

Dépannage de la découverte de fabric ACI - Configuration initiale du fabric

Contenu

[Introduction](#)

[Informations générales](#)

[Workflow de découverte de fabric](#)

[Check01 — État du système](#)

[Check02 : état DHCP](#)

[Check03 — Détails AV](#)

[Check04 — Accessibilité IP au contrôleur APIC](#)

[Check05 — VLAN infrarouge](#)

[Check06 — contiguïté LLDP](#)

[Check07 — Version du commutateur](#)

[Check08 — FPGA/EPLD/BIOS désynchronisés](#)

[Check09 — Vérification SSL](#)

[Check10 — Politique de téléchargement](#)

[Check11 — Heure](#)

[Check12 : module, bloc d'alimentation, contrôle du ventilateur](#)

[Exemple de scénarios rompus](#)

[Scénario 1 - Le premier leaf n'apparaît pas dans l'appartenance de fabric](#)

[Scénario 2 - Les autres APIC ne rejoignent pas le cluster](#)

[Scénario 3 - La colonne vertébrale n'apparaît pas dans l'appartenance au fabric](#)

[Scénario 4 : après la découverte initiale du fabric, le cluster oscille entre une configuration parfaitement adaptée et une configuration dégradée](#)

Introduction

Ce document décrit les étapes à suivre pour comprendre et dépanner le processus de détection de fabric initial, y compris des exemples de scénarios de problèmes.

Informations générales

Le matériel de ce document a été extrait de la [Dépannage de l'infrastructure axée sur les applications Cisco, deuxième édition](#), en particulier le **Découverte de fabric - Configuration initiale du fabric** chapitre.

Workflow de découverte de fabric

Le processus de découverte du fabric ACI suit une séquence d'événements spécifique. Les étapes de base sont les suivantes :

1. Connectez-vous à la **console KVM** du premier contrôleur APIC et exécutez le **script de**

configuration en entrant des valeurs telles que le nom du fabric, la taille du cluster APIC et le pool d'adresses TEP (tunnel endpoint).

2. Une fois terminé, le contrôleur APIC1 commence à envoyer le protocole **LLDP** via ses ports de fabric. Les paquets LLDP contiennent des TLV spéciaux avec des informations telles que le **VLAN infra** et son rôle en tant que APIC (également appelé contrôleur).
3. À la réception de ces paquets LLDP provenant de l'APIC1, le leaf programme le VLAN infra sur tous les ports où un APIC est détecté.
4. Le leaf commence à envoyer des découvertes DHCP sur le VLAN infra connu.
5. L'utilisateur se connecte à l'adresse **IP OOB** de APIC1 via HTTPS et enregistre le premier noeud leaf dans le sous-menu **Appartenance au fabric**.
6. Une fois que le leaf reçoit un **ID de noeud**, APIC1 répond avec une adresse IP du **pool d'adresses TEP** configuré et le processus DHCP se termine.
7. Le serveur Leaf enregistré relaie les détections DHCP à partir d'autres spines directement connectés qui ont été détectés via LLDP vers le contrôleur APIC1.
8. L'utilisateur voit ces spines détectées dynamiquement apparaître dans le sous-menu **Appartenance au fabric** et peut les enregistrer.
9. Une fois les spines enregistrés, le contrôleur APIC1 répond avec une adresse IP du pool TEP et le protocole DHCP se termine pour ces noeuds.
10. Les spines relayent les détections DHCP à partir de tous les autres noeuds de pod1. (Cela suppose qu'il existe un maillage global entre les commutateurs spines et leaf, comme cela est conseillé et c'est l'architecture type).
11. Une fois les noeuds leaf connectés aux autres APIC enregistrés, le cluster APIC peut être établi via une communication TCP entre eux. Assurez-vous de terminer la boîte de dialogue de configuration sur APIC2 et APIC3.
12. Vérifiez que tous les APIC ont formé un cluster et sont parfaitement adaptés. Si tel est le cas, la découverte du fabric est terminée.

À partir de la version 4.2, une nouvelle commande CLI est disponible sur les noeuds de fabric pour faciliter le diagnostic des problèmes de détection courants. Les sections suivantes décrivent les vérifications effectuées et fournissent des commandes de validation supplémentaires pour vous aider à résoudre les problèmes.

```
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 reachability 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 — État du système

Une fois qu'un ID de noeud a été attribué au noeud leaf et enregistré dans le fabric, il commence à télécharger son bootstrap, puis passe à l'état **en service**.

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

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

Pour valider l'état actuel du leaf, l'utilisateur peut exécuter **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 : état 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
```

