REST API を使用したトラブルシューティング

収集およびテクニカル サポート情報のエクスポート

ファイルのエクスポートについて

管理者は、APIC 内で、コア ファイルとデバッグ データを処理するために、統計情報、テクニカル サポートの収集、障害およびイベントをファブリック (APIC およびスイッチ) から外部ホストにエクスポートするようエクスポート ポリシーを設定できます。エクスポートは XML、JSON、Web ソケット、Secure Copy Protocol (SCP) 、HTTP などのさまざまな形式にできます。ストリーミング、定期的、またはオンデマンドの各形式でエクスポートを登録できます。

管理者は、転送プロトコル、圧縮アルゴリズム、転送の頻度などポリシーの詳細を設定できます。ポリシーは、AAA を使用して認証されたユーザによって設定できます。実際の転送のセキュリティ メカニズムは、ユーザ名とパスワードに基づいています。内部的に、ポリシー要素はデータのトリガーを処理します。

REST API を使用したオンデマンドの techsupport ファイルの送信

手順

ステップ 1 REST API を使用して次の例のような XML を POST 送信し、テクニカル サポート ファイルのリモート宛先を設定します。

例：

```
<fileRemotePath userName="" remotePort="22" remotePath="" protocol="sftp" name="ToSupport"
 host="192.168.200.2"
dn="uni/fabric/path-ToSupport" descr="">

<fileRsARemoteHostToEpg tDn="uni/tn-mgmt/mgmtp-default/oob-default"/>

</fileRemotePath>
```

ステップ 2 REST API を使用して次のような XML を POST 送信し、オンデマンドのテクニカル サポート ファイルを生成します。

例：

```
<dbgexpTechSupOnD upgradeLogs="no" startTime="unspecified" name="Tech_Support_9-20-16"
exportToController="no"
endTime="unspecified" dn="uni/fabric/tsod-Tech_Support_9-20-16" descr="" compression="gzip"

category="forwarding" adminSt="untriggered">

<dbgexpRsExportDest tDn="uni/fabric/path-ToSupport"/>

<dbgexpRsTsSrc tDn="topology/pod-1/node-102/sys"/>

<dbgexpRsTsSrc tDn="topology/pod-1/node-103/sys"/>

<dbgexpRsTsSrc tDn="topology/pod-1/node-101/sys"/>

<dbgexpRsData tDn="uni/fabric/tscont"/>

</dbgexpTechSupOnD>
```

アトミックカウンタを使用したトラブルシューティング

アトミック カウンタ

アトミックカウンタは、ファブリック内のエンドポイント、EPG、アプリケーション間の接続をトラブルシューティングするのに便利です。アプリケーションのレポート時、ユーザは処理の遅さを経験する場合があります。また、2 つのエンドポイント間のトラフィック損失をモニタする場合、アトミック カウンタが必要な場合があります。アトミック カウンタが提供する機能の 1 つに、トラブル チケットを予防的モニタリング モードに配置する機能があります。

たとえば、問題が断続的に発生していても、オペレータが積極的にチケットの作業にあたっている時に必ずしも発生するとは限らない場合があります。

アトミック カウンタは、ファブリック内のパケット損失を検知し、接続の問題の発生源をすばやく分離できるようにするのに役立ちます。アトミック カウンタには、NTP をファブリックで有効にする必要があります。

リーフ間（TEP 間）アトミック カウンタは次を提供できます。

- ドロップ、アドミットおよび超過パケットのカウント
- 最後の 30 秒などの短期間のデータ収集、5 分、15 分、またはそれ以上の長期間のデータ収集
- スパイン トラフィックごとの詳細（TEP、リーフ、または VPC の数が 64 未満の場合に使用可能）
- 継続的なモニタリング

リーフ間（TEP間）アトミック カウンタは累積であり、クリアできません。ただし、30 秒のアトミック カウンタは 30 秒間隔でリセットされるため、断続的な問題や、再発する問題の分離に使用できます。

テナントのアトミック カウンタは次を提供できます:

- ドロップ、承認および超過パケットを含む、ファブリック全体のトラフィックのアプリケーション固有カウンタ
- モードは次を含みます。
- エンドポイント間 MAC アドレスまたはエンドポイント間 IP アドレス注：単一のターゲット エンドポイントに、それに関連付けられた複数の IP アドレスがある場合があります。
- オプションのドリル ダウン付きの EPG ツー EPG
- EPG ツー エンドポイント
- EPG ツー *（任意）
- エンドポイント ツー外部 IP アドレス

（注） アトミック カウンタは、2つのエンドポイント間のパケット量を追跡し、これを測定値として使用します。ハードウェア レベルのドロップやエラー カウンタは考慮しません。

送信元が送信した数よりも宛先で受け取ったパケットの数が少ない場合に、ドロップされたパケットが計算されます。

送信元が送信した数よりも宛先で受け取ったパケットの数が多い場合に、超過パケットが計算されます。

アトミック カウンタの有効化

アトミック カウンタを使用することを有効にしてファブリックのドロップおよびルーティングの間違いを検出し、アプリケーションの接続問題の分離を素早くデバッグできるようにするに

は、次のタイプのいずれかで、1 個以上のテナント アトミック カウンタ ポリシーを作成します。

- EP_to_EP：エンドポイントからエンドポイント（**dbgacEpToEp**）
- EP_to_EPG：エンドポイントからエンドポイント グループ（**dbgacEpToEpg**）
- EP_to_Ext：エンドポイントから外部 IP アドレス（**dbgacEpToExt**）
- EPG_to_EP：エンドポイント グループからエンドポイント（**dbgacEpgToEp**）
- EPG_to_EPG：エンドポイント グループからエンドポイント グループ（**dbgacEpgToEpg**）
- EPG_to_IP：エンドポイント グループから IP アドレス（**dbgacEpgToIp**）
- Ext_to_EP：外部 IP アドレスからエンドポイント（**dbgacExtToEp**）
- IP_to_EPG：IP アドレスからエンドポイント グループ（**dbgacIpToEpg**）
- Any_to_EP：エニーからエンドポイント（**dbgacAnyToEp**）
- EP_to_Any：エンドポイントからエニー（**dbgacEpToAny**）

