2-1 or ETH2-2, and connect another cable to either ETH2-3 or ETH2-4 (**Never connect both ETHs in a port channel pair. This will lead to fabric discovery issues.**).

For further understanding, the following is a representation of the VIC port mapping to APIC bond.

VIC 1455 ports — APIC redundant fabric port


ETH2-1
ETH2-2
ETH2-3
ETH2-4
Rear panel - VIC 1455 ports
eth2-1
eth2-2
APIC Internal ports
bond0
APIC redundant fabric port