思科ACI交換矩陣中的SNMP故障排除

簡介

本文檔介紹如何在Cisco ACI for ACI 5.x及更高版本中配置、驗證和排除SNMP故障。它涵蓋了SNMP策略模型、所需的管理合約、陷阱配置、使用CLI和託管對象(MO)查詢的操作驗證，以及枝葉/主幹交換機和APIC控制器之間最常見故障情形的結構化故障排除工作流程。

背景資訊

本文檔中的材料摘自ACI中的思科ACI解決方案交付團隊內部技術說明SNMP:概述、配置、故障排除和警告/問題，由Tomas de Leon編寫，並輔以Cisco APIC系統管理配置指南(5.x版)和Cisco ACI MIB快速參考指南。

概觀

ACI中的SNMP架構

SNMP（簡單網路管理協定）是一種基於UDP的協定，用於管理網路管理和監控。在ACI中，SNMP對每個受管實體獨立運行。每個枝葉交換機、主幹交換機和APIC控制器都是自己的SNMP代理 — 必須單獨輪詢或監控每個代理。

ACI支援以下SNMP功能：

- 讀取操作(Get、GetNext、BulkGet、Walk)-枝葉/主幹交換機和APIC控制器支援。
- 通知（陷阱） — 枝葉/主幹交換機和APIC控制器上支援的SNMPv1、v2c和v3陷阱。
- SNMPv3 -枝葉/主幹交換機和APIC控制器上支援。
- 寫入操作（設定） — 任何ACI裝置都不支援。
- IPv6 — 僅支援IPv4上的SNMP。

附註：在APIC集群中，每個APIC提供自身本地的MIB對象。您必須獨立輪詢每個APIC;沒有集群範圍的SNMP聚合。同樣，必須單獨查詢每個枝葉和主幹交換機。

APIC上的SNMPD架構

APIC運行snmpd進程，該進程有兩個內部元件：

- Agent — 處理SNMP協定處理和會話管理的開源net-snmp代理（版本5.7.6或更高版本）。
- DME（資料模型引擎） — 使用APIC管理資訊樹(MIT)介面讀取託管對象(MO)並將MO屬性轉換為SNMP對象格式。SNMP陷阱由MO上發生的事件和故障生成。

SNMP原則模型和部署鏈

ACI對SNMP使用策略驅動的模型。SNMP配置抽象為snmpPol託管對象，必須在將其部署到任何節點之前與交換矩陣的Pod策略組相關聯。完整的部署鏈包括：

1. SNMP策略(snmpPol) — 定義管理狀態、社群字串、客戶端組策略(ACL)和SNMPv3使用者。
2. Pod策略組 — 引用SNMP策略以及其他Pod級別策略（BGP、ISIS、NTP等）。
3. Pod Profile Selector — 將Pod策略組應用於交換矩陣Pod。

此外，SNMP陷阱配置要求：

1. SNMP監控目的地組(snmpGroup) — 定義陷阱目的地主機、埠、SNMP版本和社群。
2. 監控源(snmpSrc) — 將目標組連結到三個不同的監控策略範圍：Fabric Default、Fabric Common Policy和Access Policy Default。

APIC節點需要允許使用UDP埠161（SNMP請求）和UDP埠162（SNMP陷阱）的管理合約。枝葉節點和主幹節點還需要正確的iptables規則，配置客戶端組策略時會自動對其進行程式設計。

支援的MIB

APIC支援的MIB包括：

- 實體MIB — 物理表
- Cisco Entity Ext MIB - PhysicalProcessorTable， LEDTable
- Cisco實體FRU控制MIB — PowerSupplyGroupTable、PowerStatusTable、FanTrayStatusTable、PhysicalTable
- Cisco Entity Sensor MIB - SensorValueTable、SensorThresholdTable
- Cisco Process MIB — CPUTotalTable、ProcessTable、ProcessExtRevTable、ThreadTable

枝葉和主幹交換機公開標準NX-OS MIB，包括IF-MIB、IP-MIB、CISCO-CDP-MIB、CISCO-ENTITY-QFP-MIB和完整的CISCO-ENTITY-FRU-CONTROL-MIB套件。

APIC上生成的SNMP陷阱包括：cefcFRUInserted、cefcFRURemoted、cefcFanTrayStatusChange、cefcModuleStatusChange、entSensorThresholdNotification、cefcPowerStatusChange、

