

ACIファブリック検出のトラブルシューティング ：初期ファブリック設定

内容

[概要](#)

[背景説明](#)

[ファブリック検出ワークフロー](#)

[Check01：システム状態](#)

[Check02:DHCPステータス](#)

[Check03:AVの詳細](#)

[Check04:APICへのIP到達可能性](#)

[Check05：インフラストラクチャVLAN](#)

[Check06:LLDP隣接関係](#)

[Check07：スイッチバージョン](#)

[Check08:FPGA/EPLD/BIOSが同期していません。](#)

[Check09:SSLチェック](#)

[Check10：ポリシーのダウンロード](#)

[Check11：時間](#)

[Check12：モジュール、PSU、ファンチェック](#)

[壊れたシナリオの例](#)

[シナリオ1：最初のリーフがファブリックメンバーシップに表示されない](#)

[シナリオ2 – 他のAPICがクラスタに参加しない](#)

[シナリオ3：スパインがファブリックメンバーシップに表示されない](#)

[シナリオ4：最初のファブリック検出後に、クラスタが完全に適合するか機能低下するかを切り替える](#)

概要

このドキュメントでは、最初のファブリック検出プロセスを理解し、トラブルシューティングする手順について説明します。たとえば、問題のシナリオを説明します。

背景説明

このドキュメントの内容は、[Troubleshooting Cisco Application Centric Infrastructure, Second Edition](#) 特に [ファブリック検出：初期ファブリック設定](#) 章

ファブリック検出ワークフロー

ACIファブリック検出プロセスは、特定のイベントシーケンスに従います。基本的な手順は次のとおりです。

1. 最初のAPICのKVMコンソールに接続し、ファブリック名、APICクラスタサイズ、トンネル

エンドポイント(TEP)アドレスプールなどの値を入力して**セットアップスクリプト**を完了します。

- 完了すると、APIC1はファブリックポート経由でLLDPの送信を開始します。LLDPパケットには、**infra VLAN**などの情報と、APIC (コントローラとも呼ばれる)としての役割を持つ特別なTLVが含まれています。
- APIC1からこれらのLLDPパケットを受信すると、リーフはAPICが検出されたすべてのポートでインフラVLANをプログラムします。
- リーフは、既知のインフラストラクチャVLANでDHCP Discoverの送信を開始します。
- ユーザはHTTPS経由でAPIC1の**OOB IP**にログインし、最初のリーフノードを**ファブリックメンバーシップ**サブメニューに登録します。
- リーフに**ノードID**が割り当てられると、APIC1は設定された**TEPアドレスプール**からのIPアドレスで応答し、DHCPプロセスが完了します。
- 登録済みリーフは、LLDPを介して検出された他の直接接続されたスパインからAPIC1にDHCP Discoverをリレーします。
- ユーザは、動的に検出されたスパインが[Fabric Membership]サブメニューに表示されることを確認し、それらを登録できます。
- スパインが登録されると、APIC1はTEPプールからのIPアドレスで応答し、これらのノードのDHCPが完了します。
- スパインリレーDHCPは、pod1の他のすべてのノードから検出します。(これは、推奨されているように、スパインとリーフスイッチの間にフルメッシュがあることを前提としており、これが一般的なアーキテクチャです)。
- 他のAPICに接続されたリーフノードが登録されると、APICクラスタは相互のTCP通信を介して確立されます。APIC2とAPIC3の**セットアップダイアログ**が完了していることを確認します。
- すべてのAPICがクラスタを形成し、完全に適合していることを確認します。この場合、**ファブリック検出**は完了です。

4.2以降では、ファブリックノードで新しいCLIコマンドが使用できるようになり、一般的な検出問題の診断に役立ちます。次のセクションでは、実行されたチェックについて説明し、障害のトラブルシューティングに役立つ追加の検証コマンドを提供します。

```
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 : システム状態

リーフがノードIDを割り当てられ、ファブリックに登録されると、そのブートストラップのダウンロードが開始され、**in-service**状態に移行します。

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

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

リーフの現在の状態を検証するには、**moquery -c topSystem**を実行します

```
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ステータス

```
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
```

リーフは、APIC1からDHCP経由でTEPアドレスを受信し、他のAPICへのIP接続を確立する必要があります。リーフの物理TEP(PTEP)はloopback0に割り当てられます。アドレスが割り当てられていない場合、ユーザはリーフがtcpdumpユーティリティを使用してDHCP Discoverを送信していることを検証できます。これに関しては、すべてのCPUインバンドコントロールプレーンネットワークトラフィックを確認できるインターフェイスkpm_inbを使用します。

```
(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
```

