



非管理対象モードの設定

- [非管理対象モードについて \(1 ページ\)](#)
- [管理対象および非管理対象の論理デバイスについて \(2 ページ\)](#)
- [管理対象および非管理対象の機能ノードについて \(2 ページ\)](#)
- [レイヤ 4～レイヤ 7 サービスのエンドポイントグループについて \(3 ページ\)](#)
- [グラフ コネクタに対する静的なカプセル化の使用 \(4 ページ\)](#)
- [NX-OS スタイルの CLI を使用した物理デバイスの作成 \(4 ページ\)](#)
- [NX-OS スタイルの CLI を使用したハイ アベイラビリティ クラスタの作成 \(5 ページ\)](#)
- [NX-OS スタイルの CLI を使用した仮想デバイスの作成 \(7 ページ\)](#)
- [非管理対象モードの XML の例 \(8 ページ\)](#)
- [非管理対象モードの動作 \(10 ページ\)](#)

非管理対象モードについて

レイヤ 4～レイヤ 7 サービスの挿入機能によって、管理者は 1 つ以上のサービスを 2 つのエンドポイント グループ間に挿入できます。Application Policy Infrastructure Controller (APIC) はサービスにファブリック リソース (VLAN) を割り当て、サービス グラフに指定された設定に従ってファブリック (リーフ スイッチ) とサービス アプライアンスをプログラミングします。サービスをサービス グラフの一部として使用できるようにするには、APIC にそのサービスのデバイス パッケージが必要です。APIC はグラフのインスタンス化時にもサービス アプライアンスをプログラミングします。

APIC で、サービス グラフに対してネットワーク リソースのみを割り当てて、グラフのインスタンス化時にファブリック側のみをプログラミングすることができます。サービス アプライアンスのプログラミングにより適した既存のオーケストレータまたは dev-op ツールが環境にすでにあるなど、さまざまな理由でこれが必要になる場合があります。また、サービス アプライアンスのデバイス パッケージが使用できない場合もあります。

サービスの非管理対象モードでは、ネットワーク リソースの割り当てや、ファブリックのプログラミングに対する APIC の動作を選択できます。非管理対象モードを有効にすると、APIC はネットワーク リソースのみをサービス アプライアンスに割り当て、ファブリック (リーフ) のみをプログラミングするように制限されます。デバイスの設定については、データセンターの管理者が外部から実行します。

管理対象および非管理対象の論理デバイスについて

非管理対象モードは、次の XML コードに示すように、論理デバイス (LDevVip) に managed 設定を導入します。

```
<!-- Specified if the device is a managed device-->
<property name="managed"
  type="scalar:Bool"
  owner="management"
  mod="explicit">
  <default value="true"/>
</property>
```

デバイスは管理対象にも、非管理対象にもできます。デバイスを管理対象として設定すると、Application Policy Infrastructure Controller (APIC) はそのデバイスを管理してグラフのインスタンス化時にプログラミングします。デフォルトで、デバイスは APIC への登録時に管理対象モードに設定されます。

デバイスを非管理対象として設定した場合、つまり managed 設定を false に設定すると、APIC はデバイスをプログラミングしません。APIC は、ネットワーク リソースを割り当ててファブリック側で VLAN/VXLAN のプログラミングのみを実行します。

次の設定は、デバイスクラスタが非管理対象として設定されている場合は必要はありません。

- デバイス パッケージ
- 論理デバイス (vnsLDevVip) とデバイス (cDev) の接続情報 (管理 IP アドレス、クレデンシャル、およびインバンド接続情報)
- サポートされる機能タイプ (go-through、go-to) に関する情報
- コンテキスト認識に関する情報 (シングル コンテキストかマルチコンテキスト)

この場合も、APIC は論理デバイスおよびデバイスのトポロジ情報 (LIF、CIF) を把握する必要があります。この情報は、APIC がリーフ上で適切なポートをプログラミングするために必要です。また、APIC はこの情報をトラブルシューティング ウィザードに使用することもあります。