手順

ステップ 1 REST API を使用して EP_to_EP ポリシーを作成するには、次の例のように XML を使用します。

例：

```
<dbgacEpToEp name="EP_to_EP_Policy" ownerTag="" ownerKey=""
dn="uni/tn-Tenant64/acEpToEp-EP_to_EP_Policy" descr="" adminSt="enabled">
<dbgacFilter name="EP_to_EP_Filter" ownerTag="" ownerKey="" descr=""
srcPort="https" prot="tcp" dstPort="https"/>
</dbgacEpToEp>
```

ステップ 2 REST API を使用して EP_to_EPG ポリシーを作成するには、次の例のように XML を使用します。

例：

```
<dbgacEpToEpg name="EP_to_EPG_Pol" ownerTag="" ownerKey=""
dn="uni/tn-Tenant64/epToEpg-EP_to_EPG_Pol" descr="" adminSt="enabled">
<dbgacFilter name="EP_to_EPG_Filter" ownerTag="" ownerKey="" descr=""
srcPort="http" prot="tcp" dstPort="http"/>
<dbgacRsToAbsEpg tDn="uni/tn-Tenant64/ap-VRF64_app_prof/epg-EPG64"/>
</dbgacEpToEpg>
```

ファブリック レイテンシについて

ファブリック レイテンシは、ファブリック内の送信元から宛先へのパケットの移動にかかる時間をモニタするためのトラブルシューティング ツールになります。エンドポイント、エンドポ

イント グループ、外部インターフェイス、IP アドレスの組み合わせ間のレイテンシを測定するために使用できます。レイテンシは、入力リーフ スイッチの **[到着]** 時間から出力リーフ スイッチの **[出発]** 時間までで測定されます。ファブリック レイテンシの必要条件は、一定の時間ですべてのノードを同期するものとします。Precision Time Protocol（PTP）はこのために使用され、サブマイクロ秒の正確性のため、ミリ秒単位の精度のみを持つNTP と比較されます。NTP はマイクロ秒の順に ACI ファブリック内のパケット フライト時間を測定するには十分ではありません。そのため、レイテンシ機能には、PTP を使用して同期されるファブリックのすべてのノードが必要です。

レイテンシの測定 2 つのタイプがあります：

- 進行中の TEP-to-TEP レイテンシ
- オンデマンド テナント レイテンシ

進行中のレイテンシまたは Leaf-to-leaf (TEP to TEP) レイテンシは、リーフ スイッチでトンネルエンドポイント間でのレイテンシを測定するために使用されます。平均および最大レイテンシ、標準偏差、および宛先リーフ スイッチで計算されるパケット数を提供します。累積レイテンシの値と同様に、過去 30 秒の収集済みレイテンシ データが提供されます。TEP-to-TEP レイテンシ測定は、ファブリックで PTP がオンになるとすぐに有効になります。

テナント レイテンシ測定は、個々のアプリケーションのレベルで問題のトラブルシューティングをするように設定できます。これらは、レイテンシTCAM でプログラムされた特定のフロー ルールに一致する IP フローに対して有効にできます。フロー ルール セマンティクスは、現在の原子カウンタ フロー ルールに似ています。

（注） レイテンシ測定が特定の IP フローに設定され、このフローのトンネルに対してレイテンシ測定が同時に行われる場合、このフローのレイテンシには考慮されません。

原子カウンタだけでなく、レイテンシ測定では次のフロー ルールがサポートされています。

- EP から EP へのレイテンシの測定
- EP から EPG へのレイテンシの測定
- EP から 外部 IP へのレイテンシの測定
- EPG から EP へのレイテンシの測定
- EPG から EPG へのレイテンシの測定
- EPG から 外部 IP へのレイテンシの測定
- 外部 IP から EP へのレイテンシの測定
- 外部 IP から EPG へのレイテンシの測定
- いずれかから EP へのレイテンシの測定
- 外部 IP から外部 IP へのレイテンシの測定

- EP からいずれかのレイテンシの測定

（注） 原子カウンタとレイテンシ測定の両方が、同じ IP フロー ルールで独自に有効または無効にできます。

2つのモードでレイテンシデータを測定することができます。平均およびヒストグラム。モードはテナント レイテンシ ポリシーの各フロー ルールと同様に、進行中のレイテンシに個別指定できます。

平均モード

平均モードでは、次の測定を有効にできます。

- 過去 30 秒間の平均レイテンシ
- 過去 30 秒間の標準偏差
- 過去 30 秒のパケット数
- 累積平均レイテンシ
- 累積最大レイテンシ
- 累積パケット数

（注） 平均モードのレイテンシ測定は、外部測定機器と比較して、0.1 マイクロ秒低倍数で若干異なる可能性があります。

ヒストグラム モード

ヒストグラムモードでは、さまざまなレイテンシ間隔でパケット数の配信の可視化を有効にします。16 のヒストグラム バケットがあり、各バケットには測定間隔が設定されています。バケット 0 の測定間隔が 0〜5 マイクロ秒であり、5〜10 マイクロ秒間のバケット 1 は最後のバケットの 80 マイクロ秒で終了します。これらのバケットのそれぞれには 64 ビット カウンタが含まれ、バケットの設定されているレイテンシ間隔内で落ちたレイテンシを持つパケットを測定します。

ヒストグラム グラフは、レイテンシの傾向を理解するには便利ですが、正確なパケットの数が反映されない可能性があります。実際のパケット数を測定するには、原子カウンタを使用できます。

サポートされている TEP から TEP までのレイテンシ エントリの最大数は 384 です。EX ベース TOR では、平均モードでほとんどは 256 フロー、ヒストグラム モードでは 64 フローにできます。FX ベース TORS では、平均モードでほとんどは 640 フロー、ヒストグラム モードでは 320 フローにできます。