ユーザは、dhcpdがAPICで実行され、bond0サブインターフェイスでリッスンしていることを検証することもできます。ボンドインターフェイスは、APICポートに面するファブリックを表します。bond0.<infra VLAN>の形式を使用します。

```
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の詳細

Check03 - AV details check [ok]

リーフは、登録されたAPICにTEPプールの有効な範囲のIPがあるかどうかを検証します。APIC情報がまだ記録されていない場合は、このチェックに合格します。ユーザは、「acidiag avread」コマンドを使用して、リーフノードの観点から現在のAPIC情報を表示できます。次の例では、leaf/spineプロンプトに(none)#が表示されている場合、リーフ/スパインがまだファブリックのメンバではないことを示しています。

```
(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:APICへのIP到達可能性

リーフはIPアドレスを受信すると、APICとのTCPセッションの確立を試み、設定のダウンロード処理を開始します。ユーザは、「iping」ユーティリティを使用してAPICへのIP接続を検証できます。

```
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 : インフラストラクチャVLAN

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

インフラVLANチェックは、ノードがAPICが存在するポッドに接続されている場合にのみ成功します。そうでない場合は、チェックが失敗することが予想されるため、ユーザはメッセージを無視できます。

リーフは、他のACIノードから受信したLLDPパケットに基づいてインフラストラクチャVLANを決定します。最初に受信したパケットは、スイッチが検出中に受け入れられます。

```
(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
```

APICに接続されたスイッチポートインターフェイスでinfra VLANがプログラムされていない場合は、リーフによって検出された配線の問題をチェックします。

```
(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隣接関係

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

どのポートが他のACIデバイスに接続しているかを判別するには、リーフが他のファブリックノードからLLDPを受信する必要があります。LLDPが受信されたことを確認するには、[show lldp neighbors]をオンにします。

```
(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        (R)         eth2-1
apic2          Eth1/2          120        (R)         eth2-1
switch         Eth1/51         120        (B)         Eth2/32
switch         Eth1/54         120        (B)         Eth1/25
```

Total entries displayed: 4

Check07 : スイッチバージョン

Check07 - Switch version [ok]

version: n9000-14.2(1j) and apic version: 4.2(1j)

APICとリーフのバージョンが異なる場合、ファブリック検出が失敗する可能性があります。リーフで実行されているバージョンを検証するには、「show version」または「vsh -c 'show version'」を使用します。

```
(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.html Copyright (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]
```

```
...
```

同じコマンドがAPICでも機能します。

```
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が同期していません。

FPGA、EPLD、およびBIOSのバージョンは、リーフノードのモジュールを正常に起動する機能に影響を与える可能性があります。これらが古すぎると、スイッチのインターフェイスが起動できなくなる可能性があります。ユーザは、次のmoqueryコマンドを使用して、FPGA、EPLD、およびBIOSの実行バージョンと予想バージョンを検証できます。

(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
```

実行中のFPGAのバージョンが予想されるFPGAのバージョンと一致しない場合は、「Leaf/Spine EPLD/FPGA not correct, F1582」のシナリオの「Fabric discovery」セクション「Device replacement」の章にある手順で更新できます。

Check09:SSLチェック

```
Check09 - SSL check [check]
```

```
SSL certificate details are valid
```

すべてのファブリックノード間でSSL通信を使用して、コントロールプレーントラフィックを確実に暗号化します。使用されるSSL証明書は製造時にインストールされ、シャーシのシリアル番号に基づいて生成されます。件名の形式は次のとおりです。

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

スイッチの検出中にSSL証明書を検証するには、次のコマンドを使用します。

```
(none)# cd /securedata/ssl && openssl x509 -noout -subject -in server.crt
```

```
subject= /serialNumber=PID:N9K-C93180YC-EX SN:FDO20432LH1/CN=FDO20432LH1
```

上記のコマンドは、スイッチノードがまだ検出中の場合にのみ、非ルートユーザとして機能することに注意してください。

シャーシのシリアル番号は、次のコマンドで確認できます。

```
(none)# show inventory
```

```
NAME: "Chassis", DESCR: "Nexus C93180YC-EX Chassis"
```

```
PID: N9K-C93180YC-EX , VID: V00 , SN: FDO20160TPS
```

```
...
```

また、証明書は現在の時刻に有効である必要があります。証明書の有効な日付を表示するには、opensslコマンドで「-dates」フラグを使用します。

```
(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 : ポリシーのダウンロード

```
Check10 - Downloading policies [FAIL]
```

```
Registration to all PM shards is not complete
```

```
Policy download is not complete
```

リーフがAPICにIP到達可能性を持つと、APICから設定がダウンロードされ、APICはダウンロードが完了したことを認識します。このプロセスのステータスは、次のコマンドで表示できます。

