



# Cisco DCNM LAN ファブリックの展開でのテンプレートの使用

templateType	使用するテンプレートのタイプを指定します。	<ul style="list-style-type: none"><li>• CLI</li><li>• ポリシー</li><li>• SHOW</li><li>• プロファイル</li><li>• [抽象 (ABSTRACT) ]</li></ul>
--------------	-----------------------	---

- [ポリシーテンプレート \(1 ページ\)](#)
- [ファブリックのテンプレート \(5 ページ\)](#)
- [プロファイルテンプレート \(5 ページ\)](#)
- [ポリシーの表示、編集、および追加 \(7 ページ\)](#)
- [新しい構成の展開 \(11 ページ\)](#)
- [switch\\_freeform テンプレートの使用 \(11 ページ\)](#)
- [使用中テンプレートのコンテンツの変更, on page 15](#)

## ポリシーテンプレート

ポリシーテンプレートには、CLI と PYTHON の2つのテンプレートコンテンツタイプがあります。CLI コンテンツタイプでは、ポリシーテンプレートはパラメータ化されたCLIテンプレートです。それらは多くの変数とCLIを持つことができます。通常、CLIポリシーテンプレートは小さく、if-else-for などのような構造はありません。AAA サーバー構成のCLIポリシーテンプレートの例を以下に示します。

```

1  ##template variables
2
3  # Copyright (C) 2018 by Cisco Systems, Inc.
4  # All rights reserved.
5
6  #(DisplayName="AAA Server Name/IP", Description="Name or IPv4/IPv6 Address of an AAA Server")
7  ipAddressWithoutPrefix AAA_SERVER;
8
9  #(DisplayName="AAA group", Description="Name of AAA Group")
10 string AAA_GROUP {
11   minLength = 1;
12   maxLength = 127;
13 };
14
15 ##
16 ##template content
17
18 aaa group server radius $AAAA_GROUP$$
19   server $AAAA_SERVER$$
20
21 ##