さらに、APIC はカプセル化の割り当てに使用する DomP との関係も把握する必要があります。

管理対象および非管理対象の機能ノードについて

非管理対象モードは、次の XML コードに示すように、機能ノード (AbsNode) に managed 設定を導入します。

```
<!-- Specified if the function is using a managed device-->
<property name="managed"
  type="scalar:Bool"
  owner="management"
  mod="explicit">
  <default value="true"/>
</property>
```

機能ノードは管理対象にも、非管理対象にもできます。機能ノードを管理対象として設定すると、その機能ノードは管理対象デバイスを使用できます。Application Policy Infrastructure Controller (APIC) は、グラフのインスタンス化時にデバイスをプログラミングします。デフォルトでは、機能ノードをサービスグラフに追加すると、その機能ノードは管理対象モードで設定されます。

機能ノードを非管理対象として設定した場合、つまり managed 設定を false に設定すると、APIC はパラメータ解決もデバイスのプログラミングも行いません。APIC は、ネットワークリソースを割り当ててファブリック側でVLAN/VXLANのプログラミングのみを実行します。

次の設定は、機能ノードが非管理対象として設定されている場合は必要はありません。

- MFunc の関係
- AbsFuncProfile
- 設定パラメータ (AbsNode またはエンドポイントグループ上)
- サポートされる機能タイプ (go-through、go-to) に関する情報

この場合も、APICは機能ノードのネットワーク情報 (LIF、CIF) を把握する必要があります。この情報は、APIC がリーフ上でネットワークを適切にプログラミングするために必要です。また、APICはこの情報をトラブルシューティングウィザードに使用することもあります。

さらに、次の設定が必要です。

- グラフ インスタンス化時に LDevVip の選択を可能にする LDevCtx
- グラフ インスタンス化時に LIf の選択を可能にする LIfCtx
- LIfCtx 内のブリッジドメイン
- LIfCtx でのルートピアリング
- LIfCtx 内のサブネット

レイヤ4～レイヤ7サービスのエンドポイントグループについて

非管理対象モード機能の一部として、Application Policy Infrastructure Controller (APIC) では、グラフのインスタンス化時にグラフコネクタに使用するエンドポイントグループを指定できます。これにより、グラフ導入のトラブルシューティングが容易になります。APIC は、指定されたレイヤ4～レイヤ7サービスエンドポイントグループを使用してリーフにカプセル化情報をダウンロードします。また、APICはこのエンドポイントグループを使用して仮想デバイスの分散仮想スイッチにポートグループを作成します。さらに、レイヤ4～レイヤ7サービスのエンドポイントグループを使用して、グラフコネクタのエラー情報や統計情報も集約します。

導入されたグラフリソースへの可視性の向上に加えて、レイヤ4～レイヤ7サービスのエンドポイントグループも使用して、特定のグラフインスタンスに使用する静的なカプセル化を指定することもできます。このカプセル化は、複数のグラフインスタンス間でレイヤ4～レイヤ7サービスのエンドポイントグループを共有することによって、複数のグラフインスタンス間で共有することもできます。

グラフコネクタと共にレイヤ4～レイヤ7サービスのエンドポイントをどのように使用できるかを示すXMLコードの例については、[レイヤ4～レイヤ7サービスのエンドポイントグループとコネクタを関連付けるXMLの例（9ページ）](#)を参照してください。

グラフコネクタに対する静的なカプセル化の使用

Application Policy Infrastructure Controller (APIC) は、処理中にさまざまなサービスグラフにカプセル化を割り当てます。一部の使用例では、サービスグラフ内の特定のコネクタに使用するカプセル化を明示的に指定できます。これは静的なカプセル化と呼ばれます。静的なカプセル化は、物理サービスを持つサービスデバイスクラスタがあるサービスグラフコネクタについてのみサポートされます。仮想サービスデバイスがあるサービスデバイスクラスタは、そのサービスデバイスクラスタに関連付けられたVMwareドメインから動的に割り当てられたVLANを使用します。