```
(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 : 時間

```
Check11 - Checking time [ok]
```

```
2019-10-01 17:02:34
```

このチェックでは、ユーザに現在の時刻が表示されます。APICとスイッチの時間の差が大きすぎると、検出が失敗する可能性があります。APICでは、dateコマンドで時刻を確認できます。

```
apic1# date
```

```
Tue Oct 1 14:35:38 UTC 2019
```

Check12 : モジュール、PSU、ファンチェック

スイッチが他のデバイスに接続できるようにするには、モジュールが稼働していてオンラインになっている必要があります。これは、「show module」および「show environment」コマンドで検証できます。

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

モジュールがオンラインにならない場合は、モジュールを取り付け直し、FPGA、EPLD、またはBIOSの不一致を確認します。

壊れたシナリオの例

シナリオ1：最初のリーフがファブリックメンバーシップに表示されない

このシナリオでは、ユーザはセットアップスクリプトの完了後にAPIC1にログインし、ファブリ

ックメンバーシップにスイッチが表示されません。最初のリーフの検出が正常に行われるように、APICは検出フェーズのリーフからDHCP Discoverを受信する必要があります。

APIC1がセットアップスクリプトで設定されたパラメータに一致するLLDP TLVを送信していることを確認します。

```
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
```

また、APIC1が直接接続されたリーフノードからLLDPを受信していることも検証します。

```
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/tac Copyright (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

```

APIC1が直接接続されたリーフノードからLLDPを受信している場合、リーフはAPICに接続されたポート上でインフラVLANをプログラムする必要があります。このVLANプログラミングは、「show vlan encap-id <x>」コマンドを使用して検証できます。ここで、「x」は設定済みのインフラストラクチャVLANです。

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

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

```
VLAN Type Vlan-mode
```

```
8 enet CE
```

インフラストラクチャVLANがプログラムされていない場合は、リーフノードによって検出された配線の問題を確認します。

```
(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
```

wiring issues属性が「infra-vlan-mismatch」に設定されている場合、リーフはAPICが送信している値とは異なるインフラストラクチャVLANを学習したことを示しています (APICの送信値はコマンド「moquery -c lldpInst」を使用して確認できます)。このシナリオは、かつて別のファブリックの一部であったノードからリーフがLLDPを受信した場合に発生します。基本的に、ディスクバリ中のノードは、LLDPを介して受信した最初のインフラストラクチャVLANを受け入れます。これを解決するには、このリーフと他のACIノード (APICを除く) 間の接続を削除してから、「acidiag touch clean」および「reload」コマンドを使用してスイッチをクリーンリロードします。スイッチが起動したら、正しいインフラストラクチャVLANがプログラムされていることを確認します。これが当てはまる場合、他のノードへの接続を復元でき、ユーザはACIファブリックのセットアップをさらに進めることができます。

シナリオ2 – 他のAPICがクラスタに参加しない

このシナリオでは、すべてのファブリックノードが検出されましたが、APIC2と3はまだAPICクラスタに参加していません。

APIC間でセットアップスクリプトの値を検証します。一致する必要がある値は次のとおりです。

- ファブリックドメイン
- ファブリックID
- TEPプール
- インフラVLAN
- GIPo
- クラスタ サイズ
- ファームウェア バージョン

```
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
```

3つすべてのAPICで「acidiag cluster」コマンドを使用して、一般的な問題を確認します。

```
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!
```

最後に、「avread」を使用して、これらの設定がすべてのAPICで一致するかどうかを検証します。このコマンドは、同様の出力を示す一般的な「acidiag avread」とは異なるコマンドですが、より簡単に使用できるように解析されています。

```
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#
```

シナリオ3 : スパインがファブリックメンバーシップに表示されない

このシナリオでは、最初のリーフはファブリック内で検出されていますが、[Fabric Membership]サブメニューに検出用のスパインが表示されていません。

リーフからスパインへの物理的な接続を検証します。次の例では、リーフスイッチはインターフェイスe1/49を介してスパインに接続されています。

```
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
```

...

ポートがout-of-serviceステータスの場合は、直接接続されたリーフからLLDPを受信したことをスパインで確認します。

```
(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
```