PTP について

Precision Time Protocol （PTP）は、IEEE 1588 で定義された、ネットワークに分散したノードの時刻同期プロトコルです。PTP を使用すると、分散したクロックを、イーサネット ネットワークを経由して、1 マイクロ秒以下の精度で同期させることができます。PTP の精度は ACI ファブリック スパインおよびリーフでサポートされているハードウェアから提供されます。これにより、プロトコルが正確にネットワーク全体のメッセージ遅延と変動を補正できます。

PTPは、システムのリアルタイムPTPクロックが相互に同期する方法を指定する分散プロトコルです。これらのクロックは、グランドマスタークロック（階層の最上部にあるクロック）を持つマスター/スレーブ同期階層に編成され、システム全体の時間基準を決定します。同期は、タイミング情報を使用して階層のマスターの時刻にクロックを調整するメンバーと、PTPタイミングメッセージを交換することによって実現されます。PTPは、PTPドメインと呼ばれる論理範囲内で動作します。

PTPプロセスは、マスター/スレーブ階層の確立とクロックの同期の2つのフェーズで構成されます。PTPドメイン内では、オーディナリ クロックまたは境界クロックの各ポートが、次のプロセスに従ってステートを決定します。

- 受信したすべての（マスタ状態のポートによって発行された）アナウンス メッセージの内容を検査します。
- 外部マスタのデータ セット（アナウンス メッセージ内）とローカル クロックで、優先順位、クロック クラス、精度などを比較します。
- 自身の状態がマスタまたはスレーブのいずれであるかを決定します。

マスター/スレーブ階層が確立されると、クロックは次のように同期されます。

- マスターはスレーブに同期メッセージを送信し、送信された時刻を記録します。
- スレーブは同期メッセージを受信し、受信した時刻を記録します。すべての同期メッセージには、フォローアップ メッセージがあります。したがって、同期メッセージの数は、フォローアップ メッセージの数と同じである必要があります。
- スレーブはマスターに遅延要求メッセージを送信し、送信された時刻を記録します。
- マスターは遅延要求メッセージを受信し、受信した時刻を記録します。
- マスターはスレーブに遅延応答メッセージを送信します。遅延要求メッセージの数は、遅延応答メッセージの数と同じある必要があります。
- スレーブは、これらのタイムスタンプを使用して、クロックをマスターの時刻に調整します。

ACIファブリックでは、PTP機能がAPIC内でグローバルに有効になっている場合、ソフトウェアがサポートされているスパインおよびリーフの特定のインターフェイスで自動的にPTPを有効にします。この自動設定は、サポートされているすべてのノードで最適にPTPが有効になっていることを保証します。外部グランドマスタ クロックがない場合、スパイン スイッチのいずれかがグランドマスタとして選択されます。PTPスレーブとして動作するように、マスタ スパインには他のスパインおよびリーフ スイッチと比較して異なる PTP 優先順位が与えられま

す。この方法により、ファブリック内のすべてのリーフ スイッチがマスタ スパインの PTP クロックと同期します。

外部グランドマスタ クロックがスパインに接続している場合、スパインは外部GMに同期し、リーフ ノードに対してマスタとして動作します。

PTP デフォルト設定

次の表に、PTP パラメータのデフォルト設定を示します。

パラメータ	デフォルト
PTP デバイス タイプ	境界クロック
PTP クロック タイプ	ツーステップ クロック
PTP ドメイン	0
クロックをアドバタイズする場合、PTP プライオリティ 1 値	255
クロックをアドバタイズする場合、PTP プライオリティ 2 値	255
PTP アナウンス間隔	1 ログ秒
PTP アナウンス タイムアウト	3 アナウンス間隔
PTP 遅延要求間隔	0 ログ秒
PTP 同期間隔	-2 ログ秒
PTP VLAN	1

（注） PTPは境界クロック モードのみで動作します。シスコでは、スイッチに接続された、同期を必要とするクロックが含まれるサーバを使用して、グランドマスター クロック（10 MHz）アップストリームを配置することを推奨します。

PTP の検証

コマンド	目的
show ptp brief	PTP のステータスを表示します。
show ptp clock	ローカル クロックのプロパティ（クロック ID など）を表示します。

コマンド	目的
show ptp clock foreign-masters record interface ethernet *slot/port*	PTP プロセスが認識している外部マスターの状態を表示します。外部マスターごとに、出力に、クロック ID、基本的なクロック プロパティ、およびクロックがグランドマスターとして使用されているかどうかが表示されます。
show ptp corrections	最後の数個の PTP 修正を表示します。
show ptp counters [**all** |**interface Ethernet** *slot/port*]	すべてのインターフェイスまたは指定されたインターフェイスの PTP パケット カウンタを表示します。
show ptp parent	PTP の親のプロパティを表示します。

REST API でアトミック カウンタを使用したトラブルシューティング

手順

ステップ 1 ファブリック内で展開されているエンドポイントツーエンドポイント アトミック カウンタのリストを取得して、ドロップしたパケットの統計情報やパケット カウントなどの詳細を関連付けるには、次のように XML で **dbgEpToEpTsIt** を使用します。

例：

```
https://apic-ip-address/api/node/class/dbgEpToEpRslt.xml
```

ステップ 2 外部 IP からエンドポイント アトミック カウンタと関連付けられているリストを取得するには、次の例のように XML で **dbgacExtToEp** クラスを使用します。

例：

```
https://apic-ip-address/api/node/class/dbgExtToEpRslt.xml
```

REST API を使用した遅延および PTP の設定

フロー ポリシー パラメータを設定するには、Cisco APIC トラブルシューティング ガイドで原子カウンタの設定の同じ手順に従います: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/troubleshooting/b_APIC_Troubleshooting/b_APIC_Troubleshooting_chapter_0110.html#id_40942 。

手順

ステップ 1 PTP モードを有効にするには：

例：

```
/api/node/mo/uni/fabric/ptpmode.xml
<latencyPtpMode state=”enabled”>
```

ステップ 2 EP に EP ポリシーを設定します。

例：

```
<dbgacEpToEp name="EP_to_EP_Policy" adminSt="enabled" usage=“latency-stats” latencyCollect
 = “histogram”>
</dbgacEpToEp>
```