cpmCPURisingThreshold、 cpmCPUFallingThreshold。

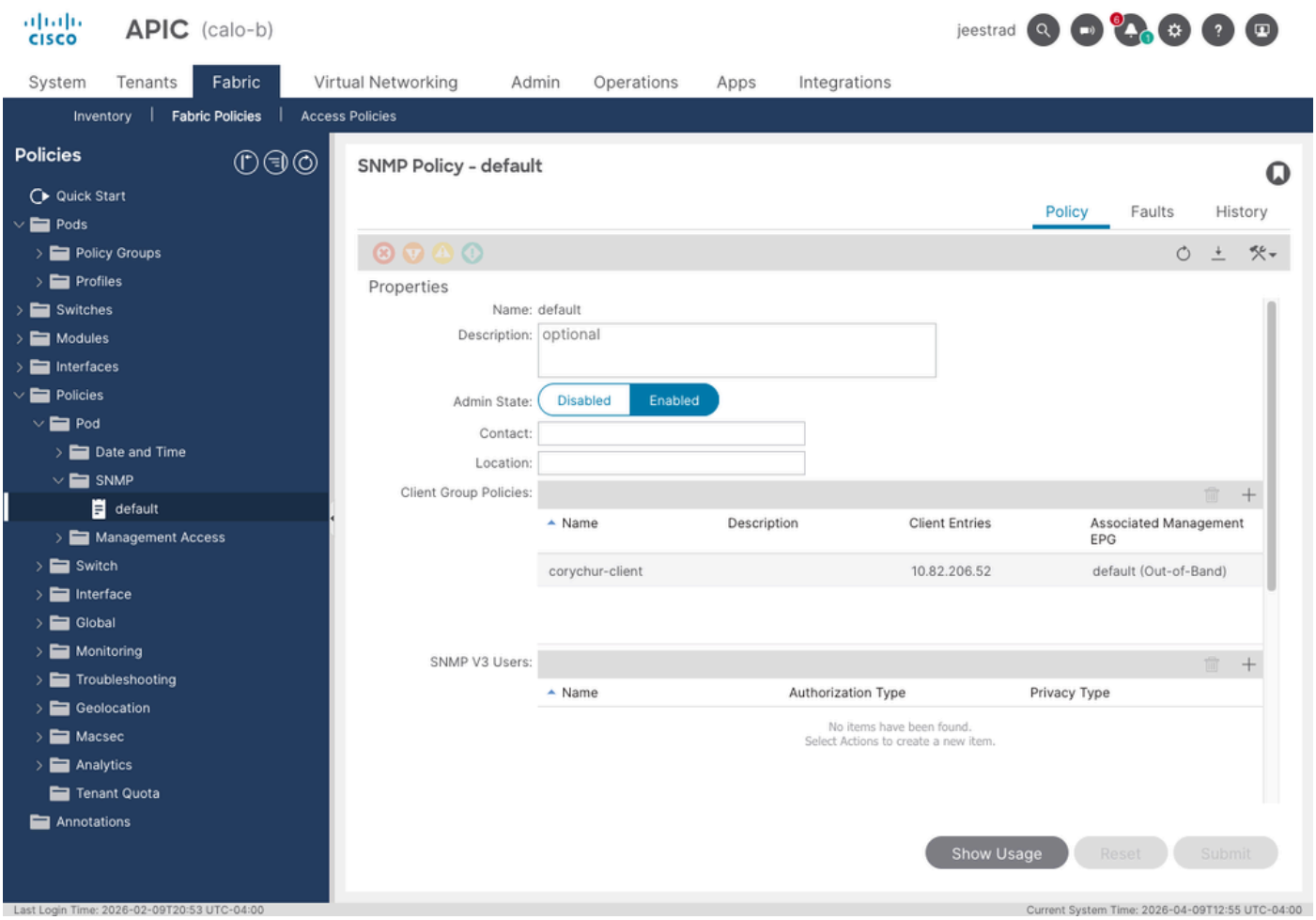
在ACI中配置SNMP

附註：本節提供配置工作流的摘要，作為後面驗證和疑難解答章節的上下文。請參閱《思科APIC系統管理配置指南》以瞭解全面的配置過程。

步驟 1:配置SNMP策略

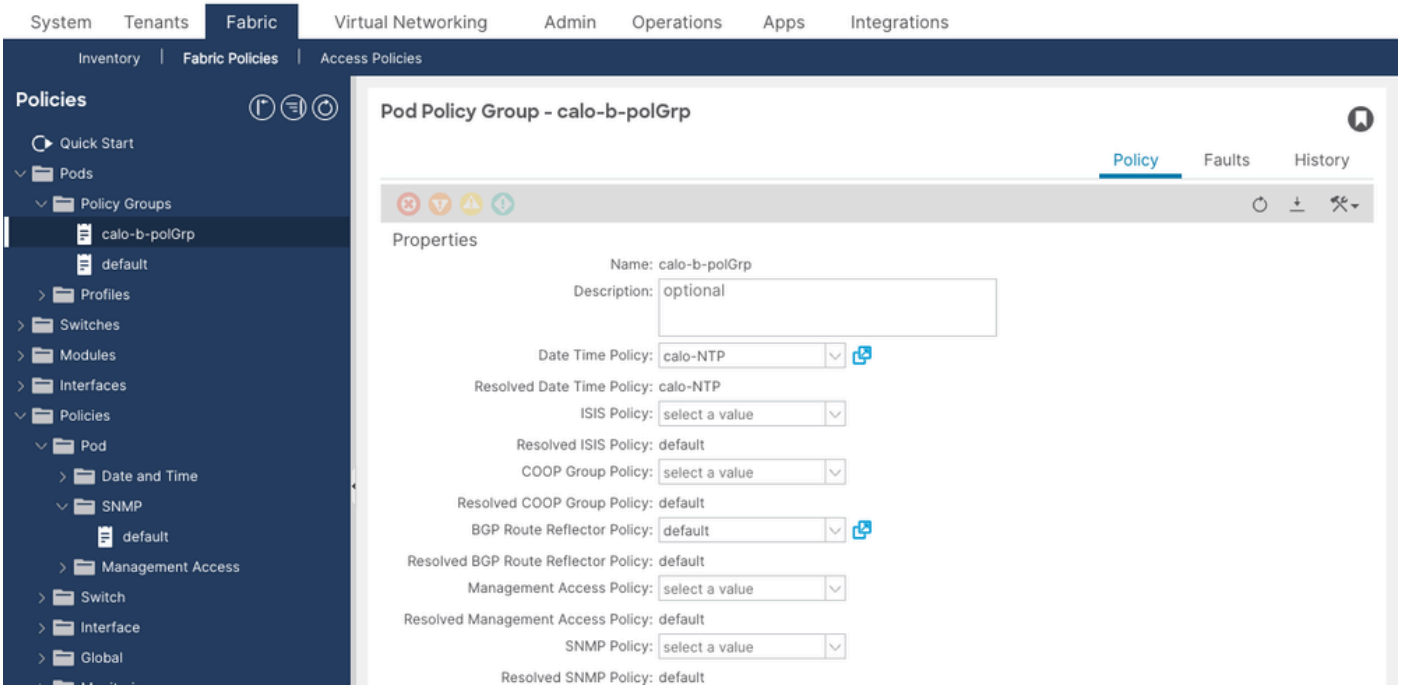
導航到Fabric > Fabric Policies > Policies > Pod > SNMP。選擇（或建立）SNMP策略，通常命名為default。設定:

- Admin State — 設定為Enabled。
- Community Policies — 新增NMS使用的社群字串。
- 客戶端組策略 — 定義一個或多個客戶端組配置檔案，每個配置檔案指定允許的SNMP客戶端IP和關聯的管理EPG（帶外或帶內）。
- SNMPv3使用者 — 如果使用SNMPv3，請在此處新增具有身份驗證和隱私引數的使用者。



步驟 2:將SNMP策略與Pod策略組關聯

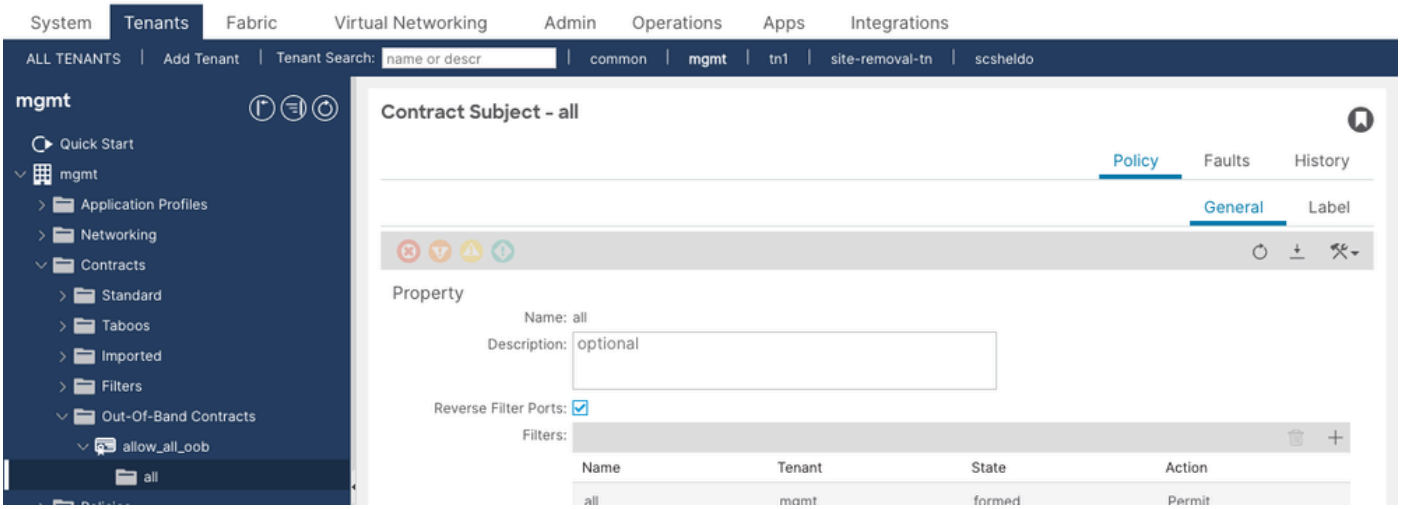
導航到Fabric > Fabric Policies > Pod > Policy Groups。選擇活動的Pod策略組(通常命名為default)。 將SNMP Policy欄位設定為指向步驟1中建立的SNMP策略。驗證Resolved SNMP Policy欄位是否顯示正確的策略名稱。



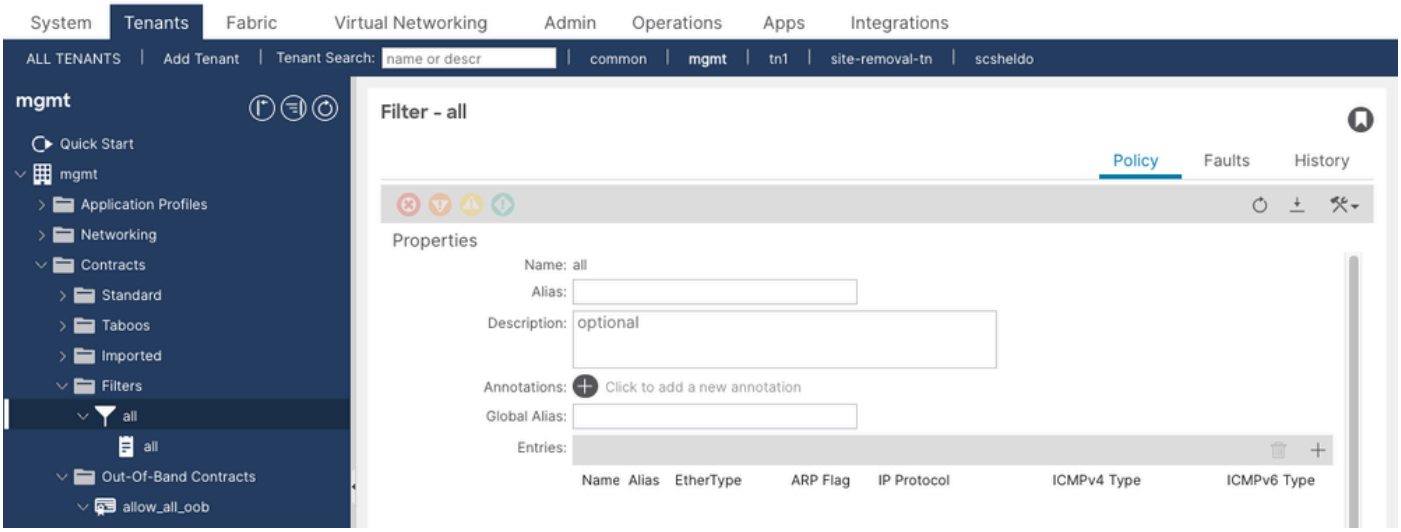
然後導航到Fabric > Fabric Policies > Pod > Profiles，展開預設Pod配置檔案，並確認活動選擇器引用正確的Pod策略組。

步驟 3:配置UDP埠161的管理合約

導覽至Tenants > Mgmt > Contracts > Out-Of-Band Contracts。驗證活動的OOB合約的Subject是否引用允許UDP埠161(SNMP請求)的過濾器條目。 如果沒有在APIC上的此合約，所有SNMP GET/WALK資料包都將以靜默方式丟棄。



