Cisco ACIファブリックのSNMPのトラブルシューティング

はじめに

このドキュメントでは、ACIリリース5.x以降のCisco ACIでSNMPを設定、確認、およびトラブルシューティングする方法について説明します。リーフ/スパイン型スイッチとAPICコントローラの最も一般的な障害シナリオにおける、SNMPポリシーモデル、必要な管理契約、トラップ設定、CLIおよび管理対象オブジェクト(MO)クエリを使用した動作検証、構造化されたトラブルシューティングワークフローについて説明します。

バックグラウンド情報

このドキュメントの資料は、Tomas de Leonによって執筆されたCisco ACI Solutions Delivery Team(ACI)の内部テクニカルノート『ACIのSNMP：概要、設定、トラブルシューティング、および注意/問題』から引用され、Cisco APICシステム管理設定ガイド（リリース5.x）およびCisco ACI MIBクイックリファレンスガイドで補足されています。

概要

ACIのSNMPアーキテクチャ

SNMP(Simple Network Management Protocol)は、ネットワーク管理とモニタリングを制御するUDPベースのプロトコルです。ACIでは、SNMPは各管理対象エンティティで独立して動作します。すべてのリーフスイッチ、スパインスイッチ、およびAPICコントローラは、それぞれ独自のSNMPエージェントです。各エージェントは個別にポーリングまたはモニタする必要があります。

ACIは次のSNMP機能をサポートしています。

- 読み取り操作(Get、GetNext、BulkGet、Walk)：リーフ/スパインスイッチおよびAPICコントローラでサポートされます。
- 通知（トラップ）：リーフ/スパインスイッチおよびAPICコントローラでサポートされるSNMPv1、v2c、およびv3トラップ。
- SNMPv3：リーフ/スパインスイッチおよびAPICコントローラでサポートされます。

- 書き込み操作（設定）：どのACIデバイスでもサポートされません。
- IPv6:SNMPはIPv4でのみサポートされます。

注:APICクラスタでは、各APICが自身にローカルなMIBオブジェクトを提供します。各APICを個別にポーリングする必要があります。クラスタ全体のSNMP集約はありません。同様に、各リーフスイッチとスパインスイッチに対して個別にクエリーを実行する必要があります。

APICでのSNMPDアーキテクチャ

APICは、次の2つの内部コンポーネントを持つsnmpdプロセスを実行します。

- Agent:SNMPプロトコル処理とセッション管理を処理するオープンソースのnet-snmpエージェント（バージョン5.7.6以降）。
- DME（データモデルエンジン）:APIC Management Information Tree（MIT；管理情報ツリー）とインターフェイスし、管理対象オブジェクト(MO)を読み取って、MO属性をSNMPオブジェクト形式に変換します。SNMPトラップは、MOで発生したイベントと障害から生成されます。

SNMPポリシーモデルと導入チェーン

ACIはSNMPにポリシー駆動型モデルを使用します。SNMP設定は、snmpPol管理オブジェクトとして抽象化されます。また、ファブリックを任意のノードに展開する前に、ファブリックのポッドポリシーグループに関連付ける必要があります。完全な導入チェーンは次のとおりです。

1. SNMPポリシー(`snmpPol`)：管理状態、コミュニティストリング、クライアントグループポリシー(ACL)、およびSNMPv3ユーザを定義します。
2. ポッドポリシーグループ：他のポッドレベルのポリシー（BGP、ISIS、NTPなど）とともにSNMPポリシーを参照します。
3. ポッドプロファイルセレクタ：ファブリックポッドにポッドポリシーグループを適用します。

また、SNMPトラップの設定には次が必要です。

1. SNMP Monitoring Destination Group(`snmpGroup`)：トラップ宛先ホスト、ポート、SNMPバージョン、およびコミュニティを定義します。
2. Monitoring Sources(snmpSrc)：宛先グループを3つの別個のモニタリングポリシースコープ（ファブリックデフォルト、ファブリック共通ポリシー、およびアクセスポリシーデフォルト）にリンクします。

APICノードには、UDPポート161（SNMP要求）およびUDPポート162（SNMPトラップ）を許可する管理契約が必要です。リーフノードとスパインノードにも正しいiptablesルールが必要です。このルールは、クライアントグループポリシーの設定時に自動的にプログラムされます。

サポートされたMIB

APICでサポートされるMIBは次のとおりです。

- エンティティMIB:PhysicalTable
- シスコエンティティ拡張MIB:PhysicalProcessorTable、LEDTable
- シスコエンティティFRU制御MIB:PowerSupplyGroupTable、PowerStatusTable、FanTrayStatusTable、PhysicalTable
- Cisco Entity Sensor MIB:SensorValueTable、SensorThresholdTable
- シスコプロセスMIB:CPUTotalTable、ProcessTable、ProcessExtRevTable、ThreadTable

リーフスイッチとスパインスイッチは、IF-MIB、IP-MIB、CISCO-CDP-MIB、CISCO-ENTITY-QFP-MIB、および完全なCISCO-ENTITY-FRU-CONTROL-MIBスイートを含む標準のNX-OS MIBを公開します。

APICで生成されるSNMPトラップには、cefcFRUInserted、cefcFRURemoved、cefcFanTrayStatusChange、cefcModuleStatusChange、entSensorThresholdNotification、cefcPowerStatusChange、cpmCPURisingThreshold、cpmCPUFallingThresholdがあります。

ACIでのSNMPの設定

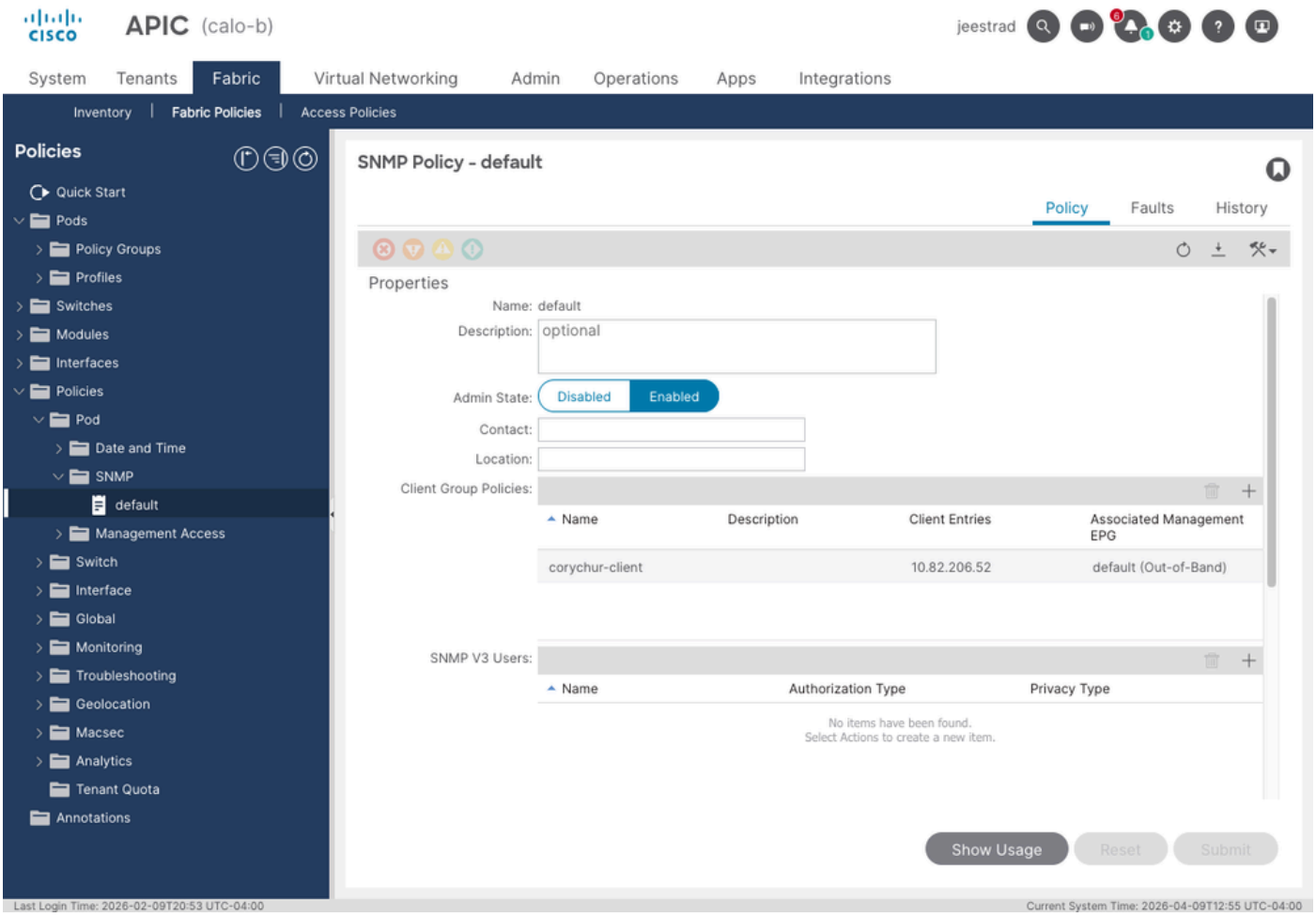
注：このセクションでは、以降の検証およびトラブルシューティングのセクションのコンテキストとして、設定ワークフローの要約を示します。設定手順の詳細については、『Cisco APICシステム管理設定ガイド』を参照してください。