ステップ 3 原子カウンタと遅延 (平均モード) の両方を有効にするのには、XML が次のとおり

例：

```
<dbgacEpToEp name="EP_to_EP_Policy" adminSt="enabled" usage=“latency-stats|atomic-counter”
 latencyCollect = “average”>
</dbgacEpToEp>
```

ステップ 4 コレクションのタイプを変更する **継続中モード** ヒストグラム平均から。

例：

```
<latencyOngoingMode userMode=”histogram”>
```

Troubleshooting Using Faults

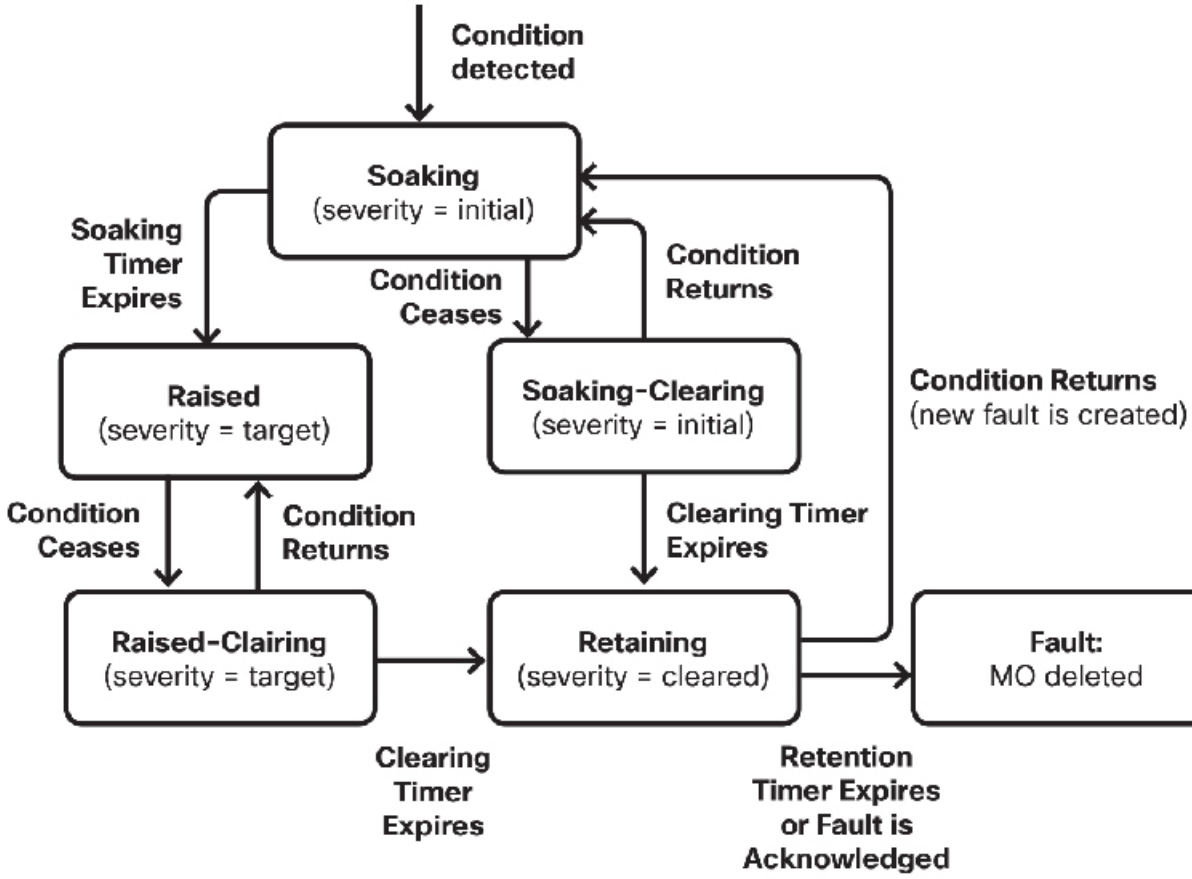
APIC 障害の理解

From a management point of view we look at the Application Policy Infrastructure Controller (APIC) from two perspectives:

- ポリシーコントローラ：すべてのファブリック構成が作成され、管理され、適用されている場合。管理状態または構成状態の、総合的で最新のランタイム表示を維持します。
- Telemetry device - All devices (Fabric Switches, Virtual Switches, integrated Layer 4 to Layer 7 devices) in an Cisco Application Centric Infrastructure (ACI) fabric report faults, events and statistics to the APIC.

Faults, events, and statistics in the ACI fabric are represented as a collection of Managed Objects (MOs) within the overall ACI Object Model/Management Information Tree (MIT). All objects within ACI can be queried, including faults. このモデルでは、フォールトは、可変でステートフルな永続的 MO として表されます。

図 *1 :* フォールトのライフサイクル



コンポーネントのフォールトやアラームなど、特定の状態が発生した場合は、システムはフォールトMOを、そのMOの子オブジェクトとして作成し、基本的にそのフォールトに関連付けます。フォールトオブジェクト クラスでは、親オブジェクト クラスのフォールトルールによりフォールト状態が定義されます。フォールト MO は、親、DN、RN などを持つ通常の MO としてMITに表示されます。フォールト「コード」は、**FXXX**形式の英数字文字列です。フォールトコードの詳細については、『*Cisco APIC Faults, Events, and System Messages Management Guide*』を参照してください。

REST API で障害を使用したトラブルシューティング

MO はプロパティ フィルタ、改ページなどを使用し、クラスや DN 別に照会できます。

ほとんどの場合、フォールトMOは特定の状態が検出されたときに、システムによって自動的に作成、エスカレーション、エスカレーション解除、および削除されます。MOには、所定のコードを持つフォールトが最大で1つある可能性があります。対応するフォールトMOがアクティブな間にある条件が複数回検出された場合は、そのMOに新たなインスタンスが作成され

ることはありません。For example, if the **same condition** is detected multiple times for the **same affected object**, only **one fault** is raised while a counter for the recurrence of that fault will be incremented.

