REST API を使用した APIC の管理

管理アクセスの追加

インバンドおよびアウトオブバンド管理アクセス

管理テナントでは、ファブリック管理機能へのアクセスを設定するための便利な方法が提供されます。APIC を介してファブリック管理機能にアクセスできると同時に、インバンドおよびアウトオブバンドのネットワーク ポリシー経由で直接アクセスすることもできます。

スタティック管理アクセスについて

スタティック インバンドおよびアウトオブバンド管理接続の設定は、ダイナミック インバンドおよびアウトオブバンド管理接続の設定より簡単です。インバンドスタティック管理を設定する際に、各ノードの IP アドレスを指定し、一意の IP アドレスを割り当てることを確認する必要があります。ユーザがいくつかのリーフとスパイン スイッチの IP アドレスを管理する単純な展開、スタティック管理アクセスの設定は簡単です。さらに複雑な展開では、多くの IP アドレスの管理を必要とするリーフとスパイン スイッチの数が多いを持っているのスタティック管理アクセスは推奨されません。自動的に IP アドレスの重複を回避するダイナミック管理アクセスを設定することをお勧めします。

（注）
- インバンドまたはアウトオブバンドのいずれかのスタティックを設定することをお勧め管理またはインバンドおよびアウトオブバンドの動的な管理します。自分の導入では、2 つの方法を組み合わせて使用しません。
- インバンド管理アクセスでは、IPv4 アドレスと IPv6 アドレスがサポートされます。スタティック設定を使用した IPv6 設定がサポートされます（インバンドとアウトバンドの両方）。IPv4 および IPv6 のインバンドおよびアウトオブバンドのデュアル設定は、スタティック設定を使用する場合にのみサポートされます。詳細については、「*Configuring Static Management Access in Cisco APIC*」の KB 記事を参照してください。
- 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。

REST API を使用したインバンド管理アクセスの設定

インバンド管理アクセスでは、IPv4 アドレスと IPv6 アドレスがサポートされます。スタティック設定を使用した IPv6 設定がサポートされます（インバンドとアウトバンドの両方）。IPv4 および IPv6 のインバンドおよびアウトオブバンドのデュアル設定は、スタティック設定を使用する場合にのみサポートされます。詳細については、「*Configuring Static Management Access in Cisco APIC*」の KB 記事を参照してください。

手順

ステップ 1 VLAN ネームスペースを作成します。

例：

```
POST
https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/uni.xml -->
<polUni>
  <infraInfra>
    <!-- Static VLAN range -->
    <fvnsVlanInstP name="inband" allocMode="static">
      <fvnsEncapBlk name="encap" from="vlan-10" to="vlan-11"/>
    </fvnsVlanInstP>
  </infraInfra>
</polUni>
```

ステップ 2 物理ドメインを作成します。

例：

```
POST
https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/uni.xml -->
```

```
<polUni>
  <physDomP name="inband">
    <infraRsVlanNs tDn="uni/infra/vlanns-inband-static"/>
  </physDomP>
</polUni>
```

ステップ 3 インバンド管理用のセレクタを作成します。

例：

```
POST
https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <infraInfra>
    <infraNodeP name="vmmNodes">
      <infraLeafS name="leafS" type="range">
        <infraNodeBlk name="single0" from_="101" to_="101"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-vmmPorts"/>
    </infraNodeP>

    <!-- Assumption is that VMM host is reachable via eth1/40. -->
    <infraAccPortP name="vmmPorts">
      <infraHPortS name="portS" type="range">
        <infraPortBlk name="block1"
                      fromCard="1" toCard="1"
                      fromPort="40" toPort="40"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-inband" />
      </infraHPortS>
    </infraAccPortP>


    <infraNodeP name="apicConnectedNodes">
      <infraLeafS name="leafS" type="range">
        <infraNodeBlk name="single0" from_="101" to_="102"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-apicConnectedPorts"/>
    </infraNodeP>

    <!-- Assumption is that APIC is connected to eth1/1. -->
    <infraAccPortP name="apicConnectedPorts">
      <infraHPortS name="portS" type="range">
        <infraPortBlk name="block1"
                      fromCard="1" toCard="1"
                      fromPort="1" toPort="3"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-inband" />
      </infraHPortS>
    </infraAccPortP>

    <infraFuncP>
      <infraAccPortGrp name="inband">
        <infraRsAttEntP tDn="uni/infra/attentp-inband"/>
      </infraAccPortGrp>
    </infraFuncP>

    <infraAttEntityP name="inband">
      <infraRsDomP tDn="uni/phys-inband"/>
    </infraAttEntityP>
  </infraInfra>
</polUni>
```

ステップ 4 インバンド ブリッジ ドメインとエンドポイント グループ（EPG）を設定します。

例：

```
POST https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="mgmt">
    <!-- Configure the in-band management gateway address on the
         in-band BD. -->
    <fvBD name="inb">
      <fvSubnet ip="10.13.1.254/24"/>
    </fvBD>

    <mgmtMgmtP name="default">
      <!-- Configure the encap on which APICs will communicate on the
           in-band network. -->
      <mgmtInB name="default" encap="vlan-10">
        <fvRsProv tnVzBrCPName="default"/>
      </mgmtInB>
    </mgmtMgmtP>
  </fvTenant>
</polUni>
```