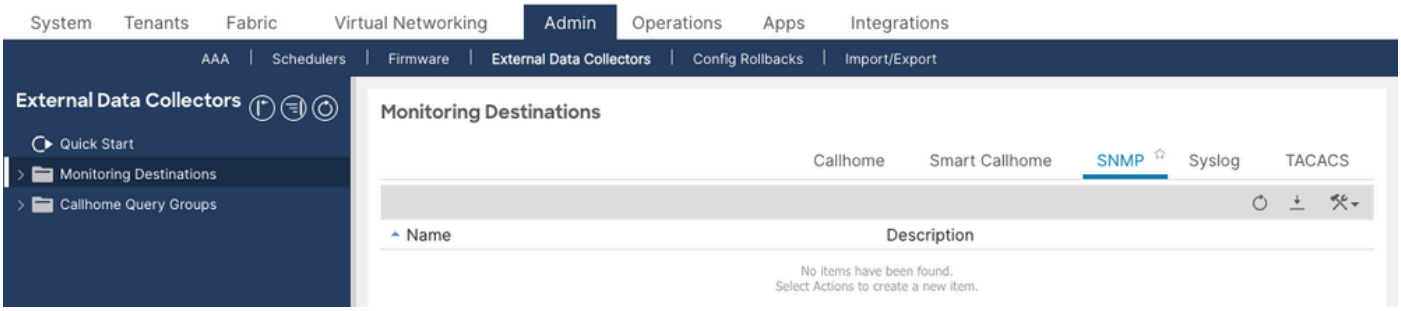
附加到合約主題的過濾器條目必須包含具有EtherType IP、協定UDP和目標埠161的條目。上面的示例顯示允許所有（未指定的協定）過濾器 — 這允許SNMP，但比建議用於生產的過濾器範圍更廣。最好使用具有特定UDP/161和UDP/162條目的專用SNMP過濾器條目。



附註：在早期的ACI韌體版本中，某些埠在枝葉和主幹節點上始終處於開啟狀態，SNMP不需要管理合約。在ACI 5.x中，APIC節點需要合約。枝葉節點和主幹節點使用源自客戶端組策略的iptables規則，而不是管理合約。

步驟 4:配置SNMP陷阱目標

導航到Admin > External Data Collectors > Monitoring Destinations > SNMP。按一下右鍵並選擇建立SNMP監控目標組。SNMP頁籤顯示所有已配置的目標組。空表表示尚未配置陷阱目標。



定義：

- 組名稱
- 陷阱目標:主機名/IP、UDP埠（預設162）、SNMP版本、社群字串和管理EPG

步驟 5:配置監控源

監控源將SNMP目標組連結到監控策略，這些策略控制哪些事件和故障生成陷阱。必須在以下所有三個位置中配置監控源，否則將不會傳送來自某些節點型別的陷阱：

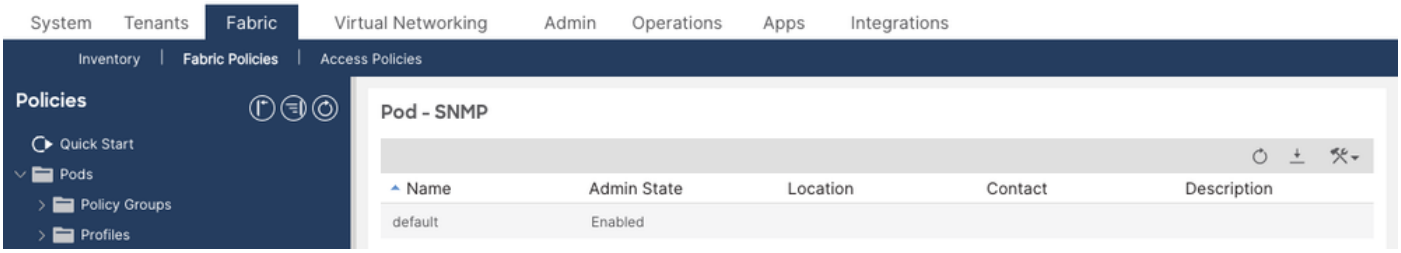
- Fabric > Fabric Policies > Policies > Monitoring > Default > Callhome/Smart Callhome/SNMP/Syslog/TACACS（涵蓋交換矩陣基礎設施事件）
- Fabric > Fabric Policies > Policies > Monitoring > Common Policy > Callhome/Smart Callhome/SNMP/Syslog/TACACS（涵蓋交換矩陣範圍的常見事件）
- Fabric > Access Policies > Policies > Monitoring > Default > Callhome/Smart Callhome/SNMP/Syslog（涵蓋接入/基礎設施事件）

在每個位置，選擇SNMP作為源型別，並建立一個引用步驟4中建立的目標組的新SNMP源。

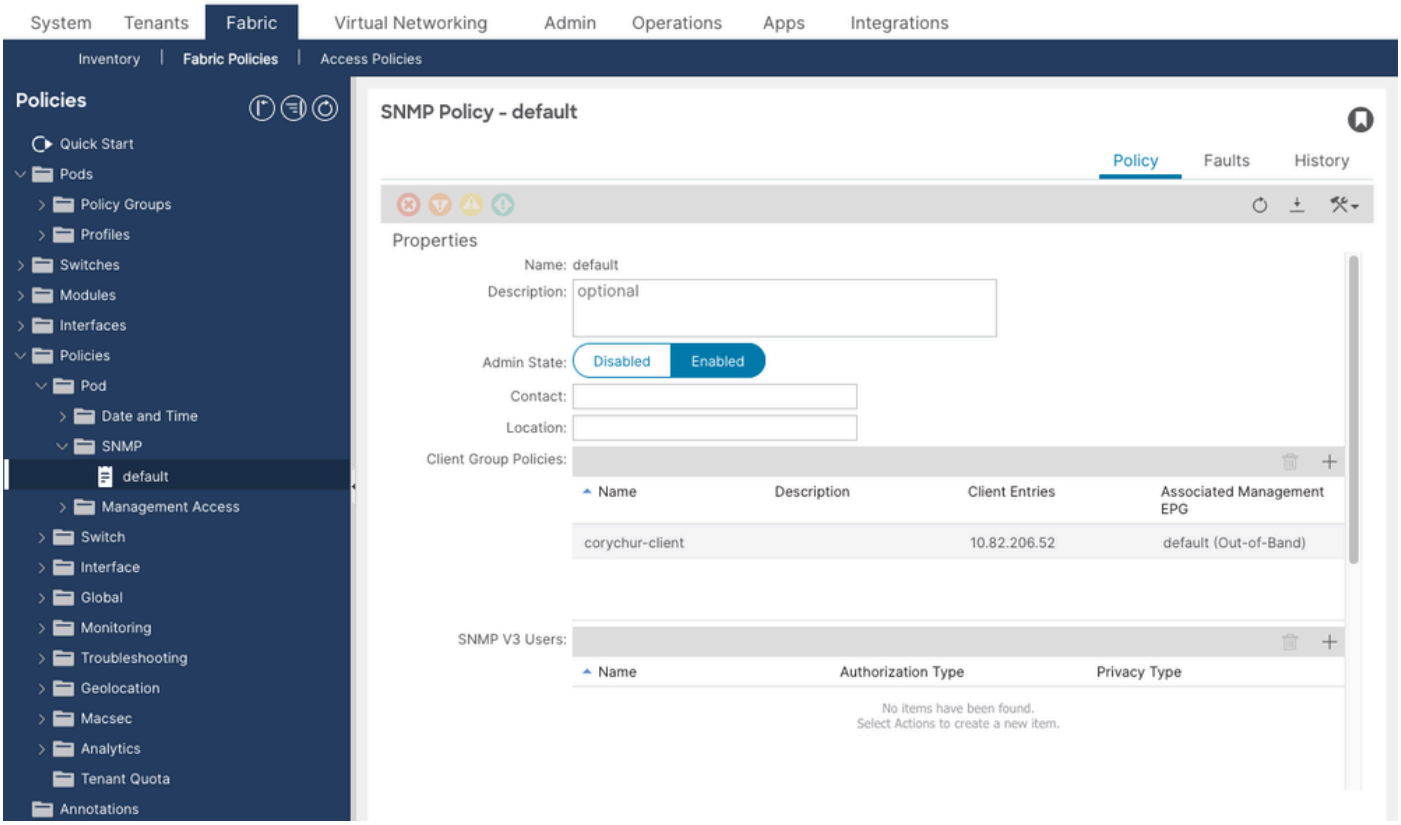
驗證設定

驗證SNMP策略部署

導覽至Fabric > Fabric Policies > Policies > Pod > SNMP，確認預設SNMP策略存在，並且其Admin State設定為Enabled。Policy Groups清單顯示所有已配置的SNMP策略及其管理狀態一覽。



有關詳細驗證，請按一下策略名稱將其開啟。確認管理狀態切換已設定為啟用，並且客戶端組策略列出所有允許的NMS主機及其關聯的管理EPG。



對任何APIC運行以下MO查詢，以確認交換矩陣中存在並啟用了SNMP策略：

<#root>

```
apic1#

moquery -c snmpPol

Total Objects shown: 1

# snmp.Pol
name          : default
adminSt       : enabled                    <--- must be "enabled"
contact       : NOC Team
descr         : ACI Fabric SNMP Policy
dn            : uni/fabric/snmppol-default
loc           : DC1 ACI Fabric
```

```
monPolDn     : uni/fabric/monfab-default
```

如果adminSt已禁用，則SNMP不會在任何節點上運行。在APIC GUI中的Fabric > Fabric Policies > Policies > Pod > SNMP > default下啟用它。

驗證社群字串組態

<#root>

```
apic1#

moquery -c snmpCommunityP

Total Objects shown: 1

# snmp.CommunityP
name         : public                      <--- confirm this matches your NMS community string
dn           : uni/fabric/snmppol-default/community-public
descr        : SNMP Community String
```