フォールト状態がクリアされるまでは、フォールトMOはシステムに残ります。障害を排除するには、構成によって、またはファブリックの実行時状態を変更することによってのいずれかで、そのフォールトを発生させている状態を解消する必要があります。フォールトが解消された状態またはそのままの状態にあり、その場合にそのフォールトをユーザが確認することで削除できる場合は例外です。

重大度は、サービスを提供するシステムまたはコンポーネントの能力の状態への予測される影響力を示します。

値は次のとおりです。

- Warning（ほとんど影響しない）
- Minor
- Major
- Critical（システムまたはコンポーネントが完全に使用できなくなる）

フォールトMOの作成が、以下のような内部プロセスによって引き起こされる可能性があります。

- 有限状態マシン（FSM）の遷移、またはコンポーネントフォールトの検出
- さまざまな障害ポリシーによって指定された状態（一部はユーザーによる設定が可能）

（注） ヘルススコア、データ トラフィック、温度などの統計的な測定値にフォールトしきい値を設定できます。

手順

ステップ 1 「3tierapp」という名前のテナントのヘルス スコアを取得するには、次のようにファブリックに REST クエリを送信します。

例：

```
https://apic-ip-address/api/node/mo/uni/tn-3tierapp.xml?query-target=self&rsp-subtreeinclude=
health
```

ステップ 2 「3tierapp」という名前のテナントの統計情報を取得するには、次のようにファブリックにREST クエリを送信します。

例：

```
https://apic-ip-address/api/node/mo/uni/tn-3tierapp.xml?query-target=self&rsp-subtreeinclude=
stats
```

ステップ 3 リーフ ノードの障害を取得するには、次のようにファブリックに、REST クエリを送信します。

例：

```
https://apic-ip-address/api/node/mo/topology/pod-1/node-103.xml?query-target=self&rspsubtree-include=faults
```

統計情報

REST API を使用した統計情報モニタリング ポリシーの設定する

ファブリックのモニタリングおよびトラブルシューティングの統計情報を使用するには、統計情報収集ポリシーおよび統計情報エクスポート ポリシーを設定して、APIC の多くのオブジェクトをモニタできます。

手順

ステップ 1 REST API を使用して統計情報収集ポリシーを作成するには、次の例のように XML で POST 要求を送信します。

例：

```
<monEPGPol name="MonPol1" dn="uni/tn-tenant64/monepg-MonPol1">
     <monEPGTarget name="" descr="" scope="eventSevAsnP"/>
     <monEPGTarget name="" descr="" scope="faultSevAsnP"/>
     <monEPGTarget name="" descr="" scope="fvBD">
     <statsHierColl name="" descr="" histRet="inherited" granularity="5min"
adminState="inherited"/>
</monEPGTarget>
<monEPGTarget name="" descr="" scope="syslogRsDestGroup"/>
<monEPGTarget name="" descr="" scope="syslogSrc"/>
<monEPGTarget name="" descr="" scope="fvCtx"/>
<statsHierColl name="" descr="" histRet="none" granularity="1w" adminState="enabled"/>
<statsHierColl name="" descr="" histRet="none" granularity="1qtr" adminState="enabled"/>
<statsHierColl name="" descr="" histRet="1w" granularity="1h" adminState="enabled"/>
<statsHierColl name="" descr="" histRet="1d" granularity="15min" adminState="enabled"/>
<statsHierColl name="" descr="" histRet="none" granularity="1year" adminState="enabled"/>
<statsHierColl name="" descr="" histRet="none" granularity="1mo" adminState="enabled"/>
<statsHierColl name="" descr="" histRet="1h" granularity="5min" adminState="enabled"/>
<statsHierColl name="" descr="" histRet="10d" granularity="1d" adminState="enabled"/>
<syslogSrc name="VRF64_SyslogSource" descr="" minSev="warnings" incl="faults">
<syslogRsDestGroup tDn="uni/fabric/slgroup-tenant64_SyslogDest"/>
</syslogSrc>
</monEPGPol>
```

ステップ 2 統計情報エクスポート ポリシーを設定するには、次の例のように XML で POST 要求を送信します（JSON または XML 形式のいずれかを使用できます）。

例：

```
<statsExportP
          name="" descr="" frequency="stream" format="xml" compression="gzip">
     <statsDestP name="tenant64_statsExportDest" descr="" userName="" remotePort="0"
          remotePath="192.168.100.20" protocol="sftp" host="192.168.100.1">
        <fileRsARemoteHostToEpg tDn="uni/tn-mgmt/mgmtp-default/oob-default"/>
     </statsDestP>
</statsExportP>
```

切断されたリーフの復旧

切断されたリーフの復旧

リーフにプッシュされた設定により、リーフのすべてのファブリック インターフェイス（リーフをスパインに接続するインターフェイス）が無効になっている場合、リーフへの接続は完全に失われ、リーフはファブリックで無効になります。リーフに設定をプッシュしようとしても、接続が失われているため実行されません。この章では、切断したリーフの回復方法について説明します。

REST API を使用して切断されたリーフを回復する

切断されたリーフを回復するには、次の手順を使用して少なくとも 1 つのファブリック インターフェイスを有効する必要があります。残りのインターフェイスは、GUI、REST API または CLI を使用して有効にできます。

最初のインターフェイスを有効にするには、REST API を使用してポリシーを POST し、POST したポリシーを削除してファブリック ポートを無効にします。以下のように、リーフにポリシーを POST して、無効のポートを有効にすることができます。

（注） 次の例では、1/49 はスパインに接続しているリーフ ポートの 1 つであると想定しています。

手順

ステップ 1 APIC からブラックリスト ポリシーをクリアします（REST API を使用）。

例：

```
$APIC_Address/api/policymgr/mo/.xml
<polUni>
    <fabricInst>
        <fabricOOServicePol>
                        <fabricRsOosPath
tDn="topology/pod-1/paths-$LEAF_Id/pathep-[eth1/49]" lc="blacklist" status ="deleted"
/>
        </fabricOOServicePol>
```

```
    </fabricInst>
</polUni>
```

