

# Identificar e solucionar problemas da ACI Fabric Discovery - Configuração inicial da estrutura

## Contents

[Introduction](#)

[Informações de Apoio](#)

[Fluxo de trabalho de descoberta de malha](#)

[Check01 — Estado do sistema](#)

[Check02 — Status do DHCP](#)

[Verificação 03 — Detalhes AV](#)

[Check04 — Acessibilidade IP para APIC](#)

[Check05 — Infra VLAN](#)

[Check06 — Adjacência LLDP](#)

[Check07 — Versão do switch](#)

[Check08 — FPGA/EPLD/BIOS não sincronizado](#)

[Check09 — Verificação SSL](#)

[Check10 — Política de download](#)

[Cheque11 — Hora](#)

[Check12 — Verificação do módulo, da PSU e do ventilador](#)

[Exemplo de cenários desfeitos](#)

[Cenário 1 - A primeira folha não aparece na Associação de Estrutura](#)

[Cenário 2 - Outros APICs não participam do cluster](#)

[Cenário 3 - A lombada não aparece na associação de estrutura](#)

[Cenário 4 - Após a descoberta inicial da malha, o cluster está oscilando entre totalmente adequado e degradado](#)

## Introduction

Este documento descreve as etapas para entender e solucionar problemas do processo inicial de descoberta de estrutura, incluindo exemplos de situações de problema.

## Informações de Apoio

O material deste documento foi extraído do [Solução de problemas da Cisco Application Centric Infrastructure, segunda edição](#) livro, especificamente o **Fabric Discovery - Configuração inicial da malha** capítulo.

## Fluxo de trabalho de descoberta de malha

O processo de descoberta de estrutura da ACI segue uma sequência específica de eventos. As etapas básicas são as seguintes:

1. Conecte-se ao **console KVM** do primeiro APIC e conclua o **script de configuração** inserindo

valores como o nome da malha, o tamanho do cluster APIC e o pool de endereços do ponto de extremidade do túnel (TEP).

2. Depois de concluído, o APIC1 começará a enviar o **LLDP** por meio de suas portas de estrutura. Os pacotes LLDP contêm TLVs especiais com informações como a **infra VLAN** e sua função como um APIC (também conhecido como controlador).
3. Ao receber esses pacotes LLDP do APIC1, o leaf programará a infra VLAN em todas as portas onde um APIC for detectado.
4. O leaf começa a enviar DHCP Discovers na infra VLAN agora conhecida.
5. O usuário faz login no **IP OOB** do APIC1 via HTTPS e registra o primeiro nó folha no submenu **Fabric Membership**.
6. Depois que a folha receber um **ID de nó**, o APIC1 responderá com um endereço IP do **pool de endereços TEP configurado** e o processo DHCP será concluído.
7. A folha registrada retransmite DHCP Discovers de outros spines diretamente conectados que foram descobertos via LLDP para o APIC1.
8. O usuário verá esses spines descobertos dinamicamente aparecerem no submenu Fabric Membership (Associação de estrutura) e poderá registrá-los.
9. Depois que os spines são registrados, o APIC1 responde com um endereço IP do pool TEP e o DHCP é concluído para esses nós.
10. Os spines relay DHCP Descobre de todos os outros nós do pod1. (Isso pressupõe que haja uma malha completa entre spines e switches leaf, como é recomendado e é a arquitetura típica).
11. Depois que os nós de folha conectados aos outros APICs são registrados, o cluster APIC pode ser estabelecido através da comunicação TCP entre eles. Conclua o diálogo de configuração no APIC2 e APIC3.
12. Confirme se todos os APICs formaram um cluster e se encaixam completamente. Se esse for o caso, a detecção de estrutura está completa.

A partir do 4.2, um novo comando CLI está disponível nos nós de estrutura para auxiliar no diagnóstico de problemas comuns de descoberta. As seções a seguir abordarão as verificações realizadas e fornecerão comandos de validação adicionais para ajudar na solução de problemas de falhas.

```
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 — Estado do sistema

Quando a folha tiver um ID de nó alocado e estiver registrada na estrutura, ela começará a fazer o download do bootstrap e, em seguida, a transição para um estado **em serviço**.

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

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

Para validar o estado atual do leaf, o usuário pode executar **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 — Status do DHCP

```
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
```

O leaf precisa receber um endereço TEP via DHCP do APIC1 e, em seguida, estabelecer conectividade IP com os outros APICs. O **TEP Físico (PTEP)** da folha é atribuído ao loopback0. Se nenhum endereço for atribuído, o usuário poderá validar se a folha está enviando um DHCP Discover com o utilitário **tcpdump**. Observe que para isso usaremos a interface **kpm\_inb**, que permite que você veja todo o tráfego de rede do plano de controle da banda da CPU.