静的なカプセル化は、レイヤ4～レイヤ7サービスのエンドポイントグループの一部としてカプセル化値を指定することによってグラフコネクタで使用できます。レイヤ4～レイヤ7サービスのエンドポイントで静的なカプセル化の使用法を示すXMLコードの例については、[レイヤ4～レイヤ7サービスのエンドポイントグループで静的なカプセル化を使用するXMLの例（9ページ）](#)を参照してください。

NX-OS スタイルの CLI を使用した物理デバイスの作成

次に、NX-OS スタイルの CLI を使用して物理デバイスを作成する手順の例を示します。

ステップ1 コンフィギュレーションモードを開始します。

例：

```
apic1# configure
```

ステップ2 テナントのコンフィギュレーションモードを開始します。

```
tenant tenant_name
```

例：

```
apic1(config)# tenant t1
```

ステップ3 クラスタを作成します。

例：

```
apic1(config-tenant)# 1417 cluster name ifav108-asa type physical vlan-domain phyDom5 servicetype FW
```

ステップ4 クラスタデバイスを追加します。

例：

```
apic1(config-cluster)# cluster-device C1
```

ステップ5 プロバイダー クラスタ インターフェイスを追加します。

例 :

```
apicl(config-cluster)# cluster-interface provider
```

ステップ6 インターフェイスにメンバー デバイスを追加します。

例 :

```
apicl(config-cluster-interface)# member device C1 device-interface Pol
apicl(config-member)# interface vpc VPCPolASA leaf 103 104
apicl(config-member)# exit
apicl(config-cluster-interface)# exit
```

ステップ7 コンシューマ クラスタ インターフェイスを追加します。

例 :

```
apicl(config-cluster)# cluster-interface consumer
```

ステップ8 コンシューマ インターフェイスに同じメンバー デバイスを追加します。

例 :

```
apicl(config-cluster-interface)# member device C1 device-interface Pol
apicl(config-member)# interface vpc VPCPolASA leaf 103 104
apicl(config-member)# exit
apicl(config-cluster-interface)# exit
```

ステップ9 クラスタ作成モードを終了します。

例 :

```
apicl(config-cluster)# exit
```

NX-OS スタイルの CLI を使用したハイ アベイラビリティ クラスタの作成

次に、NX-OS スタイルの CLI を使用してハイ アベイラビリティ クラスタを作成する手順の例を示します。

ステップ1 コンフィギュレーション モードを開始します。

例 :

```
apicl# configure
```

ステップ2 テナントのコンフィギュレーション モードを開始します。

```
tenant tenant_name
```

例 :

```
apicl(config)# tenant t1
```

ステップ3 クラスタを作成します。

例 :

```
apic1(config-tenant)# 1417 cluster name ifav108-asa type physical vlan-domain phyDom5 servicetype
FW
```

ステップ4 クラスタ デバイスを追加します。

例 :

```
apic1(config-cluster)# cluster-device C1
apic1(config-cluster)# cluster-device C2
```

ステップ5 プロバイダー クラスタ インターフェイスを追加します。

例 :

```
apic1(config-cluster)# cluster-interface provider vlan 101
```

ステップ6 インターフェイスにメンバー デバイスを追加します。

例 :

```
apic1(config-cluster-interface)# member device C1 device-interface Po1
apic1(config-member)# interface vpc VPCPolASA leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
apic1(config-cluster-interface)# member device C2 device-interface Po2
apic1(config-member)# interface vpc VPCPolASA-2 leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

ステップ7 別のプロバイダー クラスタ インターフェイスを追加します。

例 :

```
apic1(config-cluster)# cluster-interface provider vlan 102
```

ステップ8 最初のインターフェイスからこの新しいインターフェイスに同じメンバー デバイスを追加します。

例 :