ステップ 2 **l1EthIfSetInServiceLTask** を使用して必要なインターフェイスを起動するために、ローカル タスクをノード自体に POST します。

例：

```
$LEAF_Address/api/node/mo/topology/pod-1/node-$LEAF_Id/sys/action.xml
<actionLSubj oDn="sys/phys-[eth1/49]">
<l1EthIfSetInServiceLTask adminSt='start'/>
</actionLSubj>
```

Permit と契約し、ロギングを拒否契約および Taboo のトラブルシューティング

契約、タブー契約は、REST API を使用してフィルタの確認

このトピックでは、契約、タブー契約は、およびフィルタを確認する REST API XML を提供します。

手順

ステップ 1 プロバイダーの EPG または XML で、次の例などの外部ネットワークには、契約を確認します。

例：

```
QUERY https://apic-ip-address/api/node/class/fvRsProv.xml
```

ステップ 2 消費者の次の例など、EPG と XML の契約を確認します。

例：

```
QUERY https://apic-ip-address/api/node/class/fvRsCons.xml
```

ステップ 3 次の例など XML を使用してエクスポートされた契約を確認します。

例：

```
QUERY https://apic-ip-address/api/node/class/vzCPif.xml
```

ステップ 4 次の例などと XML の VRF の契約を確認します。

例：

```
QUERY https://apic-ip-address/api/node/class/vzBrCP.xml
```

ステップ 5 次の例などと XML タブー契約を確認します。

例：

```
QUERY https://apic-ip-address/api/node/class/vzTaboo.xml
```

EPG のタブー契約は、Epg の契約と同じクエリを使用します。

ステップ 6 次の例など XML を使用してフィルタを確認します。

例：

```
QUERY https://apic-ip-address/api/node/class/vzFilter.xml
```

REST API を使用した ACL 許可および拒否ログ

次の例は、REST API を使用して、トラフィック フローのレイヤ 2 拒否ログ データを表示する方法を示しています。次の MO を使用してクエリを送信することができます。

- acllogDropL2Flow
- acllogPermitL2Flow
- acllogDropL3Flow
- acllogPermitL3Flow
- acllogDropL2Pkt
- acllogPermitL2Pkt
- acllogDropL3Pkt
- acllogPermitL3Pkt

始める前に

ACL 契約許可および拒否ログのデータを表示する前に、許可または拒否ロギングを有効にする必要があります。

手順

レイヤ 3 ドロップ ログ データを表示するには、REST API を使用して次のクエリを送信します。

```
GET https://apic-ip-address/api/class/acllogDropL3Flow
```

例：

次の例では、サンプル出力をいくつか示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<imdata totalCount="2">
    <acllogPermitL3Flow childAction=""
dn="topology/pod-1/node-101/ndbgs/acllog/tn-common/ctx-inb

/permitl3flow-spctag-333-dpctag-444-sepgname-unknown-depgname-unknown-sip-[100:c000:a00:700:b00:0:f00:0]
```

```
-dip-[19.0.2.10]-proto-udp-sport-17459-dport-8721-smac-00:00:15:00:00:28-dmac-00:00:12:00:00:25-sintf-

        [port-channel5]-vrfencap-VXLAN: 2097153" dstEpgName="unknown" dstIp="19.0.2.10"
 dstMacAddr="00:00:12:00:00:25"
        dstPcTag="444" dstPort="8721" lcOwn="local" modTs="never" monPolDn=""
protocol="udp" srcEpgName="unknown"
        srcIntf="port-channel5" srcIp="100:c000:a00:700:b00:0:f00:0"
srcMacAddr="00:00:15:00:00:28" srcPcTag="333"
        srcPort="17459" status="" vrfEncap="VXLAN: 2097153"/>
    <acllogPermitL3Flow childAction=""
dn="topology/pod-1/node-102/ndbgs/acllog/tn-common/ctx-inb

/permitl3flow-spctag-333-dpctag-444-sepgname-unknown-depgname-unknown-sip-[100:c000:a00:700:b00:0:f00:0]-dip-


[19.0.2.10]-proto-udp-sport-17459-dport-8721-smac-00:00:15:00:00:28-dmac-00:00:12:00:00:25-sintf-

        [port-channel5]-vrfencap-VXLAN: 2097153" dstEpgName="unknown" dstIp="19.0.2.10"
 dstMacAddr="00:00:12:00:00:25"
        dstPcTag="444" dstPort="8721" lcOwn="local" modTs="never" monPolDn=""
protocol="udp" srcEpgName="unknown"
        srcIntf="port-channel5" srcIp="100:c000:a00:700:b00:0:f00:0"
srcMacAddr="00:00:15:00:00:28" srcPcTag="333"
        srcPort="17459" status="" vrfEncap="VXLAN: 2097153"/>
</imdata>
```

Troubleshooting Using Digital Optical Monitoring Statistics

REST API を使うデジタル オプティカル モニタリングを使用したトラブルシューティング

DOM 統計情報を XML の REST API クエリを使用して表示するには、次の手順に従います。

始める前に

インターフェイスのDOM統計情報を表示するには、インターフェイスのデジタル オプティカル モニタリング（DOM）を事前に有効にしておく必要があります。

手順

次の例は、REST API クエリを使用して、物理インターフェイスについての DOM 統計情報（node-104 の eth1/25）を表示する方法を示しています。

```
GET
https://apic-ip-address/api/node/mo/topology/pod-1/node-104/sys/phys-[eth1/25]/phys/domstats.xml?
query-target=children&target-subtree-class=ethpmDOMRxPwrStats&subscription=yes
```

次の応答が返されます。