```
(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
```

O usuário também pode validar se o **dhcpd** está sendo executado no APIC e ouvindo na subinterface **bond0**. A interface de ligação representa a malha voltada para as portas APIC. Usaremos o formato **bond0.<infra VLAN>**.

```
apicl# 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
```

## Verificação 03 — Detalhes AV

Check03 - AV details check [ok]

A folha será validada se o APIC registrado tiver um IP em um intervalo válido para o pool TEP. Se nenhuma informação do APIC tiver sido registrada ainda, essa verificação será aprovada. O usuário pode ver as informações atuais do APIC da perspectiva do nó folha por meio do comando 'acidiag avread'. Observe no exemplo abaixo que quando o prompt leaf/spine está mostrando (none)#, essa é uma indicação de que leaf/spine ainda não é um membro da estrutura.

```
(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 — Acessibilidade IP para APIC

Quando a folha receber um endereço IP, ela tentará estabelecer sessões TCP com o APIC e iniciar o processo de download de sua configuração. O usuário pode validar a conectividade IP para o APIC usando o utilitário 'iping'.

```

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]

A verificação de VLAN infra só será bem-sucedida se o nó estiver conectado a um Pod onde exista um APIC. Se esse não for o caso, o usuário poderá ignorar a mensagem porque a verificação deverá falhar.

O leaf determinará a infra VLAN com base nos pacotes LLDP recebidos de outros nós da ACI. O primeiro que ele receber será aceito quando o switch estiver em descoberta.

```

(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
```

Se a VLAN infra não tiver sido programada nas interfaces da porta do switch conectadas aos APICs, verifique os problemas de fiação detectados pelo 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 — Adjacência LLDP

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

Para determinar quais portas se conectam a outros dispositivos da ACI, a folha deve receber o LLDP de outros nós de estrutura. Para validar o LLDP recebido, o usuário pode marcar '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 — Versão do switch

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

Se as versões do APIC e das folhas não forem as mesmas, a detecção de estrutura poderá falhar. Para validar a versão em execução no leaf, use 'show version' ou '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.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]
...
```

O mesmo comando também funcionará nos 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 não sincronizado

As versões FPGA, EPLD e BIOS podem afetar a capacidade do nó de folha de ativar os módulos conforme esperado. Se eles estiverem muito desatualizados, as interfaces do switch podem falhar ao serem ativadas. O usuário pode validar as versões atual e esperada do FPGA, EPLD e BIOS com os seguintes comandos moquery.

(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
```

Se a versão atual do FPGA não corresponder à versão esperada do FPGA, ela poderá ser atualizada com as etapas encontradas no capítulo "Fabric discovery", seção "Device replacement" (Substituição de dispositivos) no cenário "Leaf/Spine EPLD/FPGA not correct, F1582" (A versão do FPGA em execução não corresponde à versão esperada do FPGA).

## Check09 — Verificação SSL

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

A comunicação SSL é usada entre todos os nós de estrutura para garantir a criptografia do tráfego do plano de controle. O certificado SSL usado é instalado durante a fabricação e é gerado com base no número de série do chassi. O formato do assunto deve ser o seguinte:

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

Para validar o certificado SSL durante a descoberta de um switch, use o comando a seguir.

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

Observe que o item acima só funcionará como usuário não raiz se o nó do switch ainda estiver em descoberta.

O número de série do chassi pode ser encontrado com o seguinte comando:

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

Além disso, o certificado deve ser válido no momento. Para exibir as datas válidas do certificado, use o sinalizador '-dates' no comando openssl.

```
(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 — Política de download

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

Quando a folha tiver acesso IP ao APIC, ela baixará sua configuração do APIC e o APIC confirmará que o download foi concluído. O status desse processo pode ser visualizado com o seguinte comando.

```
(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
```

## Cheque11 — Hora

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

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

Essa verificação mostra o usuário na hora atual. Se houver muito delta entre o APIC e o tempo do switch, a detecção poderá falhar. No APIC, a hora pode ser verificada com o comando `date`.

```
apic1# date
```

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

## Check12 — Verificação do módulo, da PSU e do ventilador

Para que o switch tenha conectividade com outros dispositivos, os módulos precisam estar ativos e on-line. Isso pode ser validado por meio dos comandos `'show module'` e `'show environment'`.

```
(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

Se um módulo não estiver on-line, recoloque-o e verifique se há incompatibilidades de FPGA, EPLD ou BIOS.

## Exemplo de cenários desfeitos

### Cenário 1 - A primeira folha não aparece na Associação de Estrutura

Neste cenário, o usuário faz login no APIC1 após concluir o script de configuração e nenhum switch apareceu na Fabric Membership. Para que a descoberta da primeira folha ocorra com

êxito, o APIC deve receber uma descoberta DHCP da folha na fase de descoberta.

Verifique se o APIC1 está enviando TLVs LLDP correspondentes aos parâmetros definidos no script de configuração.