もう1つの検証は、リーフとスパインのバージョンに違いがないことを確認することです。存在する場合は、スパインの/bootflashに新しいバージョンをコピーして状況を修正します。次に、次のコマンドを使用して、ソフトウェアをブートするようにスイッチを設定します。

```
(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

新しいイメージがブートフラッシュから継続的に削除される場合は、古いイメージまたはauto-sファイルを削除して、フォルダが半分以下であることを確認します。スイッチで「df -h」を使用して、スペースの使用率を確認します。

ブート変数を設定した後、スイッチをリロードすると、新しいバージョンでブートします。

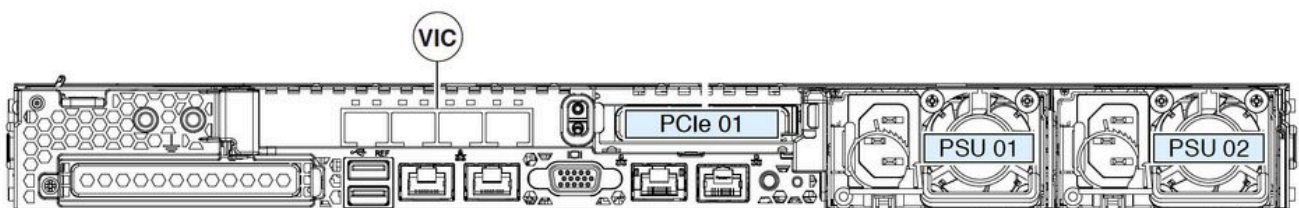
リロード後に、FPGA、EPLD、およびBIOSの検証が必要になる場合があります。この問題に関するさらなるトラブルシューティングについては、サブセクション「リーフ/スパイン EPLD/FPGAが正しくありません、F1582」を参照してください。

シナリオ4：最初のファブリック検出後に、クラスタが完全に適合するか機能低下するかを切り替える

新しいファブリックの設定後にこの問題が発生する場合は、APIC-M3またはAPIC-L3がファブリックに接続されているケーブルが正しくないことが原因である可能性があります。APICに接続された両方のリーフスイッチで「show lldp neighbors」を実行すると、このようなケーブル接続の誤りを確認できます。これを複数回実行すると、両方のリーフスイッチに同じAPICインターフェイスが表示されます。

APIC-M3/L3サーバの背面は次のようになります。

APIC-M3/L3サーバの背面図



APIC-M3/L3の場合、VICカードには4つのポートがあります。次に示すように、ETH2-1、ETH2-2、ETH2-3、およびETH2-4。

ラベル付きAPIC VIC 1455の表示



APICサーバをリーフスイッチに接続するルールは次のとおりです。

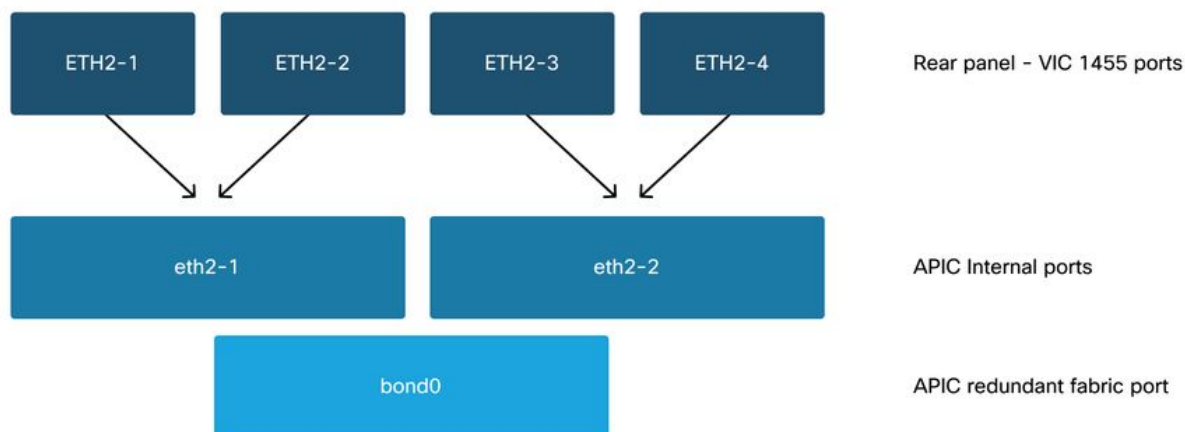
- すべてのポートの速度は、10ギガビットまたは25ギガビットのどちらかの同じである必要が

あります。

- ETH2-1とETH2-2は、APIC OSからのeth2-1 ('ifconfig' output)に対応する1つのポートチャンネルペアです。
- ETH2-3とETH2-4は他のポートチャンネルペアで、APIC OSのeth2-2 ('ifconfig' output)に対応します。
- ポートチャンネルペアごとに許可される接続は1つだけです。たとえば、1本のケーブルをETH2-1またはETH2-2に接続し、もう1本のケーブルをETH2-3またはETH2-4に接続します(ポートチャンネルペアの両方のETHを接続しないでください)。これにより、ファブリック検出の問題が発生します。)

さらに理解を深めるために、APIC結合へのVICポートマッピングを次に示します。

VIC 1455ポート : APIC冗長ファブリックポート



翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。