```
response : {
   "totalCount":"1",
      "subscriptionId":"72057611234705430",
          "imdata":[
{"ethpmDOMRxPwrStats":{
   "attributes":{
      "alert":"none",
      "childAction":"",
      "dn":"topology/pod-1/node-104/sys/phys[eth1/25]/phys/domstats/rxpower",
      "hiAlarm":"0.158490",
      "hiWarn":"0.079430",
      "loAlarm":"0.001050",
      "loWarn":"0.002630",
      "modTs":"never",
      "status":"",
       "value":"0.139170"}}}]}
```


ポートトラッキングを使用したトラブルシューティング

アップリンク障害検出のためのポート トラッキング ポリシー

アップリンク障害検出は、ファブリック アクセス グローバル ポート トラッキング ポリシーで有効化できます。ポート トラッキング ポリシーは、リーフ スイッチとスパイン スイッチ間のリンクの状態を監視します。有効なポート トラッキング ポリシーがトリガーされると、リーフ スイッチは、EPG によって導入されたスイッチ上のすべてのアクセス インターフェイスをダウンさせます。

（注） ポート トラッキングは、**[ファブリック]** > **[外部アクセス ポリシー]** > **[ポリシー]** > **[グローバル]** > **[ポート トラッキング]** の下にあります。

各リーフ スイッチには、各スパイン スイッチに最大6個のアップリンク接続があります。ポート トラッキング ポリシーは、ポリシーをトリガーするアップリンク接続の数と、指定のアップリンク数を超えた後にリーフ スイッチ アクセス ポートを復旧させる遅延タイマーを指定します。

ポート トラッキング ポリシーの動作の例を次に示します。

- 各リーフ スイッチからスパイン スイッチへのアクティブなアップリンク接続の数は、最大 6 つです。
- ポート トラッキング ポリシーは、ポリシーをトリガーする各リーフ スイッチのアクティブなアップリンク接続のしきい値を 2 に指定します。
- リーフ スイッチからスパイン スイッチへのアクティブなアップリンク接続数が 2 まで減少すると、ポート トラッキング ポリシーがトリガーされます。

- 各リーフ スイッチはそのアップリンク接続を監視し、ポリシーで指定されたしきい値に従ってポート トラッキング ポリシーをトリガーします。
- アップリンク接続が復旧すると、リーフ スイッチは遅延タイマーの時間が満了するのを待ち、その後、そのアクセス ポートを復旧させます。これにより、ファブリックには、リーフ スイッチ アクセス ポートでトラフィックが再開する前に再コンバージェンスできる時間が確保されます。大きなファブリックでは、遅延タイマーの時間を長めに設定することが必要な場合があります。

（注） このポリシーの設定には注意が必要です。ポート トラッキング設定において、ポート トラッキングをトリガーするアクティブなスパイン リンク数を過剰に大きく設定すると、すべてのリーフ スイッチ アクセス ポートがダウンします。

REST API を使用したポート トラッキング

始める前に

この手順では、REST API を使用したポート トラッキング機能の使用方法について説明します。

手順

ステップ 1 REST API を使用して、次のようにポート トラッキング機能をオンにします（**admin state**：**on**）。

```
<polUni>
<infraInfra dn="uni/infra">
<infraPortTrackPol name="default" delay="5" minlinks="4" adminSt="on">

</infraPortTrackPol>
</infraInfra>
</polUni>
```

ステップ 2 REST API を使用して、次のようにポート トラッキング機能をオフにします（**admin state**：**off**）。

```
<polUni>
<infraInfra dn="uni/infra">
<infraPortTrackPol name="default" delay="5" minlinks="4" adminSt="off">

</infraPortTrackPol>
</infraInfra>
</polUni>
```

不要な _ui_ オブジェクトの削除

REST API を使用した不要な _ui_ オブジェクトの削除

Cisco APIC GUI を使用する前に Cisco NX OS スタイル CLI で変更を行い、名前の先頭に `_ui_` が付加されたオブジェクトが表示された場合は、API に対して次を含む REST API 要求を実行することでこれらのオブジェクトを削除できます。

- クラス名（例：**infraAccPortGrp**）
- Dn 属性（例：**dn="uni/infra/funcprof/accportgrp-__ui_l101_eth1--31"**
- **status="deleted"** に設定したステータス属性

次の手順で API に POST を実行します。

手順

ステップ 1 削除するオブジェクトへの書き込みアクセス権を持つユーザ アカウントにログインします。

ステップ 2 API に次の例のような POST を送信します。

```
POST https://192.168.20.123/api/mo/uni.xml
Payload:<infraAccPortGrp dn="uni/infra/funcprof/accportgrp-__ui_l101_eth1--31"
status="deleted"/>
```

Troubleshooting Using Contract Permit and Deny Logs

ACL 契約の許可および拒否ログについて

契約ルールのトラフィック フローをログ記録および監視するには、契約の許可ルールのため送信されることが許可されたパケットまたはフローのログを有効化および表示するか、契約の許可ルールのためドロップされたフローのログを有効化および表示できます。

- 禁止契約拒否ルール
- 契約の件名でアクションを拒否する
- 契約または件名の例外
- ACL ファブリックの ACL 契約許可および拒否のログは、EX または FX で終わる名前の Nexus 9000 シリーズ スイッチと、それ以降のすべてのモデルでのみサポートされています。たとえば、N9K-C93180LC-EX や N9K-C9336C-FX のように指定してください。

- 管理契約のフィルタでログ directive を使用することはサポートされていません。ログ directive を設定すると、ゾーン分割ルールの展開エラーが発生します。

標準および禁止契約と件名についての詳細は、『*CiscoApplication Centric Infrastructure Fundamentals*』および『*Cisco APIC Basic Configuration Guide*』を参照してください。

ACL 許可および拒否ログ出力に含まれる EPG データ

Cisco APIC、リリース 3.2(1) まで、ACL 許可および拒否ログでは、記録されている契約に関連付けられた EPG を識別していませんでした。リリース 3.2(1) では、送信元 EPG と 送信先 EPG が ACI 許可および拒否ログの出力に追加されます。ACL 許可および拒否ログには、次の制限を持つ関連 EPG を含めます。

- ネットワーク内の EPG の配置によっては、ログの EPG データを使用できない場合があります。
- 設定の変更が発生するとき、ログデータが期限切れになっている可能性があります。安定した状態では、ログ データは正確です。

ログが次にフォーカスされているとき、許可および拒否ログの EPG データは最も正確になります。

- 入力 TOR で入力ポリシーがインストールされており、出力 TOR で出力ポリシーがインストールされている場合の EPG から EPG へのフロー。
- 境界リーフ TOR で 1 個のポリシーが適用され、非 BL TOR で他のポリシーが適用されている場合の EPG から L3Out へのフロー。

ログ出力の EPG は、共有サービス（共有 L3Outs を含む）で使用される uSeg Epg または Epg ではサポートされていません。

REST API を使用した ACL 契約許可ロギングの有効化

次の例は、REST API を使用して許可および拒否ロギングを有効にする方法を示しています。この例では、ACL の許可を設定し、件名 Permit 設定し、設定されたアクションを拒否するには、契約のロギングを拒否します。

手順

この設定では、次の例のように XML で post を送信します。

例：