```
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
```

Confirme também se o APIC1 está recebendo LLDP do nó de folha diretamente conectado.

```
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

```

Se o APIC1 estiver recebendo LLDP do nó de folha diretamente conectado, o leaf deve programar a infra VLAN nas portas conectadas ao APIC. Essa programação de VLAN pode ser validada por meio do comando 'show vlan encap-id <x>', onde 'x' é a infra VLAN configurada.

```

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

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

```

Se a VLAN infra não tiver sido programada, verifique se há problemas de fiação detectados pelo nó folha.

```

(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
```

Quando o atributo de problemas de fiação é definido como 'infra-vlan-mismatch', a indicação é que a folha aprendeu de uma infra VLAN diferente do valor que o APIC está enviando (o valor enviado do APIC pode ser verificado usando o comando 'moquery -c lldplnst'). Esse cenário pode ocorrer se a folha receber LLDP de um nó que já foi parte de outra estrutura. Essencialmente, um nó na descoberta aceitará a primeira infra VLAN recebida via LLDP. Para resolver isso, remova as conexões entre esta folha e os outros nós da ACI, exceto o APIC, e limpe e recarregue o switch com os comandos 'acidiag touch clean' e 'reload'. Depois que o switch for inicializado, verifique se a infra VLAN correta está programada. Se isso for verdadeiro, as conexões podem ser restauradas para os outros nós e o usuário pode prosseguir com a configuração da estrutura da ACI.

## Cenário 2 - Outros APICs não participam do cluster

Neste cenário, todos os nós de estrutura foram descobertos, mas o APIC2 e o 3 ainda não ingressaram no cluster do APIC.

Valide os valores do script de configuração nos APICs. Os valores que devem corresponder são:

- Domínio de estrutura
- ID da malha
- pool de TEP
- Infra VLAN
- GIPo
- Tamanho do cluster
- Versão do firmware

```
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
```

Verifique problemas comuns com o comando 'acidiag cluster' em todos os 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!
```

Finalmente, use 'avread' para validar se essas configurações correspondem em todos os APICs. Observe que este é um comando diferente do típico 'acidiag avread' que mostra saída semelhante, mas é analisado para consumo mais fácil.

```
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#
```

### Cenário 3 - A lombada não aparece na associação de estrutura

Neste cenário, a primeira folha foi descoberta na estrutura, mas nenhum spine apareceu para descoberta no submenu Fabric Membership.

Valide a conectividade física de folha para coluna. No exemplo abaixo, o switch leaf está conectado a um spine através da 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
...
```

Se a porta estiver em um status **fora de serviço**, verifique no spine se o LLDP foi recebido da folha conectada diretamente.

```
(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
```

Outra validação é verificar se não há diferença de versão entre folha e coluna. Se houver, corrija a situação copiando a versão mais recente para /bootflash do spine. Em seguida, configure o switch para inicializar no software com os seguintes comandos:

```
(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

Se a nova imagem for removida continuamente do flash de inicialização, certifique-se de que a pasta esteja menos da metade cheia removendo imagens antigas ou arquivos auto-s; verifique a utilização de espaço usando 'df -h' no switch.

Após configurar a variável de inicialização, recarregue o switch e ele deverá inicializar para a nova versão.

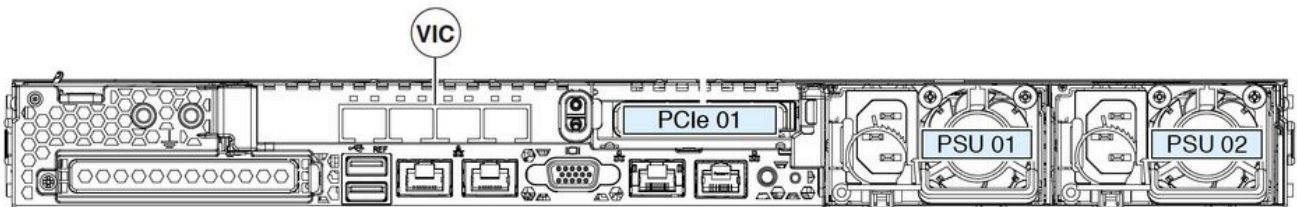
A validação de FPGA, EPLD e BIOS pode ser necessária após o recarregamento. Consulte a subseção "Leaf/Spine EPLD/FPGA not correct, F1582" para obter mais informações sobre esse assunto.