ステップ 5 アドレス プールを作成します。

例：

```
POST
https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="mgmt">
    <!-- Adresses for APIC in-band management network -->
    <fvnsAddrInst name="apicInb" addr="10.13.1.254/24">
      <fvnsUcastAddrBlk from="10.13.1.1" to="10.13.1.10"/>
    </fvnsAddrInst>

    <!-- Adresses for switch in-band management network -->
    <fvnsAddrInst name="switchInb" addr="10.13.1.254/24">
      <fvnsUcastAddrBlk from="10.13.1.101" to="10.13.1.120"/>
    </fvnsAddrInst>
  </fvTenant>
</polUni>
```

（注） IPv6 のダイナミック アドレス プールはサポートされていません。

ステップ 6 管理グループを作成します。

例：

```
POST
https://apic-ip-address/api/mo/uni.xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <infraInfra>
    <!-- Management node group for APICs -->
    <mgmtNodeGrp name="apic">
```

```
      <infraNodeBlk name="all" from_="1" to_="3"/>
      <mgmtRsGrp tDn="uni/infra/funcprof/grp-apic"/>
    </mgmtNodeGrp>

    <!-- Management node group for switches-->
    <mgmtNodeGrp name="switch">
      <infraNodeBlk name="all" from_="101" to_="104"/>
      <mgmtRsGrp tDn="uni/infra/funcprof/grp-switch"/>
    </mgmtNodeGrp>

    <!-- Functional profile -->
    <infraFuncP>
      <!-- Management group for APICs -->
      <mgmtGrp name="apic">
        <!-- In-band management zone -->
        <mgmtInBZone name="default">
          <mgmtRsInbEpg tDn="uni/tn-mgmt/mgmtp-default/inb-default"/>
          <mgmtRsAddrInst tDn="uni/tn-mgmt/addrinst-apicInb"/>
        </mgmtInBZone>
      </mgmtGrp>

      <!-- Management group for switches -->
      <mgmtGrp name="switch">
        <!-- In-band management zone -->
        <mgmtInBZone name="default">
          <mgmtRsInbEpg tDn="uni/tn-mgmt/mgmtp-default/inb-default"/>
          <mgmtRsAddrInst tDn="uni/tn-mgmt/addrinst-switchInb"/>
        </mgmtInBZone>
      </mgmtGrp>
    </infraFuncP>
  </infraInfra>
</polUni>
```

（注） IPv6 のダイナミック アドレス プールはサポートされていません。

REST API を使用した静的インバンド管理アクセスの設定

手順

ステップ 1 VLAN ネームスペースを作成します。

例：

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/uni.xml -->
<polUni>
  <infraInfra>
    <!-- Static VLAN range -->
    <fvnsVlanInstP name="inband" allocMode="static">
      <fvnsEncapBlk name="encap" from="vlan-10" to="vlan-11"/>
    </fvnsVlanInstP>
  </infraInfra>
</polUni>
```

ステップ 2 物理ドメインを作成します。

例：

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/uni.xml -->
<polUni>
  <physDomP name="inband">
    <infraRsVlanNs tDn="uni/infra/vlanns-inband-static"/>
  </physDomP>
</polUni>
```

ステップ 3 インバンド管理用のセレクタを作成します。

例：

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <infraInfra>
    <infraNodeP name="vmmNodes">
      <infraLeafS name="leafS" type="range">
        <infraNodeBlk name="single0" from_="101" to_="101"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-vmmPorts"/>
    </infraNodeP>

    <!-- Assumption is that VMM host is reachable via eth1/40. -->
    <infraAccPortP name="vmmPorts">
      <infraHPortS name="portS" type="range">
        <infraPortBlk name="block1"
                      fromCard="1" toCard="1"
                      fromPort="40" toPort="40"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-inband" />
      </infraHPortS>
    </infraAccPortP>


    <infraNodeP name="apicConnectedNodes">
      <infraLeafS name="leafS" type="range">
        <infraNodeBlk name="single0" from_="101" to_="102"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-apicConnectedPorts"/>
    </infraNodeP>

    <!-- Assumption is that APIC is connected to eth1/1. -->
    <infraAccPortP name="apicConnectedPorts">
      <infraHPortS name="portS" type="range">
        <infraPortBlk name="block1"
                      fromCard="1" toCard="1"
                      fromPort="1" toPort="3"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-inband" />
      </infraHPortS>
    </infraAccPortP>

    <infraFuncP>
      <infraAccPortGrp name="inband">
        <infraRsAttEntP tDn="uni/infra/attentp-inband"/>
      </infraAccPortGrp>
    </infraFuncP>

    <infraAttEntityP name="inband">
      <infraRsDomP tDn="uni/phys-inband"/>
    </infraAttEntityP>
  </infraInfra>
</polUni>
```

ステップ 4 インバンド ブリッジ ドメインとエンドポイント グループ（EPG）を設定します。

例：

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="mgmt">
    <!-- Configure the in-band management gateway address on the
         in-band BD. -->
    <fvBD name="inb">
      <fvSubnet ip="<subnet_ip_address>"/>
    </fvBD>

    <mgmtMgmtP name="default">
      <!-- Configure the encap on which APICs will communicate on the
           in-band network. -->
      <mgmtInB name="default" encap="vlan-10">
        <fvRsProv tnVzBrCPName="default"/>
      </mgmtInB>
    </mgmtMgmtP>
  </fvTenant>
</polUni>
```

ステップ 5 スタティック インバンド管理 IP アドレスを作成し、ノード ID に割り当てます。

例：

```
<polUni>
  <fvTenant name="mgmt">
    <mgmtMgmtP name="default">
      <mgmtInB name="default">
        <mgmtRsInBStNode tDn="topology/pod-1/node-101"
                         addr="<ip_address_1>"
                         gw="<gw_address>”
    v6Addr = “<ip6_address_1>”
    v6Gw = “<ip6_gw_address>"/>
        <mgmtRsInBStNode tDn="topology/pod-1/node-102"
                         addr="<ip_address_2>"
                         gw="<gw_address>”
    v6Addr = “<ip6_address_2>"
    v6Gw = “<ip6_gw_address>"/>
        <mgmtRsInBStNode tDn="topology/pod-1/node-103"
                         addr="<ip_address_3>"
                         gw="<gw_address>”
    v6Addr = “<ip6_address_3>"
    v6Gw = “<ip6_gw_address>"/>
        <mgmtRsInBStNode tDn="topology/pod-1/node-104"
                         addr="<ip_address_4>"
                         gw="<gw_address>”
    v6Addr = “<ip6_address_4>"
    v6Gw = “<ip6_gw_address>"/>

        <mgmtRsInBStNode tDn="topology/pod-1/node-105"
                         addr="<ip_address_5>"
                         gw="<gw_address>”
    v6Addr = “<ip6_address_5>"
    v6Gw = “<ip6_gw_address>"/>

      </mgmtInB>
    </mgmtMgmtP>
```

```
  </fvTenant>
</polUni>
```

REST API を使用したアウトオブバンド管理アクセスの設定

アウトオブバンド管理アクセスでは、IPv4 アドレスと IPv6 アドレスがサポートされます。

始める前に

APIC アウトオブバンド管理接続のリンクは、1 Gbps である必要があります。

手順

ステップ 1 アウトオブバンド コントラクトを作成します。

例：

```
POST https://apic-ip-address/api/mo/uni.xml

<polUni>
    <fvTenant name="mgmt">
        <!-- Contract -->
        <vzOOBBrCP name="oob-default">
            <vzSubj name="oob-default">
                <vzRsSubjFiltAtt tnVzFilterName="default" />
            </vzSubj>
        </vzOOBBrCP>
    </fvTenant>
</polUni>
```

ステップ 2 アウトオブバンド コントラクトをアウトオブバンド EPG に関連付けます。

例：

```
POST https://apic-ip-address/api/mo/uni.xml

<polUni>
    <fvTenant name="mgmt">
        <mgmtMgmtP name="default">
            <mgmtOoB name="default">
                <mgmtRsOoBProv tnVzOOBBrCPName="oob-default" />
            </mgmtOoB>
        </mgmtMgmtP>
    </fvTenant>
</polUni>
```

ステップ 3 アウトオブバンド コントラクトを外部管理 EPG に関連付けます。

例：

```
POST https://apic-ip-address/api/mo/uni.xml

<polUni>
    <fvTenant name="mgmt">
        <mgmtExtMgmtEntity name="default">
```

```
            <mgmtInstP name="oob-mgmt-ext">
                <mgmtRsOoBCons tnVzOOBBrCPName="oob-default" />
                <!-- SUBNET from where switches are managed -->
                <mgmtSubnet ip="10.0.0.0/8" />
            </mgmtInstP>
        </mgmtExtMgmtEntity>
    </fvTenant>
</polUni>
```

ステップ 4 管理アドレス プールを作成します。

例：

```
POST https://apic-ip-address/api/mo/uni.xml

<polUni>
    <fvTenant name="mgmt">
        <fvnsAddrInst name="switchOoboobaddr" addr="172.23.48.1/21">
            <fvnsUcastAddrBlk from="172.23.49.240" to="172.23.49.244"/>
        </fvnsAddrInst>
    </fvTenant>
</polUni>
```

ステップ 5 ノード管理グループを作成します。

例：

```
POST https://apic-ip-address/api/mo/uni.xml

<polUni>
    <infraInfra>
        <infraFuncP>
            <mgmtGrp name="switchOob">
                <mgmtOoBZone name="default">
                    <mgmtRsAddrInst tDn="uni/tn-mgmt/addrinst-switchOoboobaddr" />
                    <mgmtRsOobEpg tDn="uni/tn-mgmt/mgmtp-default/oob-default" />
                </mgmtOoBZone>
            </mgmtGrp>
        </infraFuncP>
        <mgmtNodeGrp name="switchOob">
            <mgmtRsGrp tDn="uni/infra/funcprof/grp-switchOob" />
            <infraNodeBlk name="default" from_="101" to_="103" />
        </mgmtNodeGrp>
    </infraInfra>
</polUni>
```

（注） デフォルトの接続モードとしてアウトオブバンド管理接続を使用するように APIC サーバを設定できます。

```
POST https://apic-ip-address/api/node/mo/.xml
<polUni>
<fabricInst>
    <mgmtConnectivityPrefs interfacePref="ooband"/>
</fabricInst>
</polUni>
```

REST API を使用したスタティック アウトオブバンド管理アクセスの設定

始める前に

APIC アウトオブバンド管理接続のリンクは、1 Gbps である必要があります。

手順

ステップ 1 アウトオブバンド コントラクトを作成します。

例：

```
<polUni>
    <fvTenant name="mgmt">
        <!-- Contract -->
        <vzOOBBrCP name="oob-default">
            <vzSubj name="oob-default">
                <vzRsSubjFiltAtt tnVzFilterName="default" />
            </vzSubj>
        </vzOOBBrCP>
    </fvTenant>
</polUni>
```

ステップ 2 アウトオブバンド コントラクトをアウトオブバンド EPG に関連付けます。

例：

```
<polUni>
    <fvTenant name="mgmt">
        <mgmtMgmtP name="default">
            <mgmtOoB name="default">
                <mgmtRsOoBProv tnVzOOBBrCPName="oob-default" />
            </mgmtOoB>
        </mgmtMgmtP>
    </fvTenant>
</polUni>
```

ステップ 3 アウトオブバンド コントラクトを外部管理 EPG に関連付けます。

例：

```
<polUni>
    <fvTenant name="mgmt">
        <mgmtExtMgmtEntity name="default">
            <mgmtInstP name="oob-mgmt-ext">
                <mgmtRsOoBCons tnVzOOBBrCPName="oob-default" />
                <!-- SUBNET from where switches are managed -->
                <mgmtSubnet ip="<mgmt_subnet_ip_address>" />
            </mgmtInstP>
        </mgmtExtMgmtEntity>
    </fvTenant>
</polUni>
```

ステップ 4 スタティック アウトオブバンド管理 IP アドレスを作成し、ノード ID に割り当てます。

IP アドレスのチェック

例：

```
<polUni>
  <fvTenant name="mgmt">
    <mgmtMgmtP name="default">
      <mgmtOoB name="default">
        <mgmtRsOoBStNode tDn="topology/pod-1/node-101"
                         addr="<ip_address_1>"
                         gw="<gw_address>"/>
        <mgmtRsOoBStNode tDn="topology/pod-1/node-102"
                         addr="<ip_address_2>"
                         gw="<gw_address>"/>
        <mgmtRsOoBStNode tDn="topology/pod-1/node-103"
                         addr="<ip_address_3>"
                         gw="<gw_address>"/>
      </mgmtOoB>
    </mgmtMgmtP>
  </fvTenant>
</polUni>
```

コンフィギュレーション ファイルの管理

概要

このトピックでは、次の情報を提供します。

- Cisco APIC の設定のインポートとエクスポートを使用して、設定の状態を最新の既知の良好な状態に回復する方法
- Cisco APIC の設定ファイルのセキュア プロパティを暗号化する方法

ユーザ設定のスケジュール バックアップとオンデマンド バックアップの両方を行うことができます。設定の状態を回復すると（「ロールバック」とも呼ばれます）、以前良好であった既知の状態に戻ることができます。そのためのオプションは、アトミック置換と呼ばれます。設定インポート ポリシー（configImportP）は、アトミック + 置換（importMode=atomic、importType=replace）をサポートします。これらの値に設定すると、インポートされる設定が既存の設定を上書きし、インポートされるファイルに存在しない既存の設定があれば削除されます。定期的に設定のバックアップとエクスポートを行うか、既知の良好な設定のエクスポートを明示的にトリガーすれば、後で以下の CLI、REST API、および GUI 用の手順を使用してこの設定を復元できます。

Cisco APIC を使用した設定状態の回復に関する詳細な概念情報については、『*Cisco Application Centric Infrastructure Fundamentals Guide*』を参照してください。

次の項では、設定ファイルのセキュア プロパティの暗号化に関する概念情報を提供します。

設定ファイルのバックアップ、復元、およびロールバックのワークフロー

この項では、設定ファイルのバックアップ、復元、およびロールバックのワークフローについて説明します。本書で説明されている機能はすべて同じワークフローパターンに従います。対応するポリシーを設定すると、ジョブをトリガーするために **admintSt** を **triggered** に設定する必要があります。

ジョブがトリガーされると、**configJobCont** タイプのコンテナ オブジェクトで **configJob** タイプのオブジェクト（実行を表す）が作成されます（Naming プロパティの値はポリシー DN に設定されます）。コンテナの **lastJobName** フィールドを使用して、そのポリシーに対してトリガーされた最後のジョブを確認することができます。

（注） 同時に最大 5 つの **configJob** オブジェクトが単一ジョブ コンテナに保持され、それぞれの新規ジョブがトリガーされます。そのために、最も古いジョブは削除されます。

configJob オブジェクトには、次の情報が含まれています。

- 実行時間
- 処理または生成されるファイルの名前
- 以下のステータス：
  - Pending
  - Running
  - 不合格
  - Fail-no-data
  - Success
  - Success-with-warnings
- 詳細の文字列（障害メッセージと警告）
- 進捗率 = 100 * lastStepIndex/totalStepCount
- 最後に行われた内容を示す lastStepDescr フィールド

About Configuration Export to Controllers

Configuration export extracts user-configurable managed object (MO) trees from all 32 shards in the cluster, writes them into separate files, then compresses them into a tar gzip file. The configuration export then uploads the tar gzip file to a preconfigured remote location (configured through **configRsRemotePath** pointing to a **fileRemotePath** object) or stores it as a **snapshot** on the controller(s).

（注） 詳細については、「スナップショット」の項を参照してください。

configExportP ポリシーは次のように設定されます。

- **name**—Policy name.
- **format**—Format in which the data is stored inside the exported archive (xml or json).
- **targetDn**—The domain name (DN) of the specific object you want to export. (空白にするとすべて。)
- **snapshot**—When true, the file is stored on the controller; no remote location configuration is needed.
- **includeSecureFields**—Set to true by default, this indicates whether the encrypted fields (passwords, etc.) should be included in the export archive.

（注） The **configSnapshot** object is created holding the information about this snapshot. (「スナップショット」の項を参照してください)。

エクスポートのスケジューリング

An export policy can be linked with a scheduler, which triggers the export automatically based on a preconfigured schedule. これはで行います、 **configRsExportScheduler** をポリシーからの関係を **trigSchedP** オブジェクト。(例 Configuration] セクションを参照してください)。

（注） スケジューラーはオプションです。ポリシーは、adminSt を **triggered** に設定することにより、いつでもトリガーできます。

コントローラへの設定のインポートについて

設定のインポートでは、指定されている以前にエクスポートされたアーカイブのダウンロード、抽出、解析、分析、および適用を、一度に 1 つのシャードずつ行います（infra、fabric、tn-common、その他すべて、の順）。fileRemotePath 設定は、エクスポートの場合と同様に実行されます（configRsRemotePath を使用）。スナップショットのインポートもサポートされます。

configImportP ポリシーは次のように設定されます。

- **name**—Policy name
- **fileName**—Name of the archive file (not the path file) to be imported
- **importMode**
  - ベスト エフォート モード：各 MO は個々に適用され、エラーがあっても無効な MO がスキップされるだけです。

（注） オブジェクトがコントローラに存在しない場合、そのオブジェクトの子は設定されません。ベスト エフォート モードでは、オブジェクトの子を設定しようとします。

- アトミック モード：設定はシャード全体で適用されます。1 つのエラーがあると、シャード全体が元の状態にロールバックされます。

- **importType**
  - 交換: 現在の system configuration(システム設定、システム構成) がまたはの内容をインポートするアーカイブに置き換えられます。(アトミック モードのみがサポートされます。)
  - マージ：何も削除されず、ANDアーカイブの内容が既存のシステム設定上に適用されます。

- **snapshot**—When true, the file is taken from the controller and no remote location configuration is needed.
- **failOnDecryptErrors**—(true by default) The file fails to import if the archive was encrypted with a different key than the one that is currently set up in the system.

トラブルシューティング

以下のシナリオでは、トラブルシューティングが必要な可能性があります。

- 生成されたアーカイブをリモートロケーションからダウンロードできなかった場合は、接続の問題に関する項を参照してください。
- インポートは正常に終了したが警告が表示された場合は、詳細を確認してください。
- ファイルを解析できなかった場合は、以下のシナリオを参照してください。
  - ファイルが有効なXMLまたはJSONファイルでない場合は、エクスポートされたアーカイブから取得したファイルが手動で変更されたかどうかを確認してください。
  - オブジェクトプロパティに未知のプロパティまたはプロパティ値がある場合は、以下の原因が考えられます。
    - プロパティが削除されたか、または未知のプロパティ値が手動で入力された.
    - モデル タイプの範囲が変更された（後方互換性がないモデル変更）。
    - 名前付けプロパティ リストが変更された.
- MO を設定できなかった場合は、以下に注意してください。
  - ベスト エフォート モードでは、エラーをログに記録し、その MO をスキップします.
  - アトミック モードでは、エラーをログに記録し、シャードをスキップします.

設定ファイルの暗号化

リリース 1.1(2) 以降では、AES-256 暗号化を有効にすることにより APIC 設定ファイルのセキュア プロパティを暗号化できます。AES 暗号化はグローバル設定オプションです。すべてのセキュア プロパティは AES 構成設定に従っています。テナント設定などの ACI ファブリック設定のサブセットを AES 暗号化を使用してエクスポートするが、ファブリック設定の残りの部分は暗号化しないということはできません。セキュア プロパティのリストについては、*Cisco Application Centric Infrastructure Fundamentals*の「Appendix K: Secure Properties」を参照してください。

APIC は、16 ～ 32 文字のパスフレーズを使用して AES-256 キーを生成します。APIC GUI には、AES パスフレーズのハッシュが表示されます。このハッシュを使用して、2 つの ACI ファブリックで同じパスフレーズが使用されているかどうかを確認できます。このハッシュをクライアント コンピュータにコピーして、別の ACI ファブリックのパスフレーズ ハッシュと比較できます。これにより、それらのハッシュが同じパスフレーズを使用して生成されたかどうかを確認できます。ハッシュを使用して、元のパスフレーズまたは AES-256 キーを再構築することはできません。

暗号化された設定ファイルを使用する際は、次のガイドラインに従ってください。

- AES 暗号化設定オプションを使用しているファブリックに古い ACI 設定をインポートするための後方互換性がサポートされています。

  （注） 逆の互換性はサポートされていません。AES 暗号化が有効になっている ACI ファブリックからエクスポートされた設定を古いバージョンの APIC ソフトウェアにインポートすることはできません。

- ファブリック バックアップ設定のエクスポートを実行するときは、必ず AES 暗号化を有効にします。これにより、ファブリックを復元するときに、設定のすべてのセキュア プロパティが正常にインポートされるようになります。

  （注） AES 暗号化を有効にせずにファブリック バックアップ設定がエクスポートされると、どのセキュア プロパティもエクスポートに含まれません。そのような暗号化されていないバックアップにはセキュア プロパティは何も含まれていないため、そのようなファイルをインポートしてシステムを復元すると、ファブリックの管理者およびすべてのユーザがシステムからロックアウトされてしまう可能性があります。

- 暗号化キーを生成する AES パスフレーズは、ACI 管理者やその他のユーザが復元したり読み取ったりすることはできません。AES パスフレーズは保存されません。APIC は AES パスフレーズを使用して AES キーを生成した後、そのパスフレーズを廃棄します。AES キーはエクスポートされません。AES キーは、エクスポートされず、REST API を使用して取得できないため、復元できません。

- 同じ AES-256 パスフレーズは、常に同じ AES-256 キーを生成します。設定のエクスポート ファイルは、同じ AES パスフレーズを使用する他の ACI ファブリックにインポートできます。
- トラブルシューティングを目的として、セキュアプロパティの暗号化データが含まれていない設定ファイルをエクスポートします。設定のエクスポートを実行する前に一時的に暗号化をオフにすると、エクスポートされた設定からすべてのセキュアプロパティ値が削除されます。すべてのセキュアプロパティが削除されたそのような設定ファイルをインポートするには、インポート マージ モードを使用します。インポート置換モードは使用しません。インポート マージ モードを使用すると、ACI ファブリック内の既存セキュア プロパティが保持されます。
- デフォルトで、APICは復号できないフィールドが含まれているファイルの設定のインポートを拒否します。この設定をオフにするときは注意してください。このデフォルト設定がオフになっているときに設定のインポートが適切に実行されないと、ファブリックのAES暗号化設定に一致しない設定ファイルのインポート時にACIファブリックのすべてのパスワードが削除される可能性があります。

（注） このガイドラインに従わないと、ファブリック管理者を含むすべてのユーザがシステムからロックアウトされる可能性があります。

fileRemotePath オブジェクトについて

fileRemotePath オブジェクトは、以下のリモート ロケーションパスのパラメータを保持しています。

- ホスト名または IP
- ポート
- プロトコル：FTP、SCP など
- リモート ディレクトリ（ファイル パスではない）
- [ユーザ名（Username）]
- パスワード

（注） パスワードは、変更するたびに再送信する必要があります。

設定例

以下に設定サンプルを示します。

fabricInst（uni/fabric）の下に、次のように入力します。

```
<fileRemotePath name="path-name" host="host name or ip" protocol="scp"
remotePath="path/to/some/folder" userName="user-name" userpasswd="password" />
```

REST API を使用したリモート ロケーションの設定

この手順では、REST API を使用してリモート ロケーションを作成する方法について説明します。

```
<fileRemotePath name="local" host="host or ip" protocol="ftp|scp|sftp" remotePath="path
 to folder" userName="uname" userPasswd="pwd" />
```

REST API を使用したコントローラに設定ファイル エクスポートを設定する

始める前に

リモート パスおよびスケジューリング ポリシーを作成します。

（注） リモート ロケーションを提供する時、`True` にスナップショットを設定している場合、バックアップではリモート パスを無視して、コントローラにファイルを保存します。

手順

次の例のように、 XML で POST 要求を送信して設定エクスポート ポリシーを作成します。

例：

```
<configExportP name="policy-name" format="xml" targetDn="/some/dn or empty which means
everything"
snapshot="false" adminSt="triggered">
<configRsRemotePath tnFileRemotePathName="some remote path name" />
<configRsExportScheduler tnTrigSchedPName="some scheduler name" />
</configExportP>
```

REST API を使用した設定ファイル インポート ポリシーの設定

手順

設定ファイル インポート ポリシーを設定し、次の例のように XML で POST 送信します。

例：

```
<configImportP name="policy-name" fileName="someexportfile.tgz" importMode="atomic"
     importType="replace" snapshot="false" adminSt="triggered">
<configRsRemotePath tnFileRemotePathName="some remote path name" />
</configImportP>
```

REST API を使用した設定ファイルの暗号化

手順

REST API を使用して設定ファイルを暗号化するには、次の例のような XML を POST 送信します。

例：

```
https://apic-ip-address/api/mo/uni/fabric.xml
<pkiExportEncryptionKey passphrase="abcdefghijklmnopqrstuvwxyz"
strongEncryptionEnabled="true"/>
```

スナップショットおよびロールバック

スナップショット

スナップショットは設定のバックアップのアーカイブであり、コントローラで管理されているフォルダに保存（および複製）されます。スナップショットを作成するには、**snapshot** プロパティを true に設定してエクスポートを実行します。この場合、リモート パスの設定は不要です。スナップショットをユーザに公開するために、**configSnapshot** タイプのオブジェクトが作成されます。

[管理] > **[インポート/エクスポート]** > **[エクスポート ポリシー]** > **[設定]** > **[defaultAuto]** に保存される定期的なスナップショットを作成できます。

configSnapshot オブジェクトは以下を提供します。

- ファイル名
- ファイル サイズ
- 作成日
- 何のスナップショットであるかを示すルート DN（ファブリック、インフラ、特定のテナントなど）
- スナップショットを削除する機能（retire フィールドを true に設定）

スナップショットをインポートするには、最初にインポート ポリシーを作成します。**[管理]** > **[インポート/エクスポート]** に移動し、**[インポート ポリシー]** をクリックします。右クリックし、**[設定のインポート ポリシーの作成]** を選択して、インポート ポリシーの属性を設定します。

ロールバックについて

configRollbackP ポリシーは、2つのスナップショット間で行われた変更を元に戻すために使用されます。管理対象オブジェクト（MO）は次のように処理されます。

- 削除された MO を再作成します.
- 作成された MO を削除します.
- 変更された MO を元に戻します.

（注） ロールバック機能はスナップショットでのみ動作します。リモートアーカイブはサポートされません。リモートアーカイブのデータを使用する場合は、ロールバックのデータからスナップショットを作成するスナップショット マネージャを使用します。ポリシーでは、リモート パス設定は不要です。

ロールバックのワークフロー

ポリシーの snapshotOneDN フィールドと snapshotTwoDn フィールドを設定する必要があり、最初のスナップショット（S1）がスナップショット2（S2）より前である必要があります。トリガーされると、スナップショットが抽出および分析され、それらの間の違いが計算され、適用されます。

MO の場所：

- S1 に存在するが、S2 には存在しない：これらの MO は削除され、ロールバックにより再作成されます.
- S1 には存在しないが、S2 には存在する：これらの MO は S1 後に作成されており、以下に該当する場合はロールバックにより削除されます。
  - これらの MO は S2 取得後に変更されていない.
  - S2 取得後に作成または変更された MO の子孫がない.
- S1 と S2 の両方に存在するが、プロパティ値は異なる：S2 取得後にプロパティが別の値に変更されていない限り、これらの MO プロパティは S1 に戻されます。この場合、現状どおりになります。

  ロールバック機能では、これらの計算の結果として生成された設定が含まれている diff ファイルも生成されます。この設定の適用は、ロールバック プロセスの最後のステップです。このファイルの内容は、readiff と呼ばれる特殊な REST API を使用して取得できます。apichost/mqapi2/snapshots.readiff.xml?jobdn=SNAPSHOT_JOB_DN

ロールバック（予測は困難）にはプレビューモード（previewをtrueに設定）もあり、ロールバックにより実際の変更が行われないようにします。diff ファイルを計算して生成し、ロールバックを実際に実行したときに何が発生するかを正確にプレビューできます。

Diff ツール

2 つのスナップショット間の diff 機能を提供する別の特殊な REST API を使用できます。
apichost/mqapi2/snapshots.diff.xml?s1dn=SNAPSHOT_ONE_DN&s2dn=SNAPSHOT_TWO_DN

REST API を使用したアップロードおよびダウンロード

configSnapshotManagerP ポリシーを使用すると、リモートで保存したエクスポート アーカイブのスナップショットを作成することができます。ポリシーにリモートパスを付加し、ファイル名（configImportP と同じ）を指定し、モードをダウンロードに設定し、トリガーすることができます。マネージャは、ファイルをダウンロードし、そのファイルを分析してアーカイブが有効であることを確認し、そのファイルをコントローラに保存し、対応する configSnapshot オブジェクトを作成します。スナップショット マネージャを使用すると、リモート ロケーションにスナップショット アーカイブをアップロードすることもできます。この場合、モードをアップロードに設定する必要があります。

始める前に

リモートで保存されているアーカイブをセット アップします。

手順

スナップショット ポリシーをアップロードまたはダウンロードするには、XML で POST 要求を次のように送信します。

例：

```
<configSnapshotManagerP name="policy-name" fileName="someexportfile.tgz"
    mode="upload|download" adminSt="triggered">
<configRsRemotePath tnFileRemotePathName="some remote path name" />
</configSnapshotManagerP>
```

設定と、REST API を使用して、コンフィギュレーションのロールバックの実行

始める前に

ロールバック ポリシーとスナップショットを作成します。

手順

ロールバックの実行の設定と、XML で POST 要求を次のようを送信します。

例：

```
<configRollbackP name="policy-name" snapshotOneDn="dn/of/snapshot/one"
snapshotOneDn="dn/of/snapshot/two" preview="false" adminSt="triggered" />
```

設定ゾーンの使用

設定ゾーン

設定ゾーンはACIファブリックを複数のゾーンに分割します。これらのゾーンは、別々のタイミングで設定を変更をして更新することができます。これにより、障害のある設定がファブリック全体に導入されるリスクが限定され、トラフィックが中断したり、さらにはファブリックがダウンしたりする可能性が抑えられます。管理者は、あまり重要ではないゾーンに設定を導入した後に、適切であることを確認してから重要なゾーンに導入することができます。

設定ゾーンの動作は次のポリシーによって指定します、

- `infrazone:ZoneP` は、システム アップグレードに自動的に作成されます。削除または変更することはできません。
- `infrazone:Zone` 1 つ以上のポッド グループが含まれています (`PodGrp`) または 1 つまたは複数のノード グループ (`NodeGrp`)。

  （注） 選択できるだけ `PodGrp` または `NodeGrp` ; 両方を選択することはできません。

  ノードは 1 個のゾーン（`infrazone:Zone`）だけに所属できます`NodeGrp` には、名前および導入モードという 2 つのプロパティがあります。導入モード プロパティは次のとおりです。

  - `enabled` - Pending updates are sent immediately.
  - `disabled` - New updates are postponed.

  （注） disabled の設定ゾーンでは、ノードのアップグレード、ダウングレード、コミッション、デコミッションは行わないでください。

- triggered：保留中の更新がただちに送信され、導入モードが triggered への変更前の値に自動的にリセットされます。

所定のノードセットでポリシーを作成、変更、または削除されると、ポリシーが導入されている各ノードに更新が送信されます。ポリシーのクラスと infrazone 設定に基づいて、次のような処理が行われます。

- infrazone 設定に従わないポリシーの場合、APIC がすべてのファブリック ノードにただちに更新を送信します。
- infrazone 設定に従うポリシーの場合は、infrazone 設定に従って更新が続行します。
  - ノードが infrazone:Zone に含まれている場合、更新は、ゾーンの導入モードが有効に設定されていればただちに送信されます。それ以外では更新は保留になります。
  - ノードが infrazone:Zone に含まれている場合は、すぐに更新が実行されます。これは ACI ファブリックのデフォルトの動作です。

設定ゾーンのサポート対象ポリシー

設定ゾーンでは次のポリシーがサポートされています。

```
analytics:CfgSrv
bgp:InstPol
callhome:Group
callhome:InvP
callhome:QueryGroup
cdp:IfPol
cdp:InstPol
comm:Pol
comp:DomP
coop:Pol
datetime:Pol
dbgexp:CoreP
dbgexp:TechSupP
dhcp:NodeGrp
dhcp:PodGrp
edr:ErrDisRecoverPol
ep:ControlP
ep:LoopProtectP
eqptdiagp:TsOdFabP
eqptdiagp:TsOdLeafP
fabric:AutoGEp
fabric:ExplicitGEp
fabric:FuncP
fabric:HIfPol
fabric:L1IfPol
fabric:L2IfPol
fabric:L2InstPol
fabric:L2PortSecurityPol
fabric:LeCardP
fabric:LeCardPGrp
fabric:LeCardS
fabric:LeNodePGrp
fabric:LePortP
fabric:LePortPGrp
fabric:LFPortS
```

```
fabric:NodeControl
fabric:OLeafS
fabric:OSpineS
fabric:PodPGrp
fabric:PortBlk
fabric:ProtGEp
fabric:ProtPol
fabric:SFPortS
fabric:SpCardP
fabric:SpCardPGrp
fabric:SpCardS
fabric:SpNodePGrp
fabric:SpPortP
fabric:SpPortPGrp
fc:DomP
fc:FabricPol
fc:IfPol
fc:InstPol
file:RemotePath
fvns:McastAddrInstP
fvns:VlanInstP
fvns:VsanInstP
fvns:VxlanInstP
infra:AccBaseGrp
infra:AccBndlGrp
infra:AccBndlPolGrp
infra:AccBndlSubgrp
infra:AccCardP
infra:AccCardPGrp
infra:AccNodePGrp
infra:AccPortGrp
infra:AccPortP
infra:AttEntityP
infra:CardS
infra:ConnFexBlk
infra:ConnFexS
infra:ConnNodeS
infra:DomP
infra:FexBlk
infra:FexBndlGrp
infra:FexGrp
infra:FexP
infra:FuncP
infra:HConnPortS
infra:HPathS
infra:HPortS
infra:LeafS
infra:NodeBlk
infra:NodeGrp
infra:NodeP
infra:OLeafS
infra:OSpineS
infra:PodBlk
infra:PodGrp
infra:PodP
infra:PodS
infra:PolGrp
infra:PortBlk
infra:PortP
infra:PortS
infra:PortTrackPol
infra:Profile
infra:SHPathS
infra:SHPortS
```

```
infra:SpAccGrp
infra:SpAccPortGrp
infra:SpAccPortP
infra:SpineP
infra:SpineS
isis:DomPol
l2ext:DomP
l2:IfPol
l2:InstPol
l2:PortSecurityPol
l3ext:DomP
lacp:IfPol
lacp:LagPol
lldp:IfPol
lldp:InstPol
mcp:IfPol
mcp:InstPol
mgmt:NodeGrp
mgmt:PodGrp
mon:FabricPol
mon:InfraPol
phys:DomP
psu:InstPol
qos:DppPol
snmp:Pol
span:Dest
span:DestGrp
span:SpanProv
span:SrcGrp
span:SrcTargetShadow
span:SrcTargetShadowBD
span:SrcTargetShadowCtx
span:TaskParam
span:VDest
span:VDestGrp
span:VSpanProv
span:VSrcGrp
stormctrl:IfPol
stp:IfPol
stp:InstPol
stp:MstDomPol
stp:MstRegionPol
trig:SchedP
vmm:DomP
vpc:InstPol
vpc:KAPol
```

REST API を使用した設定ゾーンの作成

始める前に

この手順では、REST API を使用して設定ゾーンを作成する方法について説明します。

手順

下の例のように、REST API リーフ スイッチまたはポッドを使用して設定ゾーンを作成します。

例：

リーフ スイッチを使用した設定ゾーンの作成

```
<infraInfra>
<infrazoneZoneP name="default">
<infrazoneZone name="Group1" deplMode="disabled">
<infrazoneNodeGrp name="nodeGroup">
<infraNodeBlk name="nodeblk1" from_=101 to_=101/>
<infraNodeBlk name="nodeblk2" from_=103 to_=103/>
</infrazoneNodeGrp>
</infrazoneZone>
<infrazoneZone name="Group2" deplMode="enabled">
<infrazoneNodeGrp name="nodeGroup2">
<infraNodeBlk name="nodeblk" from_=102 to_=102/>
</infrazoneNodeGrp>
</infrazoneZone>
</infrazoneZoneP>
</infraInfra>
```

例：

ポッドを使用した設定ゾーンの作成

```
<infraInfra>
    <infrazoneZoneP name="default">
        <infrazoneZone name="testZone" descr="testZone-Description" deplMode="enabled">
            <infrazonePodGrp name="podGroup1">
                <infraPodBlk name="group1" from_=101 to_=101/>
                <infraPodBlk name="group2" from_=103 to_=103/>
            </infrazonePodGrp>
            <infrazonePodGrp name="podGroup2">
                <infraPodBlk name="group" from_=102 to_=102/>
            </infrazonePodGrp>
        </infrazoneZone>
    </infrazoneZoneP>
</infraInfra>
```