```

ただし、テンプレートコンテンツタイプ PYTHON のポリシーテンプレートを使用することもできます。基本的に、これにより、複数の CLI ポリシーテンプレートを共通の「送信元」と組み合わせ、一度にすべての適用/適用解除を行うことができます。たとえば、vPC ホストポートを作成する場合、vPC ペアの一部である両方のピアで対称的に作成する必要があります。さらに、ポートチャンネル、メンバーインターフェイス、チャンネルグループなどを作成する必要があります。これが、Python vPC ホストポリシーテンプレートが追加された理由です。ルーテッドインターフェイスを設定するためのインターフェイス PYTHON テンプレートの例を以下に示します。

```

1  ##template variables
2
3  # Copyright (c) 2018 by Cisco Systems, Inc.
4  # All rights reserved.
5
6  @(IsInternal=true)
7  string SERIAL_NUMBER;
8
9  @(PrimaryAssociation=true, IsInternal=true)
10 interface INTF_NAME;
11
12 @(IsMandatory=false, DisplayName="Interface VRF", Description="Interface VRF name, default VRF if not specified")
13 - string INTF_VRF {
14     minLength = 1;
15     maxLength = 32;
16 };
17
18 @(IsMandatory=false, DisplayName="Interface IP", Description="IP address of the interface")
19 ipV4Address IP;
20
21 @(IsMandatory="IP!=null", DisplayName="IP Netmask Length", Description="IP netmask length used with the IP address (Min:1, Max:31)")
22 - integer PREFIX {
23     min = 1;
24     max = 31;
25 };
26
27 @(IsMandatory=false, DisplayName="Routing TAG", Description="Routing tag associated with interface IP")
28 string ROUTING_TAG;
29
30 @(DisplayName="MTU", IsMTU=true, Description="MTU for the interface (Min:576, Max:9216)")
31 - integer MTU {
32     min = 576;
33     max = 9216;
34     defaultValue=9216;
35 };
36
37 @(DisplayName="SPEED", Description="Interface Speed")
38 - enum SPEED {
39     validValues=Auto,100Mb,1Gb,10Gb,25Gb,40Gb,100Gb;
40     defaultValue=Auto;
41 };
42
43 @(IsMandatory=false, DisplayName="Interface Description", Description="Add description to the interface (Max Size 254)")
44 - string DESC {
45     minLength = 1;
46     maxLength = 254;
47 };
48
49 @(IsMandatory=false, IsMultilineString=true, DisplayName="Freeform Config", Description="Additional CLI for the interface")
50 string CONF;
51
52 @(DisplayName="Enable Interface", Description="Uncheck to disable the interface")
53 - boolean ADMIN_STATE {
54     defaultValue=true;
55 };
56
57 ##
58 ##template content
59
60 from com.cisco.dcbu.vinci.rest.services.jython import PTIWrapper
61 from com.cisco.dcbu.vinci.rest.services.jython import Wrapper
62 from com.cisco.dcbu.vinci.rest.services.jython import WrappersResp
63 from utility import *
64
65 def add():
66     try:
67         if CONF != "":
68             respObj, conf = Util.adjustIntfFreeformConfig(SERIAL_NUMBER, INTF_NAME, CONF)
69             if respObj.isRetCodeFailure():
70                 return respObj
71
72             # modify to be done, calling delete now to clean up PTIs before add
73             delete()
74
75             intfVrf = "default"
76         try:
77             if INTF_VRF != "":
78                 intfVrf = INTF_VRF
79         except:
80             Wrapper.print("Switch/Intf = [%s/%s] - Template[int_routed_host_11_1]: INTF_VRF not defined" %
81                           (SERIAL_NUMBER, INTF_NAME))
82             pass
83
84         routingTag = ""
85         try:
86             if ROUTING_TAG != "":
87                 routingTag = ROUTING_TAG
88         except:
89             Wrapper.print("Switch/Intf = [%s/%s] - Template[int_routed_host_11_1]: ROUTING_TAG not defined" %
90                           (SERIAL_NUMBER, INTF_NAME))
91             pass
92
93         # routed_interface has only one CLI command: no switchport
94         # It must be configured before interface_vrf
95         # p2p_routed_interface that configures the IP address must come after interface_vrf
96         Util.exe(PTIWrapper.createOrUpdate(SERIAL_NUMBER, "INTERFACE",
97                                           INTF_NAME, INTF_NAME,
98                                           ConfigPriority.CONFIG_PRIO_INTF,
99                                           "routed_interface",
100                                           ("INTF_NAME": INTF_NAME)))
101
102         if intfVrf != "default":
103             # Create/update PTI for interface VRF
104             Util.exe(PTIWrapper.createOrUpdate(SERIAL_NUMBER, "INTERFACE",
105                                               INTF_NAME, INTF_NAME,
106                                               ConfigPriority.CONFIG_PRIO_INTF_SUB_LVL1,
107                                               "interface_vrf",
108                                               {"INTF_NAME": INTF_NAME, "INTF_VRF": intfVrf}))
109
110         if IP != "":
111             if routingTag == "":
112                 Util.exe(PTIWrapper.createOrUpdate(SERIAL_NUMBER, "INTERFACE",
113                                                   INTF_NAME, INTF_NAME,
114                                                   ConfigPriority.CONFIG_PRIO_INTF_SUB_LVL2,
115                                                   "p2p_routed_interface",
116                                                   ("INTF_NAME": INTF_NAME, "IP": IP, "PREFIX": PREFIX)))

```

各ポリシーテンプレートには、DEVICE、INTERFACEなどのテンプレートサブタイプがあります。これにより、適切なポリシーテンプレートが適切な選択ポイントに表示されます。たと

例えば、[インターフェイス (Interface) ] ウィンドウには、インターフェイス ポリシー テンプレートののみが表示されます。

Name	Supported Platforms	Tags	Template ...	Template ...	Published	Modified T...	D...
csr1kv_loopback	CSR1KV	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
epl_routed_intf	N9K	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
GigabitEthernet	CSR1KV	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
GigabitEthernet_freeform	CSR1KV	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_access_host_11_1	All	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_loopback_11_1	All	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_mgmt_11_1	N9K	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_monitor_ethernet_11_1	N9K	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_monitor_port_channel_11_1	N9K	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_port_channel_access_host_11_1	All	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_port_channel_trunk_host_11_1	All	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_routed_host_11_1	All	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_subif_11_1	All	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_trunk_host_11_1	All	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_vpc_access_host_11_1	All	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	
int_vpc_trunk_host_11_1	All	[interface_...	POLICY	INTERFAC...	false	2019-06-03...	

ファブリック ビルダの [ポリシーの表示/編集 (View/Edit Policies) ] ウィンドウには、デバイス ポリシー テンプレートののみが表示されます。

Name	Supported Platforms	Tags	Template ...	Template ...	Published	Modified T...	D...
aaa_radius	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_radius_deadtime	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_radius_key	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_radius_src_interface	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_radius_use_vrf	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_tacacs	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_tacacs_key	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_tacacs_src_interface	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_tacacs_use_vrf	N9K		POLICY	DEVICE	false	2019-06-03...	
anycast_gateway	N9K		POLICY	DEVICE	false	2019-06-03...	
anycast_rp	N9K		POLICY	DEVICE	false	2019-06-03...	
azure_network_selector	CSR1KV		POLICY	DEVICE	false	2019-06-03...	
banner	N9K		POLICY	DEVICE	false	2019-06-03...	
base_aaa	N9K		POLICY	DEVICE	false	2019-06-03...	
base_bgp	N9K		POLICY	DEVICE	false	2019-06-03...	
base_bgp_external	N9K, N7K		POLICY	DEVICE	false	2019-06-03...	
base_dhcp	N9K		POLICY	DEVICE	false	2019-06-03...	