ステップ1:SNMPポリシーの設定

Fabric > Fabric Policies > Policies > Pod > SNMPの順に移動します。SNMPポリシー(通常はdefault)を選択（または作成）します。設定例:

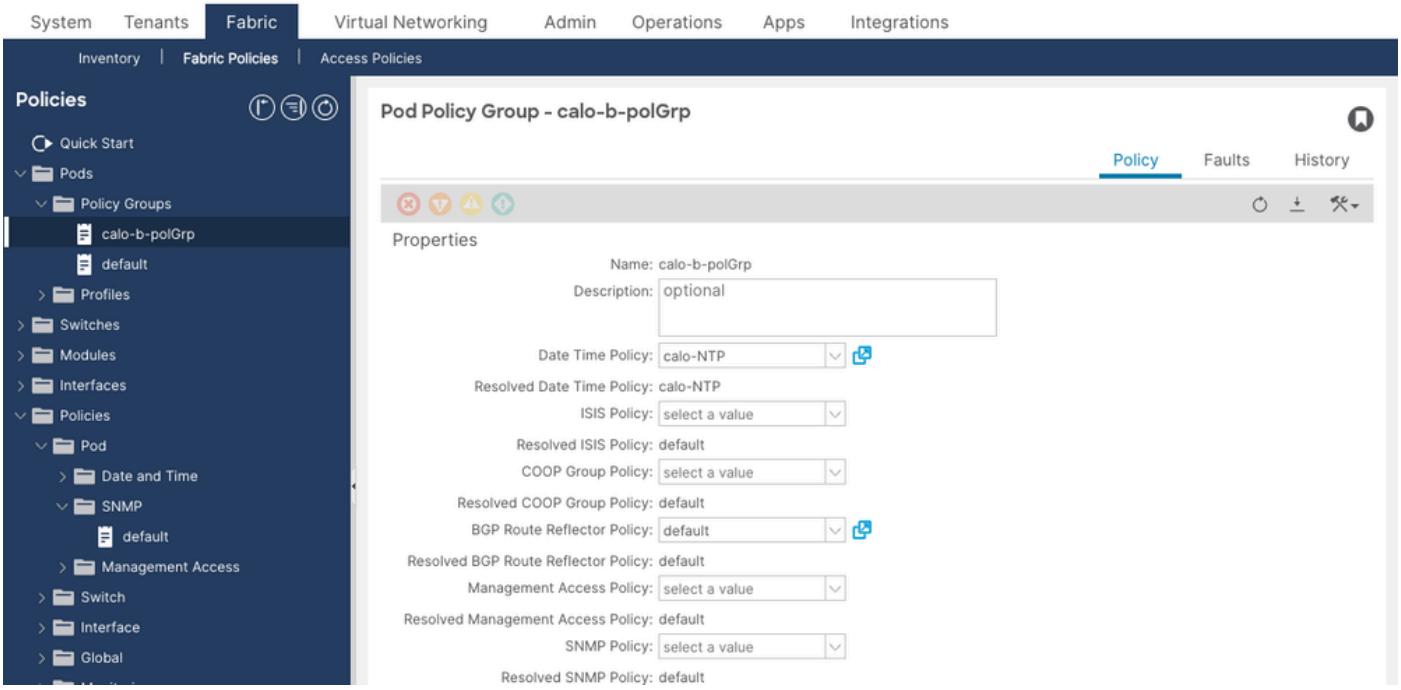
- Admin State:Enabledに設定します。
- Community Policies:NMSで使用するコミュニティストリングを追加します。
- Client Group Policies:1つ以上のクライアントグループプロファイルを定義し、各プロファイルで許可されるSNMPクライアントIP(Outside-of-Band)および関連する管理EPG（Out-of-BandまたはIn-Band）を指定します。

- SNMPv3ユーザ:SNMPv3を使用する場合は、ここで認証とプライバシーパラメータを使用してユーザを追加します。



ステップ2:SNMPポリシーをポッドポリシーグループに関連付けます。

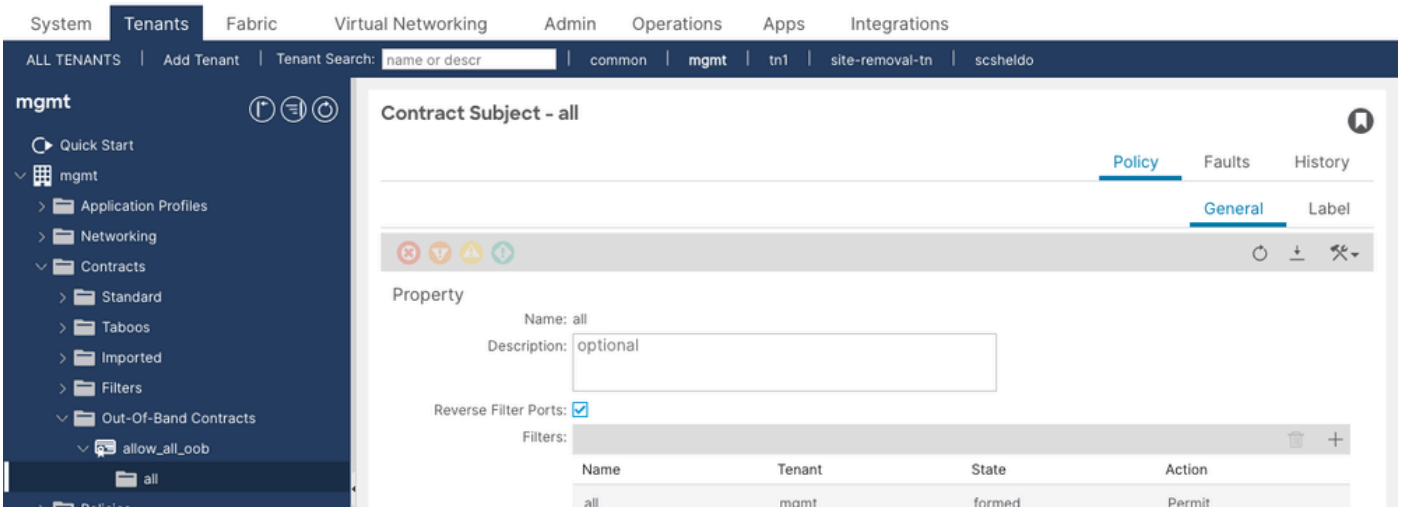
Fabric > Fabric Policies > Pods > Policy Groupsの順に移動します。アクティブなポッドポリシーグループ(通常defaultという名前)を選択します。 SNMP Policyフィールドが、ステップ1で作成したSNMPポリシーをポイントするように設定します。Resolved SNMP Policyフィールドに正しいポリシー名が表示されていることを確認します。



次に、Fabric > Fabric Policies > Pods > Profilesの順に移動し、デフォルトのポッドプロファイルを展開して、アクティブなセレクタが正しいポッドポリシーグループを参照していることを確認します。