```
<vzBrCP dn="uni/tn-Tenant64/brc-C64" name="C64" scope="context">
    <vzSubj consMatchT="AtleastOne" name="HTTPSsbj" provMatchT="AtleastOne"
revFltPorts="yes" rn="subj-HTTPSsbj">
         <vzRsSubjFiltAtt action="permit" directives="log" forceResolve="yes"
priorityOverride="default"
rn="rssubjFiltAtt-PerHTTPS" tDn="uni/tn-Tenant64/flt-PerHTTPS" tRn="flt-PerHTTPS"
tnVzFilterName="PerHTTPS"/>
```

```
    </vzSubj>
    <vzSubj consMatchT="AtleastOne" name="httpSbj" provMatchT="AtleastOne"
revFltPorts="yes" rn="subj-httpSbj">
        <vzRsSubjFiltAtt action="deny" directives="log" forceResolve="yes"
priorityOverride="default"
rn="rssubjFiltAtt-httpFilter" tDn="uni/tn-Tenant64/flt-httpFilter" tRn="flt-httpFilter"
 tnVzFilterName="httpFilter"/>
    </vzSubj>
   <vzSubj consMatchT="AtleastOne" name="subj64" provMatchT="AtleastOne" revFltPorts="yes"
 rn="subj-subj64">
        <vzRsSubjFiltAtt action="permit" directives="log" forceResolve="yes"
priorityOverride="default"
rn="rssubjFiltAtt-icmp" tDn="uni/tn-common/flt-icmp" tRn="flt-icmp" tnVzFilterName="icmp"/>

    </vzSubj>
</vzBrCP>
```

REST API を使用した禁止契約拒否ロギングの有効化

次の例は、REST API を使用して禁止契約拒否ロギングを有効にする方法を示しています。

手順

タブー契約を設定するロギングを拒否する、次の例のように XML で post を送信します。

例：

```
<vzTaboo dn="uni/tn-Tenant64/taboo-TCtrctPrefix" name="TCtrctPrefix" scope="context">
    <vzTSubj name="PrefSubj" rn="tsubj-PrefSubj"">
        <vzRsDenyRule directives="log" forceResolve="yes" rn="rsdenyRule-default"
tCl="vzFilter"
tDn="uni/tn-common/flt-default" tRn="flt-default"/>
    </vzTSubj>
</vzTaboo>
```

REST API を使用した ACL 許可および拒否ログ

次の例は、REST API を使用して、トラフィック フローのレイヤ 2 拒否ログ データを表示する方法を示しています。次の MO を使用してクエリを送信することができます。

- acllogDropL2Flow
- acllogPermitL2Flow
- acllogDropL3Flow
- acllogPermitL3Flow
- acllogDropL2Pkt
- acllogPermitL2Pkt

- acllogDropL3Pkt
- acllogPermitL3Pkt

始める前に

ACL契約許可および拒否ログのデータを表示する前に、許可または拒否ロギングを有効にする必要があります。

手順

レイヤ 3 ドロップ ログ データを表示するには、REST API を使用して次のクエリを送信します。

```
GET https://apic-ip-address/api/class/acllogDropL3Flow
```

例：

次の例では、サンプル出力をいくつか示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<imdata totalCount="2">
    <acllogPermitL3Flow childAction=""
dn="topology/pod-1/node-101/ndbgs/acllog/tn-common/ctx-inb

/permitl3flow-spctag-333-dpctag-444-sepgname-unknown-depgname-unknown-sip-[100:c000:a00:700:b00:0:f00:0]


-dip-[19.0.2.10]-proto-udp-sport-17459-dport-8721-smac-00:00:15:00:00:28-dmac-00:00:12:00:00:25-sintf-

        [port-channel5]-vrfencap-VXLAN: 2097153" dstEpgName="unknown" dstIp="19.0.2.10"
 dstMacAddr="00:00:12:00:00:25"
        dstPcTag="444" dstPort="8721" lcOwn="local" modTs="never" monPolDn=""
protocol="udp" srcEpgName="unknown"
        srcIntf="port-channel5" srcIp="100:c000:a00:700:b00:0:f00:0"
srcMacAddr="00:00:15:00:00:28" srcPcTag="333"
        srcPort="17459" status="" vrfEncap="VXLAN: 2097153"/>
    <acllogPermitL3Flow childAction=""
dn="topology/pod-1/node-102/ndbgs/acllog/tn-common/ctx-inb

/permitl3flow-spctag-333-dpctag-444-sepgname-unknown-depgname-unknown-sip-[100:c000:a00:700:b00:0:f00:0]-dip-


[19.0.2.10]-proto-udp-sport-17459-dport-8721-smac-00:00:15:00:00:28-dmac-00:00:12:00:00:25-sintf-

        [port-channel5]-vrfencap-VXLAN: 2097153" dstEpgName="unknown" dstIp="19.0.2.10"
 dstMacAddr="00:00:12:00:00:25"
        dstPcTag="444" dstPort="8721" lcOwn="local" modTs="never" monPolDn=""
protocol="udp" srcEpgName="unknown"
        srcIntf="port-channel5" srcIp="100:c000:a00:700:b00:0:f00:0"
srcMacAddr="00:00:15:00:00:28" srcPcTag="333"
        srcPort="17459" status="" vrfEncap="VXLAN: 2097153"/>
</imdata>
```