Le terminal doit recevoir une adresse TEP via DHCP du contrôleur APIC1, puis établir une connectivité IP avec les autres contrôleurs APIC. Le **TEP physique** (PTEP) du leaf est attribué à loopback0. Si aucune adresse n'est attribuée, l'utilisateur peut valider que le leaf envoie une détection DHCP avec l'utilitaire `tpcdump`. Notez que nous allons utiliser l'interface `kpm_inb` qui vous permet de voir tout le trafic réseau du plan de contrôle intrabande du processeur.

```
(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
```

L'utilisateur peut également valider que `dhcpd` est exécuté sur le contrôleur APIC et qu'il écoute sur la sous-interface `bond0`. L'interface de liaison représente le fabric faisant face aux ports APIC. Nous allons utiliser le format `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
```

Check03 — Détails AV

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

Le leaf vérifiera si le contrôleur APIC enregistré a une adresse IP dans une plage valide pour le pool TEP. Si aucune information APIC n'a encore été enregistrée, cette vérification réussira. L'utilisateur peut voir les informations APIC actuelles du point de vue du noeud leaf via la commande « `acidiag avread` ». Notez dans l'exemple ci-dessous que lorsque l'invite leaf/spine affiche `(none)#`, cela indique que le leaf/spine n'est pas encore membre du 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 — Accessibilité IP au contrôleur APIC

Lorsque le leaf a reçu une adresse IP, il tente d'établir des sessions TCP avec le contrôleur APIC et commence le processus de téléchargement de sa configuration. L'utilisateur peut valider la connectivité IP au contrôleur APIC à l'aide de l'utilitaire « ping ».

```

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 — VLAN infrarouge

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

La vérification du VLAN infra ne réussit que si le noeud est connecté à un Pod où un APIC existe. Si ce n'est pas le cas, l'utilisateur peut ignorer le message car la vérification est censée échouer.

Le leaf déterminera le VLAN infra en fonction des paquets LLDP reçus d'autres noeuds ACI. Le premier reçu sera accepté lorsque le commutateur est en phase de détection.

```

(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
```

Si le VLAN infra n'a pas été programmé sur les interfaces de port de commutation connectées aux APIC, vérifiez les problèmes de câblage détectés par le 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 — contiguïté LLDP

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

Afin de déterminer quels ports se connectent à d'autres périphériques ACI, le leaf doit recevoir le protocole LLDP des autres noeuds de fabric. Pour valider la réception du protocole LLDP, l'utilisateur peut cocher « 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 — Version du commutateur

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

Si les versions APIC et leaf ne sont pas identiques, la découverte de fabric peut échouer. Pour valider la version exécutée sur le noeud leaf, utilisez « 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]
  ...
```

La même commande fonctionne également sur les cartes APIC.

```
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 désynchronisés

Les versions FPGA, EPLD et BIOS peuvent affecter la capacité du noeud leaf à activer les modules comme prévu. Si celles-ci sont trop obsolètes, les interfaces du commutateur risquent de ne pas s'activer. L'utilisateur peut valider les versions en cours et attendues de FPGA, EPLD et BIOS avec les commandes moquery suivantes.

```
(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
```

Si la version FPGA en cours d'exécution ne correspond pas à la version FPGA attendue, elle peut être mise à jour avec les étapes décrites dans le chapitre « Détection de structure », section « Remplacement de périphérique » sous le scénario « EPLD/FPGA Leaf/Spine incorrect, F1582 ».

Check09 — Vérification SSL

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

La communication SSL est utilisée entre tous les noeuds du fabric pour assurer le chiffrement du trafic du plan de contrôle. Le certificat SSL utilisé est installé lors de la fabrication et est généré en fonction du numéro de série du châssis. Le format du sujet doit être le suivant :

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

Pour valider le certificat SSL lors de la détection d'un commutateur, utilisez la commande suivante.

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

Notez que ce qui précède ne fonctionnera en tant qu'utilisateur non racine que si le noeud de commutateur est toujours en phase de découverte.

Le numéro de série du châssis est indiqué à l'aide de la commande suivante.

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

En outre, le certificat doit être valide à l'heure actuelle. Pour afficher les dates valides du certificat, utilisez l'indicateur « -dates » dans la commande 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 — Politique de téléchargement

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

Une fois que le leaf a l'accessibilité IP au contrôleur APIC, il télécharge sa configuration à partir du contrôleur APIC et le contrôleur APIC reconnaît que le téléchargement est terminé. L'état de ce processus peut être affiché avec la commande suivante.