如果未返回任何社群，或者名稱與NMS使用的名稱不匹配，請在SNMP策略下新增或更正社群字串。

驗證客戶端組策略（SNMP訪問控制）

客戶端組策略用作SNMP GET/WALK訪問的ACL。每個策略指定允許哪些客戶端IP地址輪詢管理VRF所在的枝葉/主幹節點。在枝葉/主幹節點上，這些策略被轉換為iptables規則。

<#root>

```
apic1#

moquery -c snmpClientGrpP -x query-target=children

Total Objects shown: 3

# snmp.ClientP
addr         : 10.1.1.50                   <--- NMS server IP
dn           : uni/fabric/snmppol-default/clgrp-NMS-Clients/client-[10.1.1.50]
name         : nms-server1

# snmp.ClientP
addr         : 10.1.1.51
dn           : uni/fabric/snmppol-default/clgrp-NMS-Clients/client-[10.1.1.51]
name         : nms-server2
```

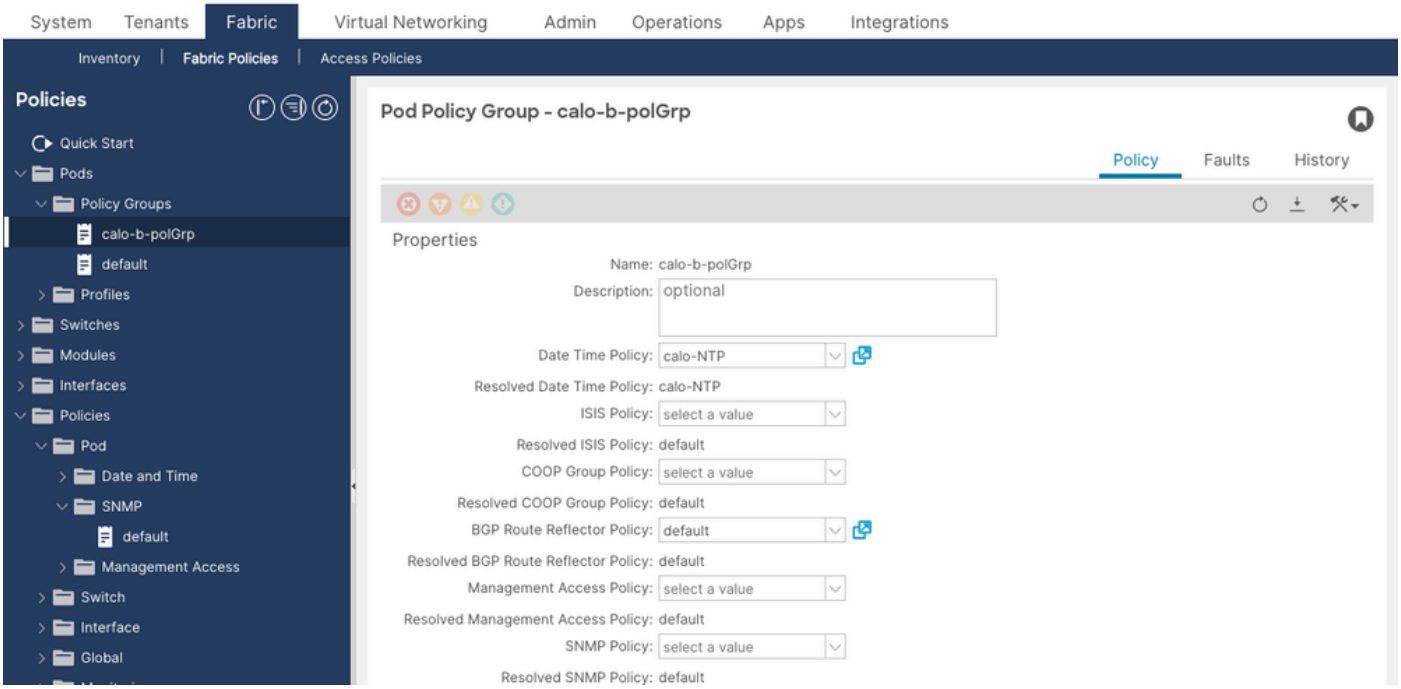
```
# snmp.ClientGrpP
name          : NMS-Clients
dn            : uni/fabric/snmppol-default/clgrp-NMS-Clients
```

確認客戶端條目中存在NMS伺服器IP。如果客戶端IP丟失，則來自該主機的SNMP GET/WALK請求將被枝葉/主幹節點上的iptables丟棄。

附註：SNMPv3警告 — 使用SNMPv3時，在APIC上不實施客戶端組策略。無論客戶端組配置如何，都允許任何SNMPv3 GET/WALK到APIC。在APIC上為SNMPv3實施客戶端組是一項已知限制。在枝葉和主幹交換機上，客戶端組實施對SNMPv2c和SNMPv3的行為相同。

驗證Pod策略組引用SNMP策略

導航到Fabric > Fabric Policies > Pod > Policy Groups，然後開啟活動的Pod Policy Group。確認SNMP Policy下拉欄位設定為所需的SNMP策略，並且Resolved SNMP Policy欄位顯示相同的名稱。策略缺失或未解析意味著永遠不會將SNMP配置推送到交換機。



在上面的螢幕截圖中，SNMP策略欄位顯示「選擇一個值」（空），而已解析的SNMP策略顯示「預設」 — 這意味著策略從交換矩陣預設值繼承，但未顯式設定。建議顯式設定SNMP策略欄位以避免歧義。

通過REST API驗證：

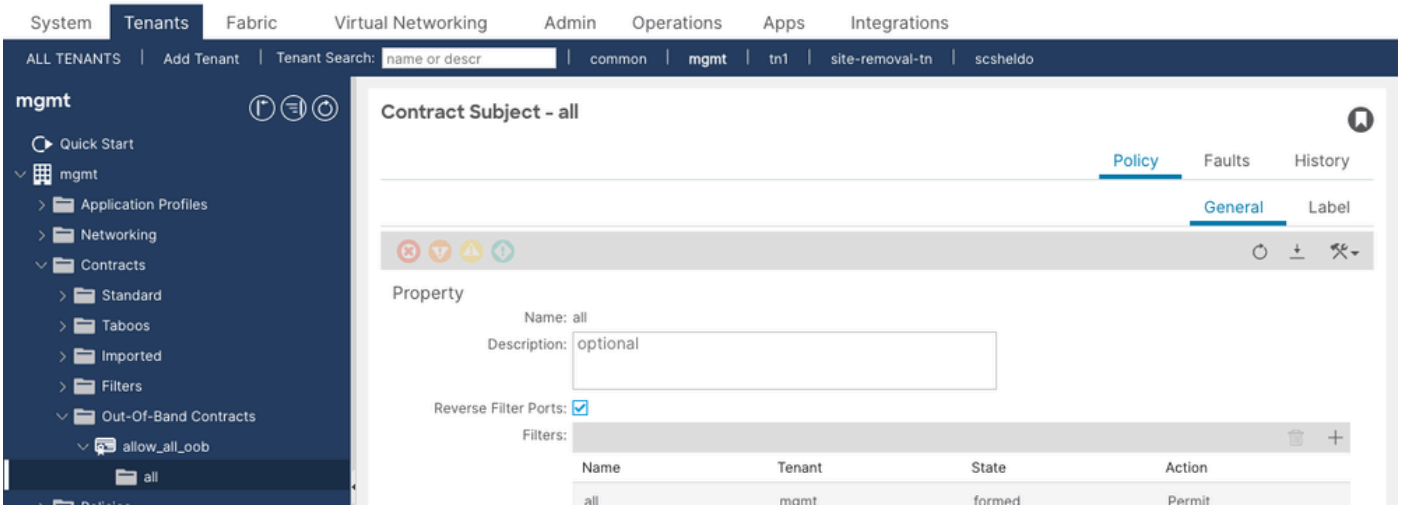
<#root>

```
apic1#

moquery -c fabricPodPGrp -x rsp-subtree=full

# fabric.PodPGrp
name          : default
dn            : uni/fabric/funcprof/podpgrp-default

  # fabric.RsSnmpPol
  tnSnmpPolName : default                  <--- must reference the SNMP policy
  state         : formed                   <--- must be "formed"
```

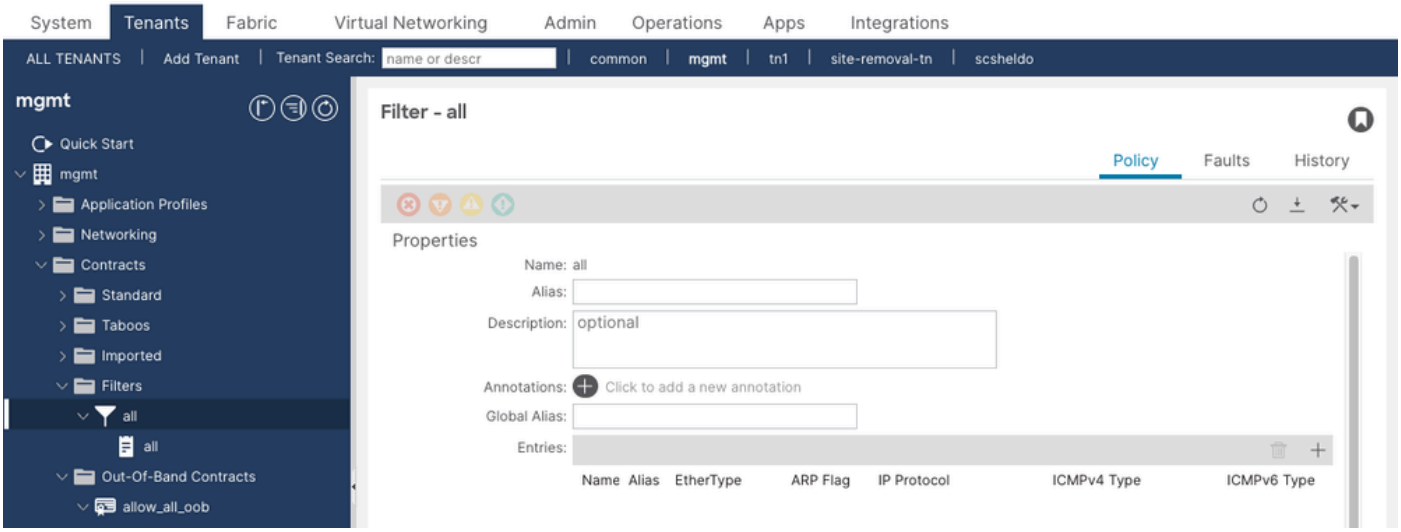
如果未形成state，則SNMP策略關係已斷開。重新選擇Pod策略組中的SNMP策略並提交。