これらのテンプレートのいずれかをコピーして、必要に応じてカスタマイズできます。これは、カスタマイズの典型的なユースケースです。既存のポリシーを変更せずにコピーを作成し、要件に従ってカスタマイズしてください。そうしないと、DCNMのアップグレード後に変更が失われる可能性があります。

一般に、すでに使用されているテンプレート、つまりファブリック内のスイッチにすでに適用されているテンプレートは編集できません。



- (注) LAN ファブリック インストールモードでは、Type-CLI テンプレートは使用されません。それらはすべて、スーパーセットであるより強力なポリシー テンプレートに置き換えられます。

## ファブリックのテンプレート

ファブリック テンプレートは基本的に python テンプレート、具体的には jython、つまり java + python です。ファブリックテンプレートは非常に包括的であり、ファブリック全体内のすべてのスイッチの目的の構成を生成するために必要なすべてのロジックを含む、ファブリックの展開に必要なルールが組み込まれています。構成は、公開されているシスコのベストプラクティス ガイドラインに基づいて生成されます。組み込みルールに加えて、ファブリック テンプレートは、リソース マネージャ、トポロジデータベース、デバイス ロール、構成コンプライアンスなどの他のエンティティとも統合し、ファブリック内のすべてのデバイスに応じて構成を生成します。これは、DCNM ファブリック ビルダに固有の部分です。

ユーザーが独自のファブリックテンプレートを作成しないことが望ましいです。DCNMには、Easy ファブリック、外部ファブリック、MSD ファブリック、eBGP ファブリック (DCNM 11.2 で導入) など、すぐに使用できるいくつかのファブリックテンプレートが用意されています。

Name	Supported Platforms	Tags	Template ...	Template ...	Published	Modified T...	D...
Easy_Fabric_11_1	All		FABRIC	NA	false	2019-06-03...	F...
Easy_Fabric_eBGP	All		FABRIC	NA	false	2019-06-03...	F...
External_Fabric_11_1	All		FABRIC	NA	false	2019-06-03...	F...
MSD_Fabric_11_1	All		FABRIC	NA	false	2019-06-03...	F...

## プロフィール テンプレート

プロフィールテンプレートは、オーバーレイ（ネットワークまたは VRF）のプロビジョニングに使用されます。オーバーレイ構成を適用する場合、複数の構成要素を組み合わせる必要があるという考え方です。たとえば、VXLAN EVPN ファブリックの有効なレイヤ 3 ネットワーク構成には、VLAN、SVI、int nve 構成、EVPN ルートターゲットなどが必要です。これらの要素はすべて、いわゆる構成プロフィール (NX-OS コンストラクト) にまとめられます。そして、一度に効果的に適用されます。スイッチ上で、構成プロフィール全体が適用されるか、何も適用されません。このようにして、スイッチにぶら下がったり迷い込んだりする構成が残

されることはありません。リーフまたはボーダーのいずれの種類オーバーレイ構成でも、DCNM はプロファイルテンプレートを使用します。

以下に示すように、タグで区別される 4 種類のプロファイルテンプレートがあります。

- ネットワーク プロファイル (ロール リーフを持つすべてのデバイスに適用)
- ネットワーク拡張プロファイル (ロール「border\*」を持つすべてのデバイスに適用)
- VRF プロファイル (ロール リーフを持つすべてのデバイスに適用)
- VRF 拡張プロファイル (ロール「border\*」を持つすべてのデバイスに適用)

Name	Supported Platforms	Tags	Template ...	Template ...	Published	Modified T...	D...
base_external_router	NSK		PROFILE	NA	false	2019-06-03...	s...
Default_Network_Extension_Universal	All	[networkEx...	PROFILE	VXLAN	false	2019-06-03...	D...
Default_Network_Universal	All	[network]	PROFILE	VXLAN	false	2019-06-03...	D...
Default_VRF_Extension_Universal	All	[vrfExtension]	PROFILE	VXLAN	false	2019-06-03...	D...
Default_VRF_Universal	All	[vrf]	PROFILE	VXLAN	false	2019-06-03...	D...
ext_base_setup	