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

```
# pcons.BootStrap
allLeaderAked      : 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 — Heure

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

Cette vérification indique l'heure actuelle à l'utilisateur. S'il y a trop de delta entre l'APIC et l'heure du commutateur, la détection peut échouer. Sur le contrôleur APIC, l'heure peut être vérifiée à l'aide de la commande `date`.

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

Check12 : module, bloc d'alimentation, contrôle du ventilateur

Pour que le commutateur soit connecté à d'autres périphériques, les modules doivent être opérationnels et en ligne. Cela peut être validé par les commandes « `show module` » et « `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	Actual	Total
-------	--------	-------

Supply	Model	Output (Watts)	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

Si un module n'est pas en ligne, réinstallez-le et vérifiez qu'il n'y a pas de discordance entre les normes FPGA, EPLD ou BIOS.

Exemple de scénarios rompus

Scénario 1 - Le premier leaf n'apparaît pas dans l'appartenance de fabric

Dans ce scénario, l'utilisateur se connecte à APIC1 après avoir terminé le script de configuration et aucun commutateur n'est apparu dans l'appartenance au fabric. Pour que la détection du

premier leaf réussisse, le contrôleur APIC doit recevoir une détection DHCP du leaf en phase de détection.

Vérifiez que le contrôleur APIC1 envoie des TLV LLDP correspondant aux paramètres définis dans le script de configuration.

```
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
```

Vérifiez également que l'APIC1 reçoit le protocole LLDP du noeud leaf directement connecté.

```
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

```

Si APIC1 reçoit le protocole LLDP du noeud leaf directement connecté, le leaf doit programmer le VLAN infra sur les ports connectés à l'APIC. Cette programmation VLAN peut être validée via la commande « show vlan encap-id <x> » où « x » est le VLAN infra configuré.

```

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

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

```

Si le VLAN infra n'a pas été programmé, vérifiez les problèmes de câblage détectés par le noeud 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
```

Lorsque l'attribut des problèmes de câblage est défini sur « infra-vlan-mismatch », l'indication est que le leaf a appris d'un autre VLAN infra que la valeur envoyée par le contrôleur APIC (la valeur envoyée par le contrôleur APIC peut être vérifiée à l'aide de la commande « moquery -c lldplnst »). Ce scénario peut se produire si le noeud terminal reçoit le protocole LLDP d'un noeud qui faisait autrefois partie d'un autre fabric. Essentiellement, un noeud en cours de détection acceptera le premier VLAN infra reçu via LLDP. Pour résoudre ce problème, supprimez les connexions entre ce noeud leaf et les autres noeuds ACI, à l'exception de l'APIC, puis nettoyez et rechargez le commutateur à l'aide des commandes « aci diag touch clean » et « reload ». Une fois que le commutateur a démarré, vérifiez que le VLAN infrarouge correct est programmé. Si cela est vrai, les connexions peuvent être restaurées vers les autres noeuds et l'utilisateur peut poursuivre la configuration du fabric ACI.

Scénario 2 - Les autres APIC ne rejoignent pas le cluster

Dans ce scénario, tous les noeuds de fabric ont été détectés, mais APIC2 et APIC3 n'ont pas encore rejoint le cluster APIC.

Validez les valeurs du script de configuration sur les cartes APIC. Les valeurs qui doivent correspondre sont :

- Domaine de fabric
- ID de fabric
- pool TEP
- VLAN infrarouge
- GIPo
- Taille du cluster
- Version du micrologiciel

```
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
```

Vérifiez les problèmes courants avec la commande « `acidiag cluster` » sur les 3 APIC.

```
apicl# 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!
```

Enfin, utilisez « `avread` » pour vérifier si ces paramètres correspondent sur tous les APIC. Notez qu'il s'agit d'une commande différente de la commande typique « `acidiag avread` » qui affiche un résultat similaire, mais elle est analysée pour une consommation plus facile.

```
apicl# 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
apicl#

```

Scénario 3 - La colonne vertébrale n'apparaît pas dans l'appartenance au fabric

Dans ce scénario, la première feuille a été découverte dans le fabric, mais aucune épine n'est apparue pour la découverte dans le sous-menu Appartenance au fabric.

Validez la connectivité physique de leaf à spine. Dans l'exemple ci-dessous, le commutateur Leaf est connecté à une colonne vertébrale via l'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
...

```

Si le port est dans un état **out-of-service**, vérifiez sur la colonne vertébrale que LLDP a été reçu du leaf connecté directement.

```

(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

```

Une autre validation consiste à vérifier qu'il n'y a pas de différence de version entre leaf et spine. Si tel est le cas, corrigez la situation en copiant la version la plus récente dans /bootflash de la colonne vertébrale. Configurez ensuite le commutateur pour démarrer le logiciel à l'aide des commandes suivantes :