驗證UDP 161管理合約（APIC節點）

導航到Tenants > Mgmt > Contracts > Out-of-Band Contracts（如果使用INB管理，則為In-Band Contracts）。 開啟活動的OOB合約，然後按一下Policy頁籤。驗證主題是否引用允許UDP埠161的過濾器。



展開主題引用的篩選器，並確認其條目中包含EtherType IP、協定UDP、目標埠161條目。篩選器條目確定允許哪些流量通過OOB管理合約到達APIC。



過濾器應顯示：

- EtherType:IP
- IP協定：UDP
- 目的地連線埠自：161
- 目的地連線埠收件人：161

此外，如果希望APIC通過OOB介面出站SNMP陷阱，請驗證是否允許UDP埠162。

通過MO查詢檢查：

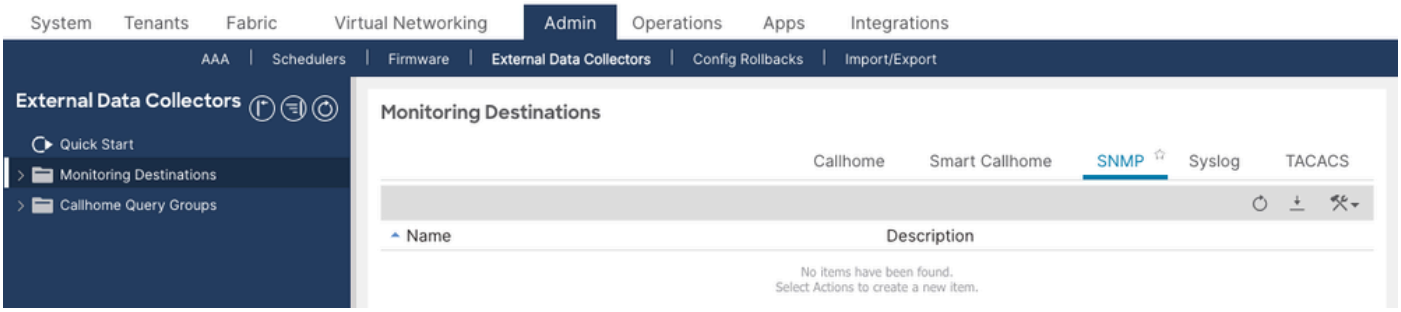
<#root>

```
apic1#

moquery -c vzEntry -x query-target-filter='and(eq(vzEntry.dFromPort,"161"),eq(vzEntry.prot,"17"))'

Total Objects shown: 2

# vz.Entry
name         : snmp-get
dn           : uni/tn-mgmt/flt-snmp-filter/e-snmp-get
dFromPort    : 161                        <--- destination port 161
dToPort      : 161
prot         : 17                         <--- UDP
stateful     : no
```

如果未返回任何結果，則不存在UDP 161的過濾器。向管理合約中新增一個。

驗證SNMP陷阱目標配置

導航到Admin > External Data Collectors > Monitoring Destinations > SNMP，檢視所有已配置的SNMP目標組。空清單表示未配置陷阱目標，並且不會從任何節點傳送陷阱。



<#root>

apic1#

moquery -c snmpTrapDest

Total Objects shown: 1

```
# snmp.TrapDest
host         : 10.1.1.50                  <--- NMS trap receiver IP
port         : 162                         <--- trap UDP port
ver          : v2c                         <--- SNMP version
secName      : public                      <--- community string (v2c) or username (v3)
v3SecLvl     : noauth
notifT       : traps
vrfName      : mgmt:inb                    <--- VRF used to reach the trap receiver
epgDn        : uni/tn-mgmt/mgmtp-default/inb-default
dn           : uni/fabric/snmpgroup-NMS-DestGrp/trapdest-10.1.1.50-port-162
```

確認陷阱目標IP、埠、版本、社群字串和管理VRF(mgmt:inb或management for OOB)與您的環境匹配。VRF必須與分配給目標的管理EPG匹配。

驗證是否在所有三個作用域中配置了監控源

SNMP源必須存在於所有三個監視策略作用域中。任何範圍中缺少源意味著將不會轉發相關事件的陷阱。

<#root>

apic1#

moquery -c snmpSrc | egrep "snmp.Src|name|dn|incl|minSev|monPolDn"

```
# snmp.Src
name         : NMS-snmpSrc
```

```
dn           : uni/fabric/monfab-default/snmpsrc-NMS-snmpSrc        <--- Fabric Default
incl         : audits,events,faults
minSev       : info
monPolDn     : uni/fabric/monfab-default

# snmp.Src
name         : NMS-snmpSrc
dn           : uni/fabric/moncommon/snmpsrc-NMS-snmpSrc              <--- Fabric Common
incl         : audits,events,faults
minSev       : info
monPolDn     : uni/fabric/moncommon

# snmp.Src
name         : NMS-snmpSrc
dn           : uni/infra/moninfra-default/snmpsrc-NMS-snmpSrc        <--- Access Default
incl         : audits,events,faults
minSev       : info
monPolDn     : uni/infra/moninfra-default
```

如果缺少這三個來源中的任何一個，則使用GUI在相應的監控策略中建立缺失的SNMP源。

操作驗證

使用show snmp summary(APIC)驗證SNMP狀態

直接在每個APIC上運行此命令，以確認SNMP代理正在運行且已應用配置：

<#root>

```
apic1#

show snmp summary


Active Policy:
default, Admin State: enabled             <--- admin state must be "enabled"

Local SNMP engineID: [Hex] 0x8000000980e2b692088976c75600000000

----------------------------------------
Community           Description
----------------------------------------
public              SNMP Community String  <--- community must be present

------------------------------------------------------------
User                Authentication   Privacy
------------------------------------------------------------
                                          <--- empty if using v2c only

------------------------------------------------------------
Client-Group        Mgmt-Epg              Clients
------------------------------------------------------------
NMS-Clients         default (In-Band)     10.1.1.50,10.1.1.51  <--- verify client IPs
```

```
-------------------------------------------------------------
Host                Port    Version   Level       SecName
-------------------------------------------------------------
10.1.1.50           162     v2c       noauth      public        <--- trap destination
```

輸出中要驗證的內容：

- 必須啟用管理狀態。
- 社群必須與NMS的配置匹配。
- 客戶端組必須列出所有允許的NMS IP以及正確的管理EPG。
- 主機（陷阱目的地）必須以正確的埠和版本列出NMS陷阱接收器。

使用show snmp summary(Leaf/Spine)驗證SNMP狀態

<#root>

```
leaf101#

show snmp summary

Admin State : enabled, running (pid:8192)  <--- must show "enabled, running" with a PID

Local SNMP engineID: [Hex] 80000009037C69F6105BF9

---------------------------------------------------------------------
Community       Context         Status
---------------------------------------------------------------------
public                          ok                                  <--- community status must be "o

---------------------------------------------------------------------
Client          VRF             Status
---------------------------------------------------------------------
10.1.1.50       mgmt:inb        ok                                  <--- client entry must be "ok"
10.1.1.51       mgmt:inb        ok

-------------------------------------------------------------------------------
Host            Port    Ver     Level   SecName         VRF
-------------------------------------------------------------------------------
10.1.1.50       162     v2c     noauth  public          mgmt:inb       <--- trap destination
```

輸出中要驗證的內容：

- 必須啟用管理狀態，並使用pid運行。如果顯示disabled，則表示未應用SNMP原則或Pod原則鏈已中斷。
- 社群狀態必須正常。error狀態表示策略部署問題。
- 每個NMS主機的客戶端VRF必須與管理EPG的VRF相匹配(mgmt:inb用於帶內，管理用於OOB)。

- 陷阱主機必須列出具有正確VRF上下文的目標。

驗證snmpd進程是否正在運行

在枝葉或主幹上：

```
<#root>
leaf101#
ps aux | grep snmp
root      5881  2.5 1907404  411444 ?    Ssl  Apr05   /isan/bin/snmpd -f -s -d udp:161 udp6:161 tcp:161
leaf101#
pidof snmpd
5881
```