#### **Cenário 4 - Após a descoberta inicial da malha, o cluster está oscilando entre totalmente adequado e degradado**

Se isso ocorrer após uma nova configuração de estrutura, pode ser causado pelo cabeamento incorreto do APIC-M3 ou APIC-L3 que se conecta à estrutura. Você pode confirmar esse cabeamento incorreto executando "show lldp neighbors" em ambos os switches leaf conectados ao APIC. Após executar isso várias vezes, você observará que ambos os switches leaf estão visualizando a mesma interface do APIC.

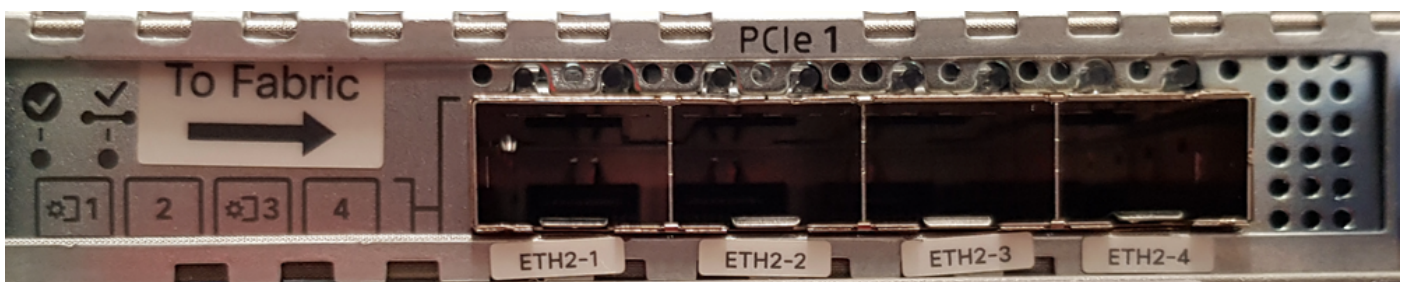
A parte traseira de um servidor APIC-M3/L3 é semelhante a:

#### **Visão traseira do servidor APIC-M3/L3**



Observe que para um APIC-M3/L3, a placa VIC tem 4 portas: ETH2-1, ETH2-2, ETH2-3 e ETH2-4, conforme mostrado abaixo:

#### **Exibição do APIC VIC 1455 com rótulos**



As regras para conectar o servidor APIC aos switches leaf são as seguintes:

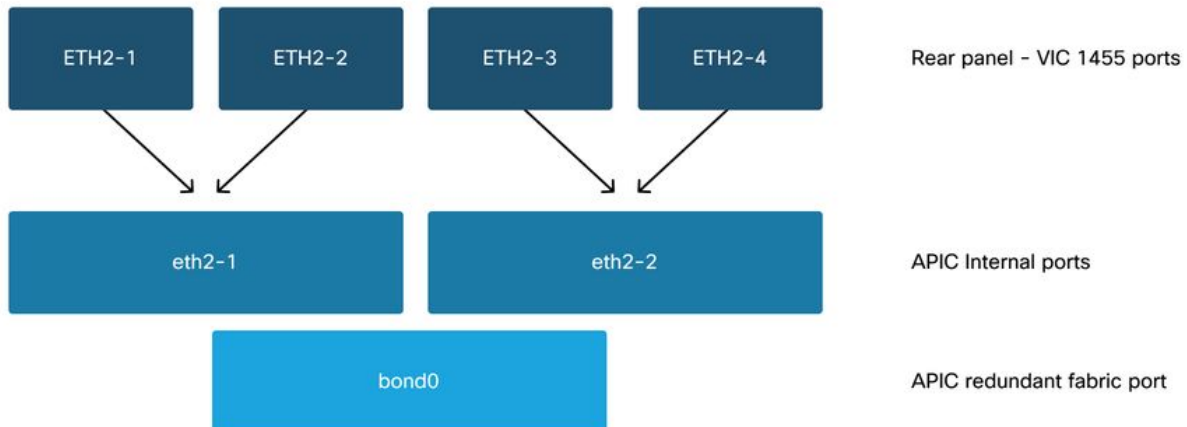
- Todas as portas devem ter a mesma velocidade, 10 Gigabit ou 25 Gigabit.
- ETH2-1 e ETH2-2 é um par de canal de porta, correspondente a eth2-1 ('ifconfig' saída) do

APIC OS.

- ETH2-3 e ETH2-4 é o outro par de canais de porta, correspondente a eth2-2 ('ifconfig' saída) no APIC OS.
- Somente uma conexão é permitida por par de canal de porta. Por exemplo, conecte um cabo a ETH2-1 ou ETH2-2 e conecte outro cabo a ETH2-3 ou ETH2-4 (**Nunca conecte ambos os ETHs em um par de canal de porta. Isso levará a problemas de detecção de estrutura.**).

Para obter mais informações, veja a seguir uma representação do mapeamento da porta VIC para a ligação APIC.

### Portas VIC 1455 — porta de estrutura redundante APIC



## Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.