```

(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
```

Si la nouvelle image est continuellement supprimée du bootflash, assurez-vous que le dossier est moins que la moitié plein en supprimant les images plus anciennes ou le fichier auto-s ; vérifiez l'utilisation de l'espace en utilisant « df -h » sur le commutateur.

Après avoir défini la variable d'amorçage, rechargez le commutateur et il devrait démarrer à la nouvelle version.

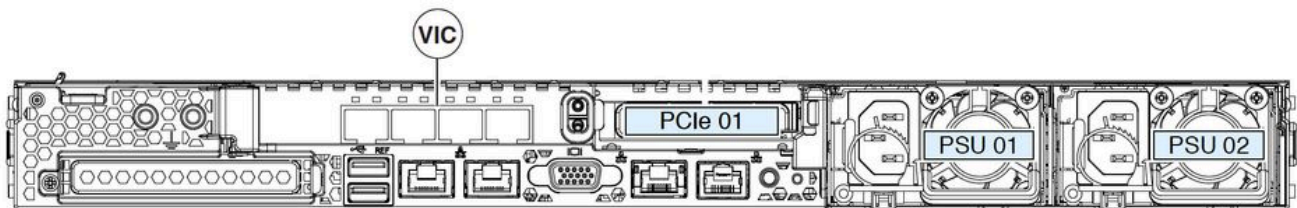
Une validation FPGA, EPLD et BIOS peut être requise après le rechargement. Reportez-vous à la sous-section « Leaf/Spine EPLD/FPGA not correct, F1582 » pour plus d'informations sur le dépannage de ce problème.

Scénario 4 : après la découverte initiale du fabric, le cluster oscille entre une configuration parfaitement adaptée et une configuration dégradée

Si cela se produit après une nouvelle configuration de fabric, cela peut être dû à un câblage incorrect du module APIC-M3 ou APIC-L3 connecté au fabric. Vous pouvez confirmer ce câblage incorrect en exécutant la commande « show lldp neighbors » sur les deux commutateurs leaf connectés au contrôleur APIC. Vous remarquerez après avoir exécuté cette commande plusieurs fois que les deux commutateurs Leaf voient la même interface APIC.

L'arrière d'un serveur APIC-M3/L3 se présente comme suit :

Vue arrière du serveur APIC-M3/L3



Notez que pour un APIC-M3/L3, la carte VIC comporte 4 ports : ETH2-1, ETH2-2, ETH2-3 et ETH2-4, comme indiqué ci-dessous :

Vue de la carte APIC VIC 1455 avec étiquettes

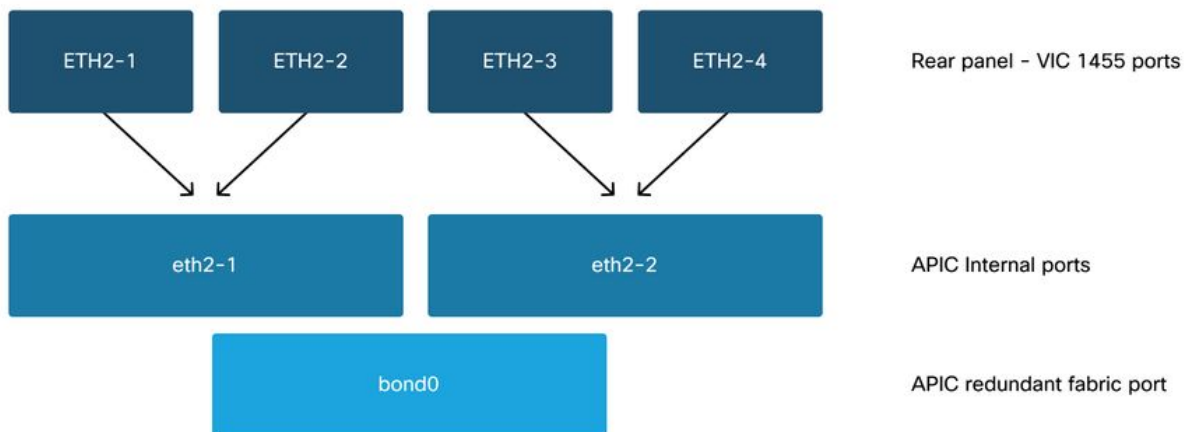


Les règles de connexion du serveur APIC aux commutateurs Leaf sont les suivantes :

- Tous les ports doivent avoir la même vitesse, 10 ou 25 gigabits.
- ETH2-1 et ETH2-2 sont une paire port-canal, correspondant à eth2-1 (sortie « ifconfig ») du système d'exploitation APIC.
- ETH2-3 et ETH2-4 est l'autre paire port-canal, correspondant à eth2-2 ('ifconfig' output) sur le système d'exploitation APIC.
- Une seule connexion est autorisée par paire port-canal. Par exemple, connectez un câble à ETH2-1 ou ETH2-2, puis connectez un autre câble à ETH2-3 ou ETH2-4 (**ne connectez jamais les deux ETH dans une paire de canaux de port. Cela entraînera des problèmes de découverte de fabric.**).

Pour une meilleure compréhension, ce qui suit est une représentation du mappage de port VIC sur la liaison APIC.

Ports VIC 1455 - port de fabric redondant APIC



À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.