在APIC上：

```
<#root>
apic1#
ps aux | grep snmp
ifc   32182  1.4   0.1  641196  239716  ?   Ssl  Apr10   /mgmt//bin/snmpd.bin \
  -f -p /tmp/snmpd2.pid -a -A -LE 0-2 -c /data//snmp/snmpd.conf
```

如果在枝葉或主幹上找不到snmpd進程，則表明該節點上未運行SNMP。檢查SNMP策略管理狀態是否已啟用，Pod策略鏈是否正確配置。

[擾流器](#)（突出顯示讀取）

驗證SNMP埠是否正在偵聽

```
<#root>
leaf101#
netstat -lutn | grep 161
```

```
Active Internet connections (only servers)
Proto  Recv-Q  Send-Q  Local Address    Foreign Address  State
tcp         0       0  0.0.0.0:161      0.0.0.0:*        LISTEN   <--- SNMP agent is accepting requests
udp         0       0  0.0.0.0:161      0.0.0.0:*
udp6        0       0  :::161           :::*
```

如果埠161未列在LISTEN狀態，則snmpd進程未運行或無法繫結到該埠。

驗證枝葉/主幹上的iptables規則

客戶端組策略被轉換為每個枝葉和主幹上的iptables規則。使用以下內容檢查規則：

<#root>

leaf101#

iptables -S | grep -i snmp

```
-N snmp_rules
-N vrf_2_snmp_rules
-N vrf_9_snmp_rules
-A INPUT -p udp -m udp --dport 161 -j snmp_rules    <--- SNMP port 161 redirects to snmp_rules chain
-A snmp_rules -m vrf --vrf 2 -j vrf_2_snmp_rules    <--- VRF 2 = OOB management
-A snmp_rules -m vrf --vrf 9 -j vrf_9_snmp_rules    <--- VRF 9 = In-Band management
-A snmp_rules -j DROP                                <--- default drop; only permitted clients pass
-A vrf_2_snmp_rules -s 10.1.1.50/32 -j ACCEPT       <--- permitted NMS client (OOB VRF)
-A vrf_9_snmp_rules -s 10.1.1.50/32 -j ACCEPT       <--- permitted NMS client (INB VRF)
```

要確定交換矩陣的正確VRF ID，請運行：

<#root>

leaf101#

show vrf

```
VRF-Name                VRF-ID  State   Reason
management              2       Up      --
mgmt:inb                9       Up      --
```

iptables規則中的VRF ID必須與`show vrf`報告匹配。如果iptables規則中沒有客戶端IP，則該主機的SNMP請求將被靜默丟棄，即使snmpd進程正在運行。

使用計數器檢查是否已匹配或丟棄任何SNMP資料包：

<#root>

leaf101#

iptables -nvL | grep -A 20 "Chain snmp_rules"

```
Chain snmp_rules (1 references)
 pkts bytes target         prot opt in     out    source       destination
    1    73 vrf_9_snmp_rules all -- *      *      0.0.0.0/0    0.0.0.0/0    vrf 9
    0     0 DROP             all -- *      *      0.0.0.0/0    0.0.0.0/0    <--- if pkts>0 here, client
```

附註：如果SNMP正在運行，但iptables未顯示snmp_rules鏈，或者鏈為空，則可以重新啟動snmpd進程以強制iptables規則重新程式設計。將SIGKILL傳送到snmpd PID是安全的 — ACI進程管理器(監管)將自動重新啟動它。運行pidf snmpd以獲取PID，然後kill -9 [snmpd_pid]。在10-15秒後使用pidf snmpd確認新的PID。

驗證SNMP埠是否正在偵聽枝葉101# netstat -lutn | grep 161活動網際網路連線（僅伺服器）Proto Recv-Q Send-Q本地地址外部地址狀態tcp 0 0.0.0.0:161 0.0.0.0:* LISTEN <— SNMP代理正在接受請求udp 0 0.0.0.0:161 0.0.0:* udp6 0 0 :::161 ::*如果埠161未列在LISTEN狀態，則snmpd進程未運行或未能繫結到該埠。　　驗證枝葉/主幹客戶端組策略上的iptables規則是否轉換為每個枝葉和主幹上的iptables規則。使用以下內容檢查規則：leaf101# iptables -S | grep -i snmp -N snmp_rules -N vrf_2_snmp_rules -N vrf_9_snmp_rules -A INPUT -p udp -m udp —dport 161 -j snmp_rules <— SNMP埠161重定向到snmp_rules鏈 — A snmp_rules -m vrf 2 -j vrf_2_snmp_rules <— VRF 2 = OOB管理 — A snmp_rules -m vrf —vrf 9 -j vrf_9_snmp_rules <— VRF 9 =帶內管理 — A snmp_j DROP < — 預設丟棄；僅允許客戶端通過 — A vrf_2_snmp_rules -s 10.1.1.50/32 -j ACCEPT < — 允許的NMS客戶端(OOB VRF)-A vrf_9_snmp_rules -s 10.1.1.50/32 -j ACCEPT < — 允許的NMS客戶端(INB VRF)要標識交換矩陣的正確VRF ID，請運行：leaf101# show vrf VRF-Name VRF-ID State Reason management 2 Up — mgmt:inb 9 Up — iptables規則中的VRF ID必須與哪個show vrf報告匹配。如果iptables規則中沒有客戶端IP，則該主機的SNMP請求將被靜默丟棄，即使snmpd進程正在運行。使用計數器檢查是否已匹配或丟棄任何SNMP資料包：leaf101# iptables -nvL | grep -A 20 "Chain snmp_rules" Chain snmp_rules（1引用）pkts bytes target prot opt in out source destination 1 73 vrf_9_snmp_rules all — * 0.0.0.0/0 0.0.0.0/0 vrf 9 0 DROP all — * 0.0.0.0/0 0.0.0.0/0 < — 如果此處的pkts>0，則客戶端IP丟失註：如果SNMP正在運行，但iptables未顯示snmp_rules鏈，或者鏈為空，則可以重新啟動snmpd進程以強制iptables規則重新程式設計。將SIGKILL傳送到snmpd PID是安全的 — ACI進程管理器（受管制）將自動重新啟動它。運行pidf snmpd以獲取PID，然後停用–9 [snmpd_pid]。在10-15秒後使用pidf snmpd確認新的PID。

驗證與SNMP埠的網路連線

<#root>

```
leaf101#

netstat -ai | grep eth0

Iface  MTU  Met  RX-OK  RX-ERR  RX-DRP  RX-OVR  TX-OK  TX-ERR  TX-DRP  TX-OVR  Flg
eth0   1500   0  501277      0       0       0  633546      0       0       0  BMRU

leaf101#

netstat -ai | grep kpm_inb

Iface     MTU  Met    RX-OK  RX-ERR  RX-DRP  RX-OVR    TX-OK  TX-ERR  TX-DRP  TX-OVR  Flg
```

```
kpm_inb  9300   0  10361421     0       0       0   8958506      0     126       0  BMRU
```

確認管理介面處於活動狀態（無RX-ERR增量）並傳輸流量。eth0是OOB管理介面；kpm_inb是交換器上的頻內管理介面。

驗證使用tcpdump傳送SNMP陷阱

要確認正在從枝葉或主幹節點生成和傳送陷阱，請捕獲相應介面上的流量。以admin身份訪問該節點並使用：

<#root>

```
leaf101#

tcpdump -i kpm_inb -f port 162 -vv

tcpdump: listening on kpm_inb, link-type EN10MB (Ethernet), capture size 65535 bytes
17:21:49.810052 IP (tos 0x0, ttl 64, id 63116, proto UDP, length 218)
    172.18.242.14.35582 > 10.1.1.50.snmp-trap: { SNMPv2c C=public
    { V2Trap(171) R=253 system.sysUpTime.0=5888267
      S:1.1.4.1.0=E:cisco.9.276.0.1
      interfaces.ifTable.ifEntry.ifIndex.436224000=436224000
      interfaces.ifTable.ifEntry.ifOperStatus.436224000=2 }}        <--- verify trap is being sent to N
```

對於OOB:

<#root>

```
leaf101#

tcpdump -i eth0 -f port 162 -vv
```

[擾流器](#)（突出顯示讀取）

對於APIC陷阱(INB):

<#root>

```
apic1#

tcpdump -i bond0.1100 -f port 162

20:01:08.453473 IP apic1-inb.cisco.com.59417 > 10.1.1.50.snmptrap: C=public V2Trap(85) S:
1.1.4.1.0=E:cisco.9.117.2.0.2 E:cisco.9.117.1.1.2.1.1.10548=1 E:cisco.9.117.1.1.2.1.2.10548=2
```

附註：在APIC上，bond0.1100是帶內管理介面VLAN子介面。使用為帶內管理EPG配置的VLAN封裝替換1100。使用oobmgmt作為APIC上OOB捕獲的介面名稱。

對於APIC陷阱(INB):apic1# tcpdump -i bond0.1100 -f埠162 20:01:08.453473 IP apic1-inb.cisco.com.59417 > 10.1.1.50.snmptrap:C=公共V2陷阱(85)S:1.1.4.1.0=E:cisco.9.117.2.0.2 E:cisco.9.117.1.1.2.1.10548=1 E:cisco.9.117.1.1.2.1.2.10548=2注：在APIC上，bond0.1100是帶內管理介面VLAN子介面。使用為您的帶內管理EPG配置的VLAN封裝替換1100。使用oobmgmt作為APIC上OOB捕獲的介面名稱。