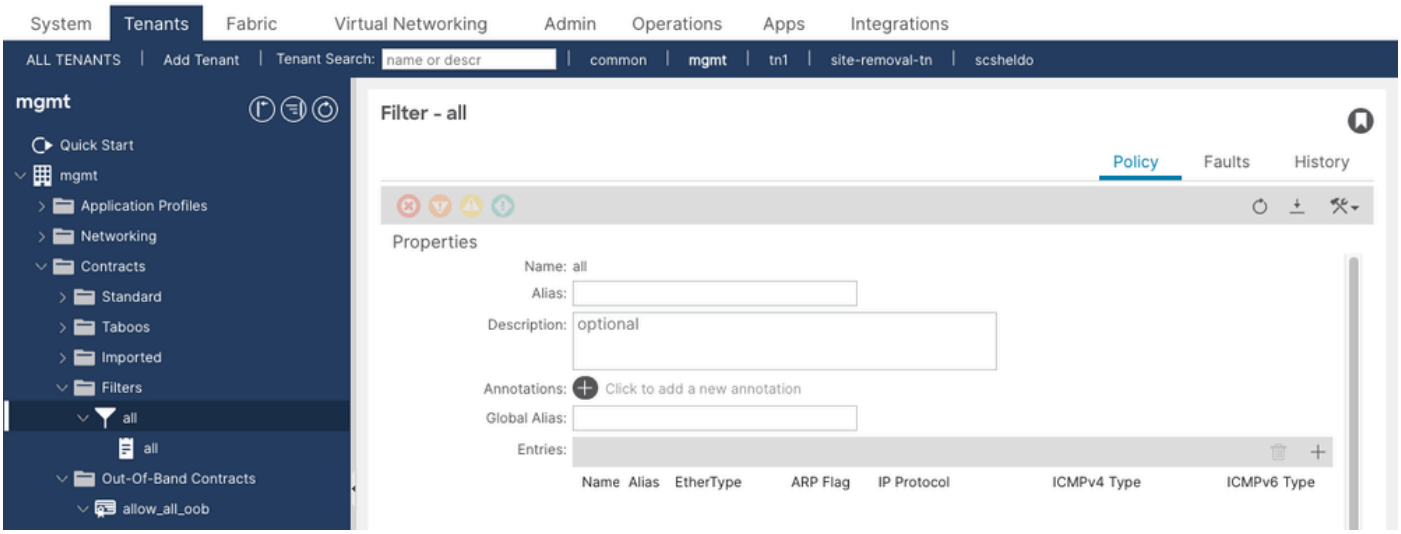
ステップ3:UDPポート161の管理コントラクトを設定します

Tenants > mgmt > Contracts > Out-Of-Band Contractsの順に移動します。アクティブなOOBコントラクトのサブジェクトが、UDPポート161（SNMP要求）を許可するフィルタエントリを参照していることを確認します。 APICでこのコントラクトがないと、すべてのSNMP GET/WALKパケットは通知なしにドロップされます。



コントラクト対象に添付されるフィルタエントリには、EtherType IP、プロトコルUDP、および宛先ポート161を含むエントリが含まれている必要があります。上記の例は、allow-all（未指定プ

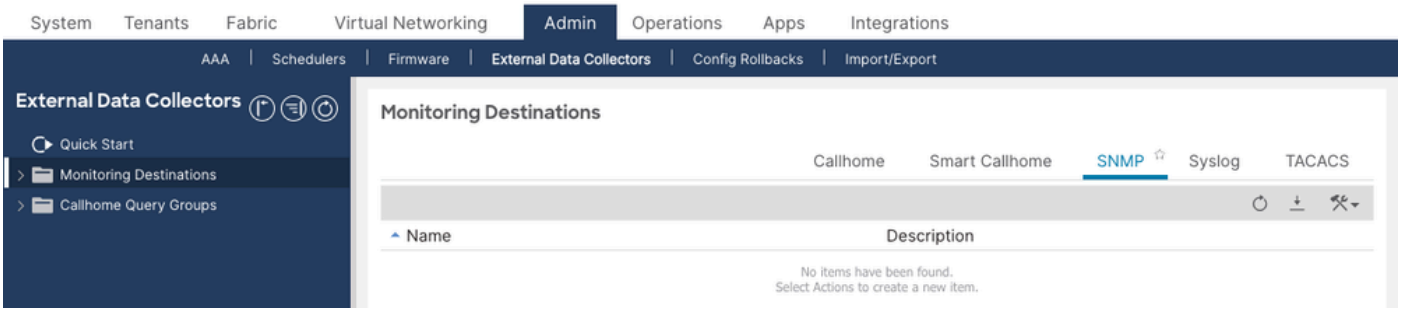
ロトコル）フィルタを示しています。これはSNMPを許可しますが、実稼働環境で推奨されるよりも範囲が広くなります。特定のUDP/161およびUDP/162エントリを持つ専用のSNMPフィルタエントリが推奨されます。



注：以前のACIファームウェアバージョンでは、特定のポートが常にリーフノードとスパインノードで開いており、SNMPに管理契約は必要ありませんでした。ACI 5.xでは、契約はAPICノードに必要です。リーフノードとスパインノードは、管理コントラクトではなく、クライアントグループポリシーから取得されたiptablesルールを使用します。

ステップ4:SNMPトラップ送信先の設定

Admin > External Data Collectors > Monitoring Destinations > SNMPの順に移動します。右クリックして、Create SNMP Monitoring Destination Groupを選択します。SNMPタブには、設定されているすべての宛先グループが表示されます。空のテーブルは、トラップの宛先がまだ設定されていないことを意味します。



定義：

- グループ名
- トラップ宛先：ホスト名/IP、UDPポート（デフォルトは162）、SNMPバージョン、コミュ

ニティストリング、および管理EPG

ステップ5：モニタリングソースの設定

モニタリングソースは、SNMP宛先グループを、トラップを生成するイベントと障害を制御するモニタリングポリシーにリンクします。モニタリングソースは、次の口ケーションの3つすべての口ケーションで設定する必要があります。設定されていないと、一部のノードタイプからのトラップが送信されません。

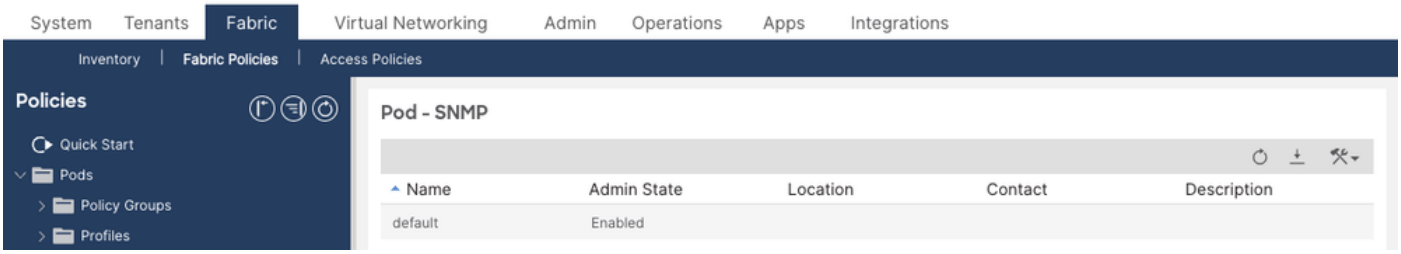
- Fabric > Fabric Policies > Policies > Monitoring > default > Callhome/Smart Callhome/SNMP/Syslog/TACACS（ファブリックインフラストラクチャイベントを対象）
- Fabric > Fabric Policies > Policies > Monitoring > Common Policy > Callhome/Smart Callhome/SNMP/Syslog/TACACS（ファブリック全体の一般的なイベントを対象）
- Fabric > Access Policies > Policies > Monitoring > default > Callhome/Smart Callhome/SNMP/Syslog（アクセス/インフラストラクチャイベントを対象）