```
apic1(config-cluster-interface)# member device C1 device-interface Po1
apic1(config-member)# interface vpc VPCPolASA leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
apic1(config-cluster-interface)# member device C2 device-interface Po2
apic1(config-member)# interface vpc VPCPolASA-2 leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

ステップ9 クラスタ作成モードを終了します。

例 :

```
apic1(config-cluster)# exit
```

NX-OS スタイルの CLI を使用した仮想デバイスの作成

次に、NX-OS スタイルの CLI を使用して仮想デバイスを作成する手順の例を示します。

ステップ 1 コンフィギュレーション モードを開始します。

例：
apicl# **configure**

ステップ 2 テナントのコンフィギュレーション モードを開始します。

tenant *tenant_name*
例：
apicl(config)# **tenant t1**

ステップ 3 クラスタを作成します。

例：
apicl(config-tenant)# **1417 cluster name ifav108-citrix type virtual vlan-domain ACIVswitch servicetype ADC**

ステップ 4 クラスタ デバイスを追加します。

例：
apicl(config-cluster)# **cluster-device D1 vcenter ifav108-vcenter vm NSVPX-ESX**

ステップ 5 コンシューマ クラスタ インターフェイスを追加します。

例：
apicl(config-cluster)# **cluster-interface consumer**

ステップ 6 コンシューマ インターフェイスにメンバー デバイスを追加します。

例：
apicl(config-cluster-interface)# **member device D1 device-interface 1_1**
apicl(config-member)# **interface ethernet 1/45 leaf 102**
ifav108-apicl(config-member)# **vnic "Network adapter 2"**
apicl(config-member)# **exit**
apicl(config-cluster-interface)# **exit**

ステップ 7 プロバイダー クラスタ インターフェイスを追加します。

例：
apicl(config-cluster)# **cluster-interface provider**

ステップ 8 プロバイダー インターフェイスに同じメンバー デバイスを追加します。

例：
apicl(config-cluster-interface)# **member device D1 device-interface 1_1**
apicl(config-member)# **interface ethernet 1/45 leaf 102**
ifav108-apicl(config-member)# **vnic "Network adapter 2"**

```
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

ステップ9 クラスタ作成モードを終了します。

例：

```
apic1(config-cluster)# exit
```

非管理対象モードのXMLの例

以降の項のXMLの例で、非管理対象モードの管理方法を示します。

非管理対象のLDevVipオブジェクトを作成するXMLの例

次に、非管理対象のLDevVipオブジェクトを作成するXMLの例を示します。

```
<polUni>
  <fvTenant name="HA_Tenant1">
    <vnsLDevVip name="ADCCluster1" devtype="VIRTUAL" managed="no">
      <vnsRsALDevToDomP tDn="uni/vmmp-VMware/dom-mininet"/>
    </vnsLDevVip>
  </fvTenant>
</polUni>
```

Cisco ACI Virtual Edge については、次の例のXMLは非管理対象LDevVipオブジェクト (Cisco ACI Virtual Edge VMM ドメインと、スイッチングモードとしてのaveに関連付けられたもの) を作成します:

```
<polUni>
  <fvTenant name="HA_Tenant1">
    <vnsLDevVip name="ADCCluster1" devtype="VIRTUAL" managed="no">
      <vnsRsALDevToDomP switchingMode="AVE" tDn="uni/vmmp-VMware/dom-mininet_ave"/>
    </vnsLDevVip>
  </fvTenant>
</polUni>
```

非管理対象のAbsNodeオブジェクトを作成するXMLの例

次に、非管理対象のAbsNodeオブジェクトを作成するXMLの例を示します。

```
<fvTenant name="HA_Tenant1">
  <vnsAbsGraph name="g1">
    <vnsAbsTermNodeProv name="Input1">
      <vnsAbsTermConn name="C1">
      </vnsAbsTermConn>
    </vnsAbsTermNodeProv>