使用tcpdump驗證SNMP GET/WALK請求

<#root>

```
leaf101#

tcpdump -i kpm_inb -f port 161 -vv

17:26:08.548149 IP 10.1.1.50.64245 > leaf101.cisco.com.snmp: { SNMPv2c C=public
  { GetRequest(28) R=949769396 system.sysDescr.0 }}              <--- GET request received
17:26:08.552290 IP leaf101.cisco.com.snmp > 10.1.1.50.64245: { SNMPv2c C=public
  { GetResponse(191) R=949769396
    system.sysDescr.0="Cisco NX-OS(tm) aci, Software (aci-n9000-system), \
Version 15.0(1k), RELEASE SOFTWARE" }}                          <--- response returned; SNMP working
```

如果您看到了GetRequest但沒有GetResponse，則表示已收到該請求，但未得到應答。檢查snmpd進程和團體字串。如果您既沒有看到請求也沒有看到響應，則在到達節點之前請求會被阻止（請檢查路由和iptables）。

工作流故障排除

分類決策樹

當工程師報告SNMP無法工作時，請使用此診斷樹。從觀察到的症狀開始，按照分支進行隔離。

症狀：沒有對SNMP GET/WALK請求的響應

1. 檢查APIC上的SNMP管理狀態。運行`moquery -c snmpPol`。如果`adminSt`已停用，請啟用它並繼續步驟7。
2. 檢查snmpd進程。在受影響的節點上，運行`ps aux | grep snmp`或`pidof snmpd`。如果沒有進程正在運行，則不會部署SNMP策略。驗證Pod策略鏈(SNMP策略→Pod策略組和→配置檔案)。
3. 檢查埠161是否正在監聽。運行`netstat -lutn | grep 161`。如果埠161未處於LISTEN狀態，則

snmpd進程已失敗；從/var/log/dme/log/svc_ifc_dbgrelem.log*收集日誌並重新啟動進程。
4. 檢查路由。運行show ip route vrf management和show ip route vrf mgmt:inb。確認通往NMS主機的路由存在於正確的VRF中。
5. 檢查APIC上的管理合約。如果目標是APIC（不是枝葉/主幹），請驗證OOB或INB管理合約中是否允許UDP 161。
6. 在節點上執行tcpdump。運行tcpdump -i kpm_inb -f埠161 -vv(或eth0用於OOB)。 如果出現GetRequest但之後沒有GetResponse，則該請求將到達節點，但snmpd沒有響應 — 請檢查社群字串。如果沒有顯示任何請求，則問題是上游（路由或合約）。
7. 從允許的客戶端進行測試。從客戶機組中所列出的NMS主機運行snmpget -v2c -c [community] [node-ip] SNMPv2-MIB::sysDescr.0。成功響應確認SNMP完全可操作。

症狀：NMS未收到SNMP陷阱

1. 檢查陷阱目標配置。運行moquery -c snmpTrapDest。確認NMS IP、埠、版本和社群與NMS預期值匹配。
2. 檢查所有三個作用域中是否存在監控源。運行moquery -c snmpSrc | egrep "snmp.Src|name|dn"。 使用uni/fabric/monfab-default、uni/fabric/moncommon和uni/infra/moninfra-default的monPolDn值確認條目存在。如果缺少任何SNMP，請在相應的監控策略中新增SNMP源。
3. 檢查snmpd進程。驗證snmpd是否正在應該傳送陷阱的節點上運行。
4. 生成測試事件並使用tcpdump捕獲。擺動介面或更改狀態以生成事件。在節點上，運行tcpdump -i kpm_inb -f port 162 -vv。如果線路上未出現陷阱流量，則事件不會生成陷阱 — 重新檢查監控源include屬性(必須包括故障或事件)。
5. 檢查與陷阱接收器的連通性。確認陷阱接收器可從管理VRF到達：show ip route vrf mgmt:inb應顯示通往NMS主機的路徑。
6. 如果陷阱顯示在tcpdump上但不顯示在NMS上，則問題在於網路側：防火牆、路由或NMS配置。檢查NMS是否正在從ACI節點的管理源IP監聽UDP 162。

常見方案

案例 1:已啟用SNMP策略，但沒有從枝葉/主幹返回任何資料

問題:APIC上的SNMP策略顯示Admin State enabled。NMS可以到達枝葉的管理IP。無響應的snmpget超時。

配置檢查：驗證Pod策略組引用SNMP策略，並且「已解析的SNMP策略」顯示正確的名稱。如果Pod策略組的SNMP策略欄位為空或未形成關係，則snmpd進程可能無法在交換機上啟動。

操作檢查：使用SSH連線到受影響的枝葉並運行show snmp summary。如果輸出顯示Admin State:已禁用，即使APIC顯示已啟用，但尚未部署該策略。檢查Pod策略鏈中是否存在缺失或引用不當的Pod策略組。

根本原因：SNMP策略未連結到Pod策略組，或者Pod配置檔案選擇器未將正確的Pod策略組應用於此Pod。

解決方案：

1. 導航到Fabric > Fabric Policies > Pod > Policy Groups > default。
2. 確認SNMP Policy欄位指向已啟用的SNMP策略。
3. 導航到Fabric > Fabric Policies > Pods > Profiles，並確認活動選擇器引用此Pod策略組。
4. 儲存後，在2分鐘內重新檢查枝葉上的`show snmp summary`。

案例 2:SNMP GET/WALK適用於某些NMS主機，但不適用於其他NMS主機

問題:一個NMS伺服器可以成功輪詢ACI節點。另一個子網上的第二台NMS伺服器沒有響應。

配置檢查：在APIC上運行`moquery -c snmpClientGrpP -x query-target=children`。確認第二台NMS伺服器的IP被列為客戶端條目。如果缺少該IP，則該IP將被snmp_rules鏈底部的iptables DROP規則阻止。

運行檢查：在受影響的枝葉上，確認OOB或INB管理合約中允許使用UDP 161。如果沒有合約或過濾器具有SNMP埠，則請求將被丟棄。

根本原因：第二個NMS伺服器IP不在客戶端組策略中。

解決方案：在Fabric > Fabric Policies > Policies > Pod > SNMP > Default > Client Group Policies下的SNMP Client Group Policy中，將缺失的NMS IP作為客戶端條目新增。所有節點上的iptables規則將在儲存策略後的幾分鐘內更新。

案例 3:未收到SNMP陷阱 — 生成但未傳送陷阱

問題:故障在APIC故障表中可見。`moquery -c snmpTrapDest`顯示正確的NMS IP。NMS未收到陷阱。

配置檢查：運行`moquery -c snmpSrc | egrep "snmp.Src|name|dn"`。 驗證所有三個作用域中是否存在監控源(`monfab-default`、`moncommon`、`moninfra-default`)。一個常見的疏忽是隻在Fabric Default（交換矩陣預設）策略中配置源，這遺漏了訪問策略事件。

操作檢查：觸發測試事件（例如，將介面切換為管理關閉狀態）。 在相關節點上，運行`tcpdump -i kpm_inb -f port 162`。如果陷阱資料包出現在節點的介面上，則ACI端工作正常，問題出在通往NMS（防火牆、路由）的網路路徑上。 如果線路上未出現陷阱，則ACI監控源丟失或事件型別未包括在源的`incl`屬性中。

根本原因1:所需範圍中缺少一個或多個監控源。

根本原因2:監視源包含屬性排除正在生成的事件型別(例如，包括：沒有faults的事件表示不會傳送基於故障的陷阱)。

解決方案：

1. 在GUI中為三個作用域（交換矩陣預設值、交換矩陣通用和訪問預設值）中的每一個作用域新增缺少的監控源。 將目標組設定為已配置的SNMP目標組。
2. 驗證incl屬性包括審計、事件、故障，以獲得全面的陷阱覆蓋範圍。
3. 更改後，重新觸發測試事件並重新檢查tcpdump。

擾流器（突出顯示讀取）

附註：在APIC上，tcpdump/code>命令僅供root使用者使用。對於APIC和交換機iptables命令僅對root使用者可用。

案例 4:SNMPv3客戶端組實施不適用於APIC

問題:不在客戶端組策略中的SNMP客戶端可以使用SNMPv3成功查詢APIC，即使從枝葉/主幹節點進行的同一查詢失敗也是如此。

根本原因：這是已知**的警告**。客戶端組策略（基於iptables的源IP實施）不適用於SNMPv3 GETs/Walks到APIC控制器。無論客戶端組配置如何，任何主機都可以通過SNMPv3查詢APIC。在枝葉和主幹交換機上，客戶端組實施對SNMPv2c和SNMPv3的工作方式相同。

緩解：在APIC上使用管理合約過濾器按源子網限制SNMP訪問。客戶端組對枝葉/主幹節點有效。對於使用SNMPv3的APIC，依靠管理合約基於源的過濾作為訪問控制機制。

案例 5:SNMP查詢成功，但MIB資料不完整或過時

問題:SNMP GET/WALK返回資料，但某些MIB OID返回空值或過時的值。特別是，介面統計資訊或運行狀態資料並不反映當前交換矩陣狀態。

操作檢查：確認正在查詢哪個APIC。每個APIC僅返回其本地資料的MIB對象。在要查詢的APIC上運行show snmp summary，並將結果與預期結果進行比較。對於交換機級資料(IF-MIB、entityMIB)，請直接查詢交換機，而不是APIC。