それぞれの場所で、送信元タイプとしてSNMPを選択し、ステップ4で作成した宛先グループを参照する新しいSNMPソースを作成します。

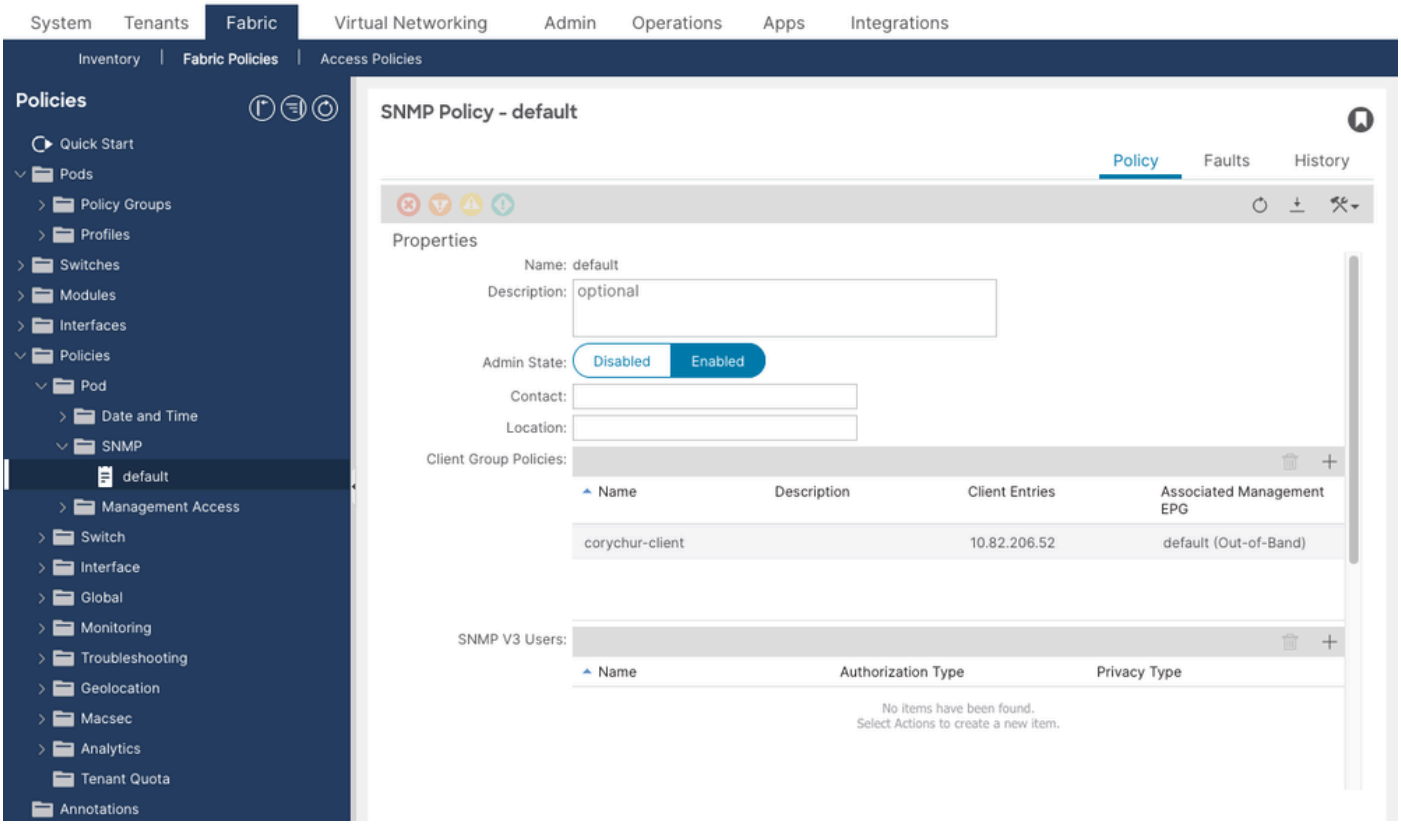
設定の確認

SNMPポリシー展開の確認

Fabric > Fabric Policies > Policies > Pod > SNMPの順に移動し、default SNMPポリシーが存在し、そのAdmin StateがEnabledに設定されていることを確認します。ポリシーグループリストには、設定されているすべてのSNMPポリシーとその管理状態が一目でわかります。



詳細な検証を行うには、ポリシー名をクリックして開きます。Admin State toggleがEnabledに設定されており、許可されたすべてのNMSホストがクライアントグループポリシーに関連付けられた管理EPGとともにリストされていることを確認します。



APICで次のMOクエリを実行して、SNMPポリシーがファブリックに存在し、有効になっていることを確認します。

<#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
```

adminStがdisabledになっている場合は、SNMPはどのノードでも機能しません。APIC GUIのFabric > Fabric Policies > Policies > Pod > SNMP > defaultで有効にします。

コミュニティストリング設定の確認

<#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
```

コミュニティが返されない場合、または名前がNMSで使用されているものと一致しない場合は、SNMPポリシーでコミュニティストリングを追加または修正します。

クライアントグループポリシーの確認（SNMPアクセスコントロール）

クライアントグループポリシーは、SNMP GET/WALKアクセスのACLとして機能します。各ポリシーは、どのクライアントIPアドレスがどの管理VRF経由でリーフ/スパインノードをポーリングできるかを指定します。リーフ/スパインノードでは、これらのポリシーはiptablesルールに変換されます。

<#root>

apic1#

moquery -c snmpClientGrpP -x query-target=children

Total Objects shown: 3

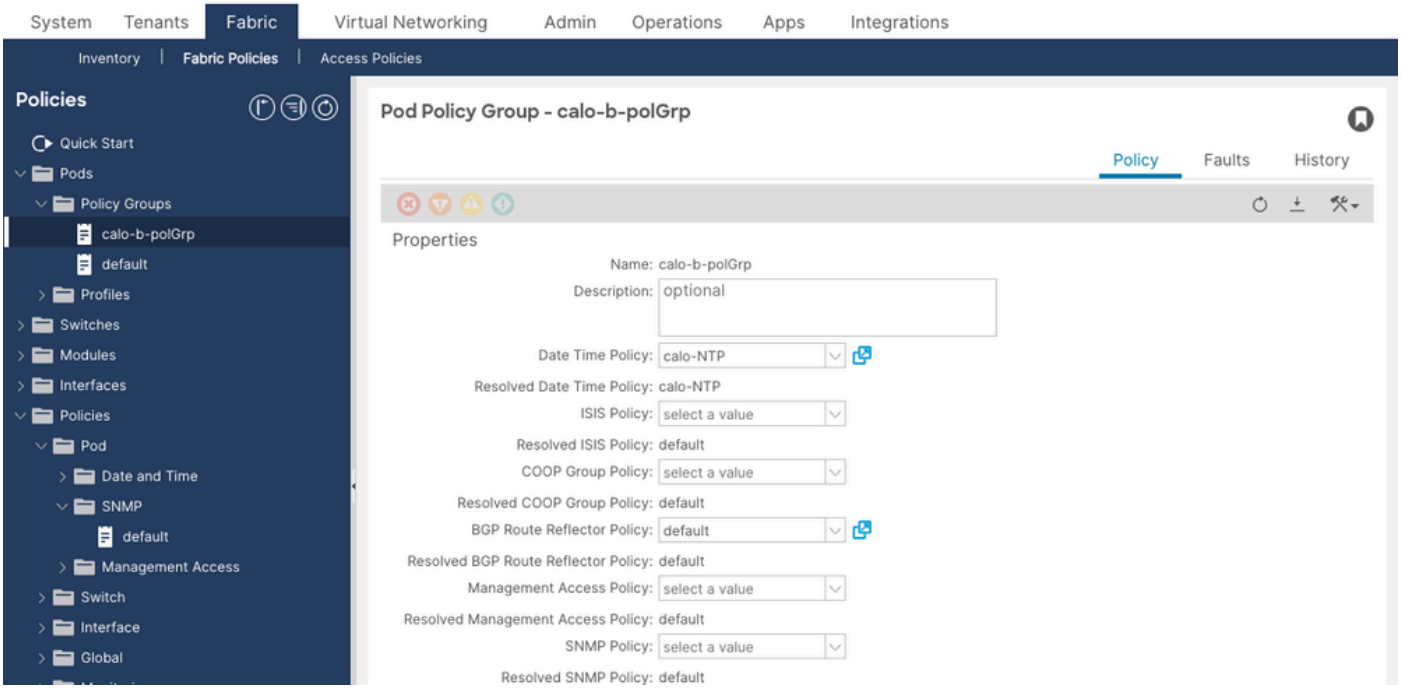
```
# snmp.ClientP
addr         : 10.1.1.50                   <--- NMS server IP
dn           : uni/fabric/snmppol-default/clgrp-NMS-Clients/client-[10.1.1.50]
name         : nms-server1

# 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
```

NMSサーバのIPアドレスがクライアントエントリに存在することを確認します。クライアントIPが欠落している場合、そのホストからのSNMP GET/WALK要求は、リーフ/スパインノードのiptablesによってドロップされます。

注:SNMPv3の警告：SNMPv3の使用時には、APICにクライアントグループポリシーは適用されません。APICへのSNMPv3 GET/WALKは、クライアントグループの設定に関係なく許可されます。APICでのSNMPv3に対するクライアントグループの適用は、既知の制限事項です。リーフスイッチとスパインスイッチでは、クライアントグループの強制は、SNMPv2cとSNMPv3の両方で同じように動作します。

ポッドポリシーグループがSNMPポリシーを参照していることを確認する

Fabric > Fabric Policies > Pods > Policy Groupsの順に移動し、アクティブなポッドポリシーグループを開きます。SNMP Policyドロップダウンフィールドに目的のSNMPポリシーが設定され、Resolved SNMP Policyフィールドに同じ名前が表示されていることを確認します。ポリシーが存在しないか未解決の場合、SNMP設定がスイッチにプッシュされることはありません。



上記のスクリーンショットでは、SNMP Policyフィールドに「select a value」（空）が表示され、解決済みSNMPポリシーに「default」が表示されています。これは、ポリシーがファブリックのデフォルトから継承されているが、明示的に設定されていないことを意味します。あいまいさを避けるために、SNMP Policyフィールドを明示的に設定することをお勧めします。

REST APIによる確認：

<#root>

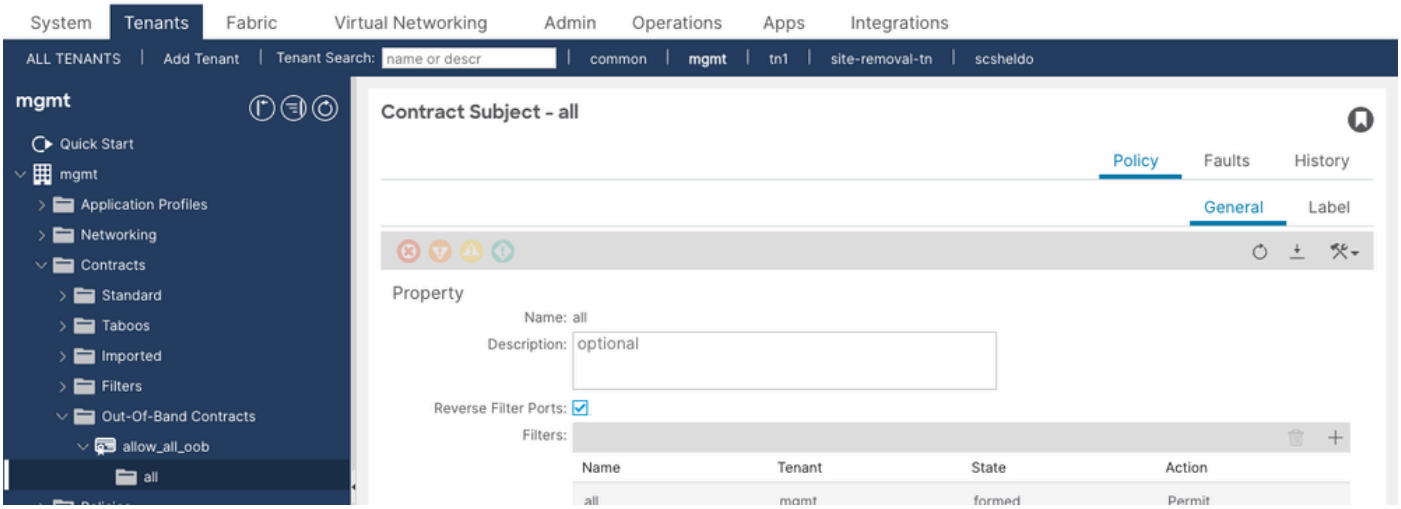
```
apic1#

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"
```

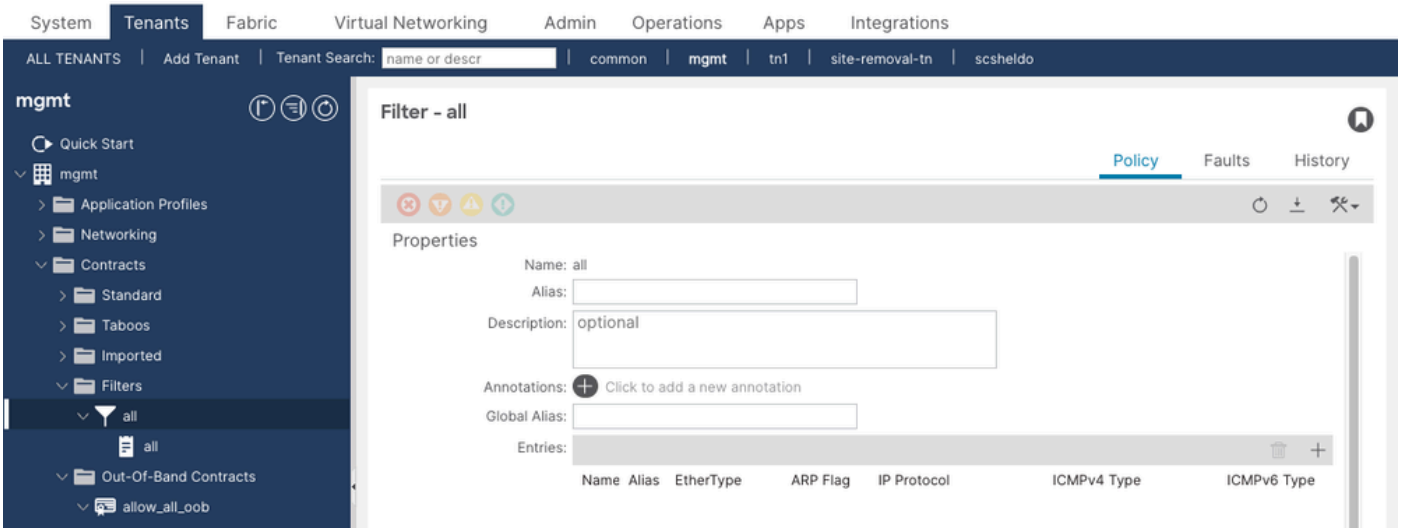
stateがformedでない場合、SNMPポリシー関係は壊れています。ポッドポリシーグループでSNMPポリシーを再選択して送信します。

UDP 161（APICノード）の管理コントラクトの確認

Tenants > mgmt > Contracts > Out-Of-Band Contracts（および、INB管理を使用している場合はIn-Band Contracts）に移動します。 アクティブなOOBコントラクトを開き、Policyタブをクリックします。SubjectがUDPポート161を許可するフィルタを参照していることを確認します。



件名で参照されているフィルタを展開し、そのエントリにEtherType IP、プロトコルUDP、宛先ポート161のエントリが含まれていることを確認します。フィルタエントリにより、OOB管理コントラクトを介してAPICに許可されるトラフィックが決まります。



フィルタには次のように表示されます。

- EtherType:IP
- IPプロトコル：UDP
- 宛先ポート：自：161
- 宛先ポート：161

また、OOBインターフェイス経由でSNMPトラップを発信するようにAPICに設定する場合は、UDPポート162が許可されていることを確認します。

MOクエリによる確認：

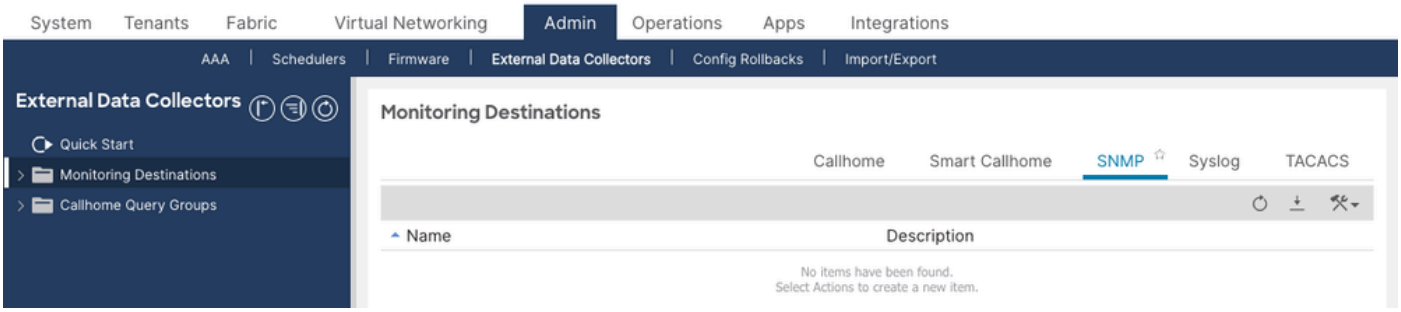
<#root>

```
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
```

結果が返されない場合は、UDP 161のフィルタは存在しません。1つを管理契約に追加します。

SNMPトラップ送信先の設定の確認

Admin > External Data Collectors > Monitoring Destinations > SNMPの順に移動し、設定されているすべてのSNMP宛先グループを表示します。空のリストは、トラップの宛先が設定されておらず、どのノードからもトラップが送信されないことを意味します。



<#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
```

トラップの宛先IP、ポート、バージョン、コミュニティストリング、および管理VRF(OOBの場合は`mgmt:inb`または`management`)が環境に一致していることを確認します。VRFは、宛先に割り当てられた管理EPGと一致する必要があります。

3つのスコープすべてに監視送信元が設定されていることを確認します。

SNMPソースは、3つすべてのモニタリングポリシースコープに存在する必要があります。いずれのスコープでも送信元が欠落している場合、関連するイベントからのトラップは転送されません。

<#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
```

3つのいずれかが欠落している場合は、GUIを使用して、対応するモニタリングポリシーで欠落しているSNMPソースを作成します。

動作検証

show snmp summary(APIC)を使用したSNMP状態の確認

各APICで次のコマンドを直接実行して、SNMPエージェントが実行されており、設定が適用されていることを確認します。

<#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
```

出力で確認する内容：

- Admin Stateは`enabled`である必要があります。
- コミュニティは、NMSで使用するように設定されているものと一致している必要があります。
- Client-Groupは、正しい管理EPGを持つ、許可されたすべてのNMS IPをリストする必要があります。
- Host(trap destination)は、正しいポートとバージョンのNMSトラップレシーバをリストする必要があります。

show snmp summary（リーフ/スパイン）を使用してSNMPの状態を確認する

<#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
```

出力で確認する内容：

- Admin Stateは`enabled`で、pidで実行されている必要があります。`disabled`と表示されている場合は、SNMPポリシーが適用されていないか、ポッドポリシーチェーンが壊れています。
- Community Statusはokである必要があります。`error`ステータスは、ポリシーの導入に問題があることを示します。
- 各NMSホストのクライアントVRFは、管理EPGのVRF(インバンドの場合は`mgmt:inb`、OOBの場合は`management`)と一致している必要があります。
- トラップホストは、正しいVRFコンテキストで宛先をリストする必要があります。

snmpdプロセスが実行されていることの確認

リーフまたはスパイン：

<#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
```

APICで次の手順を実行します。

<#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
```

リーフまたはスパインでsnmpdプロセスが見つからない場合、SNMPはそのノードで実行されて

いません。SNMPポリシーの管理状態が有効になっており、ポッドポリシーチェーンが正しく設定されていることを確認します。

スポイラー（参照用に強調表示）

SNMPポートがリッスンしていることを確認する

<#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           :::*
```

ポート161がLISTEN状態にない場合は、snmpdプロセスが実行されていないか、ポートへのバインドに失敗しています。

リーフ/スパインのiptablesルールの確認

クライアントグループポリシーは、各リーフおよびスパインでiptablesルールに変換されます。ルールを検査するには、次の手順に従います。

<#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)
```

ファブリックの正しいVRF IDを特定するには、次のコマンドを実行します。

<#root>

```
leaf101#
```

```
show vrf

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

iptablesルール内のVRF IDは、`show vrf`レポートの内容と一致している必要があります。クライアントIPがiptablesルールにない場合、snmpdプロセスが実行されていても、そのホストからのSNMP要求は通知なしにドロップされます。

カウンタを使用して、SNMPパケットが一致またはドロップされたかどうかを確認します。

<#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
```

注:SNMPが実行されていても、iptablesでsnmp_rulesチェーンが表示されない場合、またはチェーンが空の場合は、snmpdプロセスを再起動してiptablesルールを強制的に再プログラミングできます。snmpd PIDへのSIGKILLの送信は安全です。ACIプロセスマネージャ(`policed`)は自動的に再起動します。`pidof snmpd`を実行してPIDを取得し、`kill -9 [snmpd_pid]`を実行します。10 ~ 15秒後に`pidof snmpd`を使用して新しいPIDを確認します。

SNMPポートがleaf101# netstat -lutn | grep 161アクティブなインターネット接続（サーバのみ）Proto Recv-Q Send-Q Local Address Foreign Address State tcp 0 0 0.0.0.0:161 0.0.0.0をリッスンしていることを確認します。 * LISTEN <— SNMPエージェントは要求を受け付けていますudp 0 0 0.0.0.0:161 0.0.0.0:* udp6 0 :::161 ::*ポート161がLISTEN状態にない場合、snmpdプロセスは実行されていないか、ポートへのバインドに失敗しています。　リーフ/スパインクライアントグループポリシーのiptablesルールが、各リーフ/スパインのiptablesルールに変換されることを確認します。ルールを検査するには、次のコマンドを使用します。 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ポート161はsnmp_rulesチェーンにリダイレクトします。 -A snmp_rules -m vrf 2 -j vrf_2_snmp_rules <— VRF 2 = OOBmanagement -A snmp_rules -m -vrf —vrf —vrf 9 -j vrf_9_snmp_rules <— VRF 9 =インバンド管理 – A snmp_rules -j DROP < – デフォルトのドロップ。許可されたクライアントのみpass -A vrf_2_snmp_rules -s 10.1.1.50/32 -j ACCEPT < – 許可されたNMSクライアント(OOB VRF) -A vrf_9_snmp_rules -s 10.1.1.50/32 -j ACCEPT < – 許可されたNMSクライアント(INB VRF) ファブリックの正しいVRF IDを特定するには、次のコマンドを実行します。 leaf101# show vrf VRF-Name VRF-ID State Reason management 2 Up — mgmt:inb 9 Up — iptablesルール内のVRF IDは、show vrfレポートと一致している必要があります。クライアントIPがiptablesルールにない場合、snmpdプロセスが実行されていても、そのホストからのSNMP要求は通知なしにドロップされます。カウンタを使用して、SNMPパケットが一致またはドロップされたかどうかを確認します。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 IPs are missing注： SNMPは実行されているが、iptablesでsnmp_rulesチェーンが表示されない場合、またはチェーンが空の場合は、snmpdプロセスを再起動してiptablesルールを強制的に再プログラミングできます。SIGKILLをsnmpd PIDに送信しても安全です。ACIプロセスマネージャ（ポリシング）は自動的に再起動します。pidof snmpdを実行してPIDを取得し、-9 [snmpd_pid]を強制終了します。10 ~ 15秒後に新しいPIDをpidof snmpdで確認します。

SNMPポートへのネットワーク接続の確認

<#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
```

管理インターフェイスがアクティブで（RX-ERRの増分がない）、トラフィックが通過していることを確認します。eth0はOOB管理インターフェイスで、kpm_inbはスイッチのインバンド管理インターフェイスです。

tcpdumpを使用したSNMPトラップ送信の確認

トラップが生成され、リーフ/スパインノードから送信されていることを確認するには、適切なインターフェイスでトラフィックをキャプチャします。adminとしてノードにアクセスし、次のコマンドを使用します。

<#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
```

OOBの場合：

<#root>

```
leaf101#
```

tcpdump -i eth0 -f port 162 -vv

スポイラー（参照用に強調表示）

APICトラップ(INB)の場合：

<#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
```

注:APICでは、bond0.1100はインバンド管理インターフェイスのVLANサブインターフェイスです。1100を、使用しているインバンド管理EPG用に設定されているVLANカプセル化に置き換えます。APIC上のOOBキャプチャのインターフェイス名としてoobmgmtを使用します。

APICトラップ(INB)の場合：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.9.117.1.2.1.10548=E=1 :cisco.9.117.1.1.2.1.2.10548=2注：APICでは、bond0.1100がインバンド管理インターフェイスのVLANサブインターフェイスです。1100を、使用しているインバンド管理EPG用に設定されているVLANカプセル化に置き換えます。APICでのOOBキャプチャのインターフェイス名としてoobmgmtを使用します。

tcpdumpを使用したSNMP GET/WALK要求の確認

<#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
```

GetRequestは表示されるがGetResponseが表示されない場合、要求は受信されているが応答がない。snmpdプロセスとコミュニティストリングを確認します。要求も応答もない場合は、ノードに到達する前に要求がブロックされています（ルーティングとiptablesを確認してください）。

トラブルシューティングワークフロー

トリアージ決定ツリー

エンジニアがSNMPが動作していないと報告した場合に、このDecision Treeを使用します。観察された症状から始めて、ブランチに従って分離します。

症状：SNMP GET/WALK要求に応答がありません

1. APICのSNMP管理状態を確認します。`moquery -c snmpPol`を実行します。`adminSt`が`disabled`になっている場合は、イネーブルにしてステップ7に進みます。
2. snmpdプロセスを確認します。影響を受けるノードで、`ps aux | grep snmp`または`pidof snmpd`を実行します。実行中のプロセスがない場合、SNMPポリシーは展開されません。ポッドポリシーチェーン(SNMPポリシー→ポッドポリシーグループ→ポッドプロファイル)を確認します。
3. ポート161がリッスンしていることを確認します。`netstat -lutn | grep 161`を実行します。ポート161がLISTEN状態でない場合は、snmpdプロセスが失敗しています。`/var/log/dme/log/svc_ifc_dbgrelem.log*`からログを収集して、プロセスを再起動します。
4. ルーティングをチェックします。`show ip route vrf management`および`show ip route vrf mgmt:inb`を実行します。NMSホストへのルートが正しいVRFに存在することを確認します。
5. APICの管理契約を確認します。ターゲットがAPIC（リーフ/スパインではない）の場合、OOBまたはINB管理コントラクトでUDP 161が許可されていることを確認します。
6. ノードでtcpdumpを実行します。`tcpdump -i kpm_inb -f port 161 -vv`(またはOOBの場合は`eth0`)を実行します。 GetRequestが表示されてもGetResponseが続かない場合、要求はノードに到達しますが、snmpd is not responding – コミュニティストリングを確認します。要求がまったく表示されない場合は、アップストリーム（ルーティングまたはコントラクト）に問題があります。
7. 許可されたクライアントからテストします。クライアントグループにリストされているNMSホストから`snmpget -v2c -c [community] [node-ip] SNMPv2-MIB::sysDescr.0`を実行します。正常な応答により、SNMPが完全に動作していることが確認されます。

症状：NMSでSNMPトラップが受信されない

1. トラップ送信先の設定を確認します。`moquery -c snmpTrapDest`を実行します。NMSのIP、ポート、バージョン、およびコミュニティが、NMSの期待値と一致していることを確認します。
2. モニタリングソースが3つのスコープすべてに存在することを確認します。`moquery -c snmpSrc | egrep "snmp.Src|name|dn"`を実行します。 エントリが、`uni/fabric/monfab-default`、`uni/fabric/moncommon`、および`uni/infra/moninfra-default`の`monPolDn`値とともに存在することを確認します。欠落している場合は、対応するモニタリングポリシーにSNMPソースを追加します。
3. snmpdプロセスを確認します。トラップを送信するノードでsnmpdが実行されていることを確認します。
4. テストイベントを生成し、tcpdumpを使用してキャプチャします。インターフェイスをフラップするか、状態を変更してイベントを生成する。ノードで、`tcpdump -i kpm_inb -f port 162 -vv`を実行します。トラップトラフィックがネットワークに表示されない場合、イベントはトラップを生成していません。`incl`属性を再確認してモニタリングします(`faults`または`events`を含める必要があります)。
5. トラップレシーバへの接続をチェックします。管理VRFからトラップレシーバに到達可能であることを確認します。`show ip route vrf mgmt:inb`を実行すると、NMSホストへのパスが表示されます。
6. トラップがtcpdumpに表示されてもNMSに表示されない場合、問題はネットワーク側（ファイアウォール、ルーティング、またはNMS設定）にあります。NMSがACIノードの管理ソースIPからUDP 162をリッスンしていることを確認します。

一般的なシナリオ

シナリオ1:SNMPポリシーは有効になっているが、リーフ/スパインからデータが返されない

問題：APICのSNMPポリシーに「Admin State enabled」と表示されます。NMSはリーフの管理IPに到達できます。`snmpget`は応答なしでタイムアウトします。

設定確認：ポッドポリシーグループがSNMPポリシーを参照し、解決されたSNMPポリシーが正しい名前を示していることを確認します。ポッドポリシーグループのSNMPポリシーフィールドが空であるか、関係が形成されていない場合、スイッチでsnmpdプロセスが開始されない可能性があります。

動作確認：影響を受けるリーフにSSHで接続し、`show snmp summary`を実行します。APICで enabledと表示されていても出力に`Admin State: disabled`と表示される場合、ポリシーは展開されていません。ポッドポリシーチェーンで、ポッドポリシーグループが欠落しているか、正しく参照されていないかを確認します。

根本原因： SNMPポリシーがポッドポリシーグループにリンクされていないか、ポッドプロファイルセレクタが正しいポッドポリシーグループをこのポッドに適用していません。

ソリューション：

1. Fabric > Fabric Policies > Pods > Policy Groups > defaultの順に移動します。
2. SNMP Policyフィールドに、有効になっているSNMPポリシーが示されていることを確認します。
3. Fabric > Fabric Policies > Pods > Profilesの順に移動し、アクティブなセレクタがこのポッドポリシーグループを参照していることを確認します。
4. 保存後、2分以内にリーフの`show snmp summary`を再確認してください。

シナリオ2:SNMP GET/WALKが一部のNMSホストでは機能するが、他のホストでは機能しない

問題：1台のNMSサーバがACIノードを正常にポーリングできる。異なるサブネット上の2番目のNMSサーバは応答を受け取りません。

設定チェック：APICで`moquery -c snmpClientGrpP -x query-target=children`を実行します。2番目のNMSサーバのIPがクライアントエントリとしてリストされていることを確認します。欠落している場合、そのIPはsnmp_rulesチェーンの最後にあるiptables DROPルールによってブロックされます。

動作確認：該当するリーフで、OOBまたはINB管理コントラクトでUDP 161が許可されていることを確認します。SNMPポートを持つコントラクトまたはフィルタがない場合、要求はドロップされます。

根本原因：2番目のNMSサーバのIPがクライアントグループポリシーに含まれていません。

解決策：SNMPクライアントグループポリシーのFabric > Fabric Policies > Policies > Pod > SNMP > default > Client Group Policiesの下に、欠落しているNMS IPをクライアントエントリとして追加します。すべてのノードのiptablesルールは、ポリシーの保存後数分以内に更新されます。

シナリオ3:SNMPトラップが受信されない – トラップは生成されるが配信されない

問題： APIC障害テーブルに障害が表示されます。`moquery -c snmpTrapDest`は正しいNMS IPを示します。NMSはトラップを受信しません。

設定チェック：`moquery -c snmpSrc | egrep "snmp.Src|name|dn"`を実行します。 モニタリングソー

スが3つのスコープ(monfab-default、moncommon、moninfra-default)すべてに存在することを確認します。 よくある見落としとしては、送信元をFabric Defaultポリシーだけで設定し、アクセスポリシーイベントが失われることです。

動作確認：テストイベントをトリガーします（インターフェイスをadmin-down状態に切り替えるなど）。 該当するノードで、tcpdump -i kpm_inb -f port 162を実行します。トラップパケットがノードのインターフェイスに表示される場合、ACI側は機能しており、問題はNMSへのネットワークパス（ファイアウォール、ルーティング）にあります。 トラップがワイヤに表示されない場合、ACIモニタリングソースが欠落しているか、またはイベントタイプがソースのincl属性に含まれていません。

根本的な原因1:必要なスコープに1つ以上の監視ソースがありません。

根本原因2:送信元のincl属性を監視すると、生成されるイベントタイプが除外されます(例：incl:events without faultsは、障害ベースのトラップが送信されないことを意味します)。

ソリューション：

1. GUIで、3つのスコープ(Fabric Default、Fabric Common、Access Default)のそれぞれに不足しているモニタリングソースを追加します。 宛先グループを、設定したSNMP宛先グループに設定します。
2. incl属性に包括的なトラップカバレッジの監査、イベント、障害が含まれていることを確認します。
3. 変更後、テストイベントを再トリガーし、tcpdumpを再確認します。

スポイラー（参照用に強調表示）

注:APICでは、tcpdump/code>コマンドはrootユーザだけが使用できます。APICおよびスイッチの場合、iptablesコマンドはrootユーザだけが使用できます。

シナリオ4:APICでSNMPv3クライアントグループの適用が機能しない

問題：クライアントグループポリシーに含まれていないSNMPクライアントは、リーフ/スパインノードで同じクエリが失敗した場合でも、SNMPv3を使用してAPICに正常にクエリできます。

根本原因：これは既知の警告です。クライアントグループポリシー（iptablesベースのソースIPの適用）は、SNMPv3 GETs/WalksからAPICコントローラには適用されません。すべてのホストは、クライアントグループの設定に関係なく、SNMPv3経由でAPICに照会できます。リーフスイッチとスパインスイッチでは、クライアントグループの強制はSNMPv2cとSNMPv3で同じように機能します。

緩和策：APICで管理コントラクトフィルタを使用して、発信元サブネットごとにSNMPアクセスを制限します。クライアントグループは、リーフ/スパインノードに対して有効です。SNMPv3を使用するAPICでは、アクセス制御メカニズムとして管理契約のソースベースのフィルタリングに依存します。

シナリオ5:SNMPクエリは成功するが、MIBデータが不完全または古い

問題：SNMP GET/WALKがデータを返しますが、特定のMIB OIDが空または古い値を返します。特に、インターフェイスの統計情報や動作状態のデータには、現在のファブリックの状態は反映されません。

動作チェック：どのAPICが照会されているかを確認します。各APICは、ローカルのデータのMIBオブジェクトのみを返します。照会されるAPICでshow snmp summaryを実行し、その結果を予想される結果と比較します。スイッチレベルのデータ(IF-MIB、entityMIB)については、APICではなく、スイッチに直接クエリーを実行します。

根本原因：リーフレベルMIBデータのAPICへのクエリ。各APICは、独自の管理オブジェクトに対してのみMIBオブジェクトを提供します。スイッチレベルのデータ（インターフェイス統計情報、CPU、メモリ、環境センサー）は、リーフ/スパインごとに直接ポーリングして取得する必要があります。

解決策：インターフェイスおよびハードウェアMIBデータに対してリーフ/スパイン管理IPを直接ポーリングするようにNMSを設定します。APIC管理IPは、APICネイティブMIB（エンティティ、FRU、プロセス、APICサーバハードウェアに関連するセンサー）にのみ使用します。

シナリオ6:SNMPがリーフ/スパインに対して機能するが、APICに対しては機能しない

問題：NMSからリーフ/スパイン型ノードへのSNMPv2c GETが成功する。同じNMSはAPICをポーリングできません。

設定チェック：APIC SNMPには、UDP 161を許可する明示的な管理契約が必要です。**Tenants > mgmt**に移動し、OOB/INBコントラクトとUDP 161のフィルタを確認します。

動作確認：APICで、iptables -S | grep 161を実行します。UDP 161のACCEPTルールがfp-137（または同等のOOBコントラクト）チェーンの下に表示されない場合、UDP 161のコントラクトフィルタが存在しないか、展開されていません。

```
<#root>

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
```

これらのルールが存在しない場合は、UDP 161のフィルタエントリを管理契約の件名に追加し、再検証します。

根本的な原因：管理コントラクトが見つからないか、正しく設定されていません。ACI 5.xでは、APICノードは管理契約を厳密に適用します。明示的な許可が存在しない限り、SNMPパケットはドロップされます。

ソリューション：

1. **Tenants > mgmt > Security Policies > Out-Of-Band Contracts**の順に移動します。
2. OOBコントラクトを展開し、Subjectを選択して、**UDP**ポート161のフィルタを確認および追加します。
3. NMSがINB管理を介してAPICに到達する場合は、インバンド契約に対して手順を繰り返します。
4. 保存後、APICでiptables -S | grep 161を使用して確認します。

シナリオ7:SNMP iptablesルールがないか正しくない

問題：show snmp summaryを実行すると、**SNMPポリシー**は**適用されているが**、iptables -S | grep snmpによって**ルール**が返されないか、**NMSクライアントのIP**が**ルール**に**含まれていない。**

動作確認：pidof snmpdを使用してsnmpdが実行されていることを確認します。snmpdが実行されているが、iptablesにSNMPルールがない場合、クライアントのグループポリシーが展開される前にプロセスが開始されています。再起動の数が250未満の場合にsnmpdを再起動して、ルールの再プログラミングを強制します。

```
<#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プロセスマネージャが自動的にsnmpdを再起動します。再起動後、次のことを確認します。

```
<#root>

leaf101#

iptables -S | grep -i snmp
```

これで、snmp_rulesチェーンとVRFごとのクライアントACCEPTルールが表示されるようになります。

根本的な原因：クライアントのグループポリシーがノードに完全に展開される前にsnmpdプロセスが再起動または開始されたため、SNMPアクセスルールが設定されていないiptablesが残っています。

注：APICでは、tcpdump/code>コマンドはrootユーザだけが使用できます。APICおよびスイッチの場合、iptablesコマンドはrootユーザだけが使用できます。シナリオ4:APICでSNMPv3クライアントグループの適用が機能しない問題：クライアントグループポリシーに含まれていないSNMPクライアントは、リーフ/スパインノードで同じクエリが失敗した場合でも、SNMPv3を使用してAPICに正常にクエリできます。根本原因：これは既知の警告です。クライアントグループポリシー（iptablesベースのソースIPの適用）は、SNMPv3 GETs/WalksからAPICコントローラには適用されません。すべてのホストは、クライアントグループの設定に関係なく、SNMPv3経由でAPICに照会できます。リーフスイッチとスパインスイッチでは、クライアントグループの強制はSNMPv2cとSNMPv3で同じように機能します。緩和策：APICで管理コントラクトフィルタを使用して、発信元サブネットごとにSNMPアクセスを制限します。クライアントグループは、リーフ/スパインノードに対して有効です。SNMPv3を使用するAPICでは、アクセス制御メカニズムとして管理契約のソースベースのフィルタリングに依存します。　シナリオ5:SNMPクエリは成功するが、MIBデータが不完全か古い問題：SNMP GET/WALKがデータを返すが、特定のMIB OIDが空または古い値を返す。特に、インターフェイスの統計情報や動作状態のデータには、現在のファブリックの状態は反映されません。動作チェック：どのAPICが照会されているかを確認します。各APICは、ローカルのデータのMIBオブジェクトのみを返します。照会されているAPICでshow snmp summaryを実行し、その結果を期待するものと比較します。スイッチレベルのデータ(IF-MIB、entityMIB)については、APICではなく、スイッチに直接クエリーを実行します。根本原因：リーフレベルMIBデータのAPICへのクエリ。各APICは、独自の管理オブジェクトに対してのみMIBオブジェクトを提供します。スイッチレベルのデータ（インターフェイス統計情報、CPU、メモリ、環境センサー）は、リーフ/スパインごとに直接ポーリングして取得する必要があります。解決策：リーフ/スパイン管理IPをインターフェイスおよびハードウェアMIBデータに対して直接ポーリングするようにNMSを設定します。APIC管理IPは、APICネイティブMIB（エンティティ、FRU、プロセス、APICサーバハードウェアに関連するセンサー）にのみ使用します。　シナリオ6:SNMPはリーフ/スパインには機能するが、APICには機能しない問題：SNMPv2c NMSからリーフ/スパインのノードへのGETが成功する。同じNMSはAPICをポーリングできません。設定チェック：APIC SNMPには、UDP 161を許可する明示的な管理契約が必要です。Tenants > mgmtに移動し、OOB/INBコントラクトおよびUDP 161のフィルタを確認します。動作確認：APICで、iptables -S | grep 161を実行します。fp-137 （または同等のOOBコントラクト）チェーンの下にUDP 161のACCEPTルールが表示されない場合は、UDP 161のコントラクトフィルタが存在しないか、展開されていません。 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 subnet -A fp-13777.18.0.0/16 -p udp -m -m udp -m -m udp -m udp -dudp –dport 16 1 -j ACCEPT <– INB管理サブネットからSNMPを許可するこれらのルールが存在しない場合は、UDP 161のフィルタエントリを管理契約の対象に追加し、再検証します。根本的な原因：管理コントラクトが見つからないか、正しく構成されていません。ACI 5.xでは、APICノードは管理契約を厳密に適用します。明示的な許可が存在しない限り、SNMPパケッ

トはドロップされます。解決方法：［テナント］>［管理］>［セキュリティポリシー］>［アウトオブバンド契約］に移動します。OOBコントラクトを展開し、Subjectを選択して、UDPポート161のフィルタを確認/追加します。NMSがINB管理を介してAPICに到達する場合は、インバンド契約に対して手順を繰り返します。保存後、APICでiptables -S | grep 161を使用して確認します。　シナリオ7:SNMP iptablesルールがないか、正しくない問題：show snmp summaryを実行すると、SNMPポリシーは適用されているがiptables -S | grep snmpを実行してもルールが返されないか、NMSクライアントのIPがルールに含まれていない。動作確認：snmpdがpidof snmpdで実行されていることを確認します。snmpdが実行されているが、iptablesにSNMPルールがない場合、クライアントのグループポリシーが展開される前にプロセスが開始されています。再起動回数が250回未満の場合、snmpdを再起動してルールの再プログラミングを強制します。 leaf101# pidof snmpd 5881leaf101# show system internal sysmgr service name snmpdService "snmpd" ("snmpd", 127):UUID = 0x1A, PID = 5881, SAP = 1545State: SRV_STATE_HANDSHAKED （入力済8月25日1日9:23:50 2025）。再起動カウント：3最終再起動時刻：8月25日（月） 19:23:48 2025以前のPID: 32080最終終了理由： SYSMGR_DEATH_REASON_FAILURE_SIGNALTag = N/APlugin ID: 0 leaf101# kill -9 5881 ACIプロセスマネージャは自動的にsnmpdを再起動します。再起動後、次のように確認します。 leaf101# iptables -S | grep -i snmp snmp_rulesチェーンとVRFごとのクライアントACCEPTルールが表示されます。根本原因：クライアントグループポリシーがノードに完全に展開される前にsnmpdプロセスが再起動または開始され、SNMPアクセスルールなしでiptablesが残されました。

詳細なトラブルシューティングのためのログファイル

上記の検証手順で問題が解決しない場合、リーフ、スパイン、およびAPICノードの次のログファイルに、SNMP関連の診断情報が記録されます。

<#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/*

これらのログには、snmpd再起動イベント、ポリシー展開イベント、およびshow snmp summaryでは確認できないコミュニティ/クライアント設定エラーが含まれています。

参照資料

- Cisco APICシステム管理設定ガイド、リリース5.x:SNMPの管理
- Cisco ACI MIBクイックリファレンスガイド

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