    <!-- Node1 provides a service function in un-managed mode -->
    <vnsAbsNode name="Node1" managed="no">
      <vnsAbsFuncConn name="outside" >
      </vnsAbsFuncConn>
      <vnsAbsFuncConn name="inside" >
      </vnsAbsFuncConn>
    </vnsAbsNode>
  </vnsAbsGraph>
</fvTenant>
```



```

</vnsAbsNode>

<vnsAbsTermNodeCon name="Output1">
  <vnsAbsTermConn name="C6">
    </vnsAbsTermConn>
  </vnsAbsTermNodeCon>

  <vnsAbsConnection name="CON2" >
    <vnsRsAbsConnectionConns
tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeCon-Output1/AbsTConn"/>
    <vnsRsAbsConnectionConns
tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-outside"/>
  </vnsAbsConnection>

  <vnsAbsConnection name="CON1" >
    <vnsRsAbsConnectionConns
tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-inside"/>
    <vnsRsAbsConnectionConns
tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeProv-Input1/AbsTConn"/>
  </vnsAbsConnection>

</vnsAbsGraph>
</fvTenant>

```

レイヤ4～レイヤ7サービスのエンドポイントグループとコネクタを関連付けるXMLの例

次に、レイヤ4～レイヤ7サービスのエンドポイントグループとコネクタを関連付けるXMLの例を示します。

```

<fvTenant name="HA_Tenant1">
  <vnsLDevCtx ctrctNameOrLbl="any" descr=""
dn="uni/tn-HA_Tenant1/ldevCtx-c-any-g-any-n-any"
graphNameOrLbl="any" name="" nodeNameOrLbl="any">
  <vnsRsLDevCtxToLDev tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1"/>
  <vnsLIfCtx connNameOrLbl="inside" descr="" name="inside">
    <vnsRsLIfCtxToSvcEPg tDn="uni/tn-HA_Tenant1/ap-sap/SvcEPg-EPG1"/>
    <vnsRsLIfCtxToBD tDn="uni/tn-HA_Tenant1/BD-provBD1"/>
    <vnsRsLIfCtxToLIf tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1/lIf-inside"/>
  </vnsLIfCtx>
  <vnsLIfCtx connNameOrLbl="outside" descr="" name="outside">
    <vnsRsLIfCtxToSvcEPg tDn="uni/tn-HA_Tenant1/ap-sap/SvcEPg-EPG2"/>
    <vnsRsLIfCtxToBD tDn="uni/tn-HA_Tenant1/BD-consBD1"/>
    <vnsRsLIfCtxToLIf tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1/lIf-outside"/>
  </vnsLIfCtx>
</vnsLDevCtx>
</fvTenant>

```

レイヤ4～レイヤ7サービスのエンドポイントグループで静的なカプセル化を使用するXMLの例

次に、レイヤ4～レイヤ7サービスのエンドポイントグループで静的なカプセル化を使用するXMLの例を示します。

```

<polUni>
  <fvTenant name="HA_Tenant1">
    <fvAp name="sap">

```

```
<vnsSvcEPg name="EPG1" encap="vlan-3510">
</vnsSvcEPg>
</fvAp>
</fvTenant>
</polUni>
```

非管理対象モードの動作

非管理対象モードについて次の動作が適用されます。

- パラメータ解決と非管理対象機能：非管理対象機能では、Application Policy Infrastructure Controller (APIC) はパラメータ解決を実行しません。AbsGraph、エンドポイントグループ、またはその他のすべてのレベルでパラメータを設定する必要はありません。
- vDev と非管理対象機能：非管理対象機能では、APIC はパラメータ解決やデバイス側のプログラミングを実行しません。非管理対象サービス グラフ機能では、vDev ツリーは作成されません。
- 非管理対象モードでのルート ピ어링：非管理対象モードはルート ピ어링機能に影響しません。
- 非管理対象モードでの VNIC の自動配置：非管理対象モードは VNIC の配置機能に影響しません。