根本原因：查詢APIC以獲取枝葉級MIB資料。每個APIC僅為其自己的託管對象提供MIB對象。必須通過直接輪詢每個枝葉和主幹來檢索交換機級資料（介面統計資料、CPU、記憶體、環境感測器）。

解決方案：將NMS配置為直接輪詢枝葉和主幹管理IP以獲取介面和硬體MIB資料。僅對APIC本地MIB（與APIC伺服器硬體相關的實體、FRU、進程和感測器）使用APIC管理IP。

案例 6:SNMP適用於枝葉/骨幹，但不適用於APIC

問題:從NMS到枝葉和主幹節點的SNMPv2c GET成功。同一個NMS無法輪詢APIC。

配置檢查：APIC SNMP要求明確的管理合約允許UDP 161。導航到**Tenants > mgmt**，然後檢查OOB/INB合約及其用於UDP 161的過濾器。

操作檢查：在APIC上，運行iptables -S | grep 161。如果在fp-137（或等效的OOB合約）鏈下沒有出現UDP 161的ACCEPT規則，則UDP 161的合約篩選器缺失或未部署。

```
apic1#

iptables -S | grep 161

-A fp-137 -s 10.0.0.0/8 -p udp -m udp --dport 161 -j ACCEPT     <--- permit SNMP from the management su
-A fp-137 -s 172.18.0.0/16 -p udp -m udp --dport 161 -j ACCEPT  <--- permit SNMP from INB management su
```

如果沒有這些規則，請將UDP 161的過濾器條目新增到管理合約主題中，然後重新驗證。

根本原因：管理合約丟失或配置錯誤。在ACI 5.x中，APIC節點嚴格執行管理合約 — 除非存在顯式允許，否則會丟棄SNMP資料包。

解決方案：

1. 導航到**Tenants > Mgmt > Security Policies > Out-Of-Band Contracts**。
2. 展開OOB合約，選擇主題，然後驗證/新增UDP埠**161的過濾器**。
3. 如果NMS通過INB管理到達APIC，請重複帶內合約。
4. 使用iptables -S驗證 |儲存後,APIC上的grep 161。

案例 7:SNMP iptables規則缺失或不正確

問題:show snmp summary顯示已應用SNMP策略，但iptables -S | grep snmp不返回任何規則，或NMS客戶端IP不在規則中。

操作檢查：確認snmpd正在使用pidf snmpd運行。如果snmpd正在運行，但iptables沒有SNMP規則，則進程是在部署客戶端組策略之前啟動的。重新啟動snmpd以在重新啟動數小於250時強制規則重新程式設計：

```
<#root>

leaf101#

pidof snmpd

5881

leaf101# show system internal sysmgr service name snmpd
Service "snmpd" ("snmpd", 127):
UUID = 0x1A, PID = 5881, SAP = 1545
State: SRV_STATE_HANDSHAKED (entered at time Mon Aug 25 19:23:50 2025).

Restart count: 3


Time of last restart: Mon Aug 25 19:23:48 2025.
Previous PID: 32080
Reason of last termination: SYSMGR_DEATH_REASON_FAILURE_SIGNAL
Tag = N/A
Plugin ID: 0
leaf101#

kill -9 5881
```

ACI進程管理器將自動重新啟動snmpd。重新啟動後，驗證：

```
<#root>

leaf101#

iptables -S | grep -i snmp
```

現在應顯示snmp_rules鏈和每個VRF客戶端ACCEPT規則。

根本原因：snmpd進程在客戶端組策略完全部署到節點之前重新啟動或啟動，使iptables沒有SNMP訪問規則。

附註：在APIC上，tcpdump/code>命令僅供root使用者使用。對於APIC和交換機，iptables命令僅對root使用者可用。案例 4:SNMPv3客戶端組實施未處理APIC問題：不在客戶端組策略中的SNMP客戶端可以使用SNMPv3成功查詢APIC，即使從枝葉/主幹節點進行的同一查詢失敗也是如此。根本原因：這是已知警告。客戶端組策略（基於iptables的源IP實施）不適用於SNMPv3 GETs/Walks到APIC控制器。無論客戶端組配置如何，任何主機都可以通過SNMPv3查詢APIC。在枝葉和主幹交換機上，客戶端組實施對SNMPv2c和SNMPv3執行相同的操作。緩解：在APIC上使用管理合約過濾器按源子網限制SNMP訪問。客戶端組對枝葉/主幹節點有效。對於使用SNMPv3的APIC，依靠管理合約基於源的過濾作為訪問控制機制。 案例 5:SNMP查詢成功，但MIB資料不完整或陳舊問題：SNMP GET/WALK返回資料，但某些MIB OID返回空值或過時的值。特別是，介面統計資訊或運行狀態資料並不反映當前交換矩陣狀態。操作檢查：確認正在查詢哪個APIC。每個APIC僅返回其本地資料的MIB對象。對要查詢的APIC運行show snmp summary，並將結果與預期結果進行比較。對於交換機級資料(IF-MIB、entityMIB)，請直接查詢交換機，而不是APIC。根本原因：查詢APIC以獲取枝葉級MIB資料。每個APIC僅為其自己的託管對象提供MIB對象。必須通過直接輪詢每個枝葉和主幹來檢索交換機級資料（介面統計資料、CPU、記憶體、環境感測器）。解決方案：將NMS配置為直接輪詢枝葉和主幹管理IP以獲取介面和硬體MIB資料。僅對APIC本地MIB（與APIC伺服器硬體相關的實體、FRU、進程和感測器）使用APIC管理IP。 案例 6:SNMP適用於枝葉/主幹，但不適用於APIC問題：從NMS到枝葉和主幹節點的SNMPv2c GET成功。同一個NMS無法輪詢APIC。配置檢查：APIC SNMP需要允許UDP 161的明確管理合約。導航到Tenants > mgmt，然後檢查UDP 161的OOB/INB合約及其過濾器。操作檢查：在APIC上，運行iptables -S | grep 161。如果fp-137（或等效的OOB合約）鏈下未顯示UDP 161的ACCEPT規則，則缺少或未部署UDP 161的合約過濾器。apic1# iptables -S | grep 161 -A fp-137 -s 10.0.0.0/8 -p udp -m udp –dport 161 -j ACCEPT < – 允許來自管理子網的SNMP -A fp-137 -s 172.18.0.0/16 -p udp -m udp –dport 161 -j ACCEPT < – 允許來自INB管理子網的SNMP如果不存在這些規則，請將UDP 161的過濾器條目新增到管理合約主題並重新驗證。根本原因：管理合約丟失或配置錯誤。在ACI 5.x中，APIC節點嚴格執行管理合約 — 除非存在顯式允許，否則會丟棄SNMP資料包。解決方案：導航到Tenants > Mgmt > Security Policies > Out-Of-Band Contracts。展開OOB合約，選擇主題，然後驗證/新增UDP埠161的過濾器。如果NMS通過INB管理到達APIC，請重複帶內合約。使用iptables -S驗證 儲存後，APIC上的|grep 161。 案例 7:SNMP iptables規則不存在或不正確的問題：show snmp summary顯示SNMP策略已應用，但iptables -S | grep snmp不返回任何規則，或者規則中沒有NMS客戶端IP。操作檢查：確認snmpd正在使用pidf snmpd運行。如果snmpd正在運行，但iptables沒有SNMP規則，則進程是在部署客戶端組策略之前啟動的。重新啟動snmpd以在重新啟動數小於250時強制規則重新程式設計：leaf101# pidf snmpd 5881leaf101# show system internal sysmgr service name snmpdService "snmpd"("snmpd", 127):UUID = 0x1A，PID = 5881,SAP = 1545狀態：SRV_STATE_HANDSHAKED（在8月25日週一19:23:50 2025輸入）。重新啟動計數：3上次重新啟動時間：2025年8月25日週一19:23:48 2025。上一個PID:32080上次終止原因：SYSMGR_DEATH_REASON_FAILURE_SIGNALTag = N/ALugin ID:0 leaf101# kill -9 5881 ACI進程管理器將自動重新啟動snmpd。重新啟動後，驗證：leaf101# iptables -S | grep -i snmp現在應顯示snmp_rules鏈和每個VRF客戶端ACCEPT規則。根本原因：snmpd進程在客戶端組策略完全部署到節點之前重新啟動或啟動，使iptables沒有SNMP訪問規則。

用於擴展故障排除的日誌檔案

當上述驗證步驟無法解決問題時，枝葉、主幹和APIC節點上的以下日誌檔案包含與SNMP相關的診斷資訊：

<#root>

leaf101#

zgrep "snmp" /var/log/dme/log/svc_ifc_dbgrelem.log*

leaf101#

zgrep "snmpd" /var/log/dme/log/svc_ifc_dbgrelem.log*

leaf101#

zgrep "snmpd_log" /var/log/dme/log/*

這些日誌包含通過show snmp summary不可見的snmpd重新啟動事件、策略部署事件以及社群/客戶端配置錯誤。

參考資料

- 思科APIC系統管理配置指南5.x版 — 管理SNMP
- Cisco ACI MIB快速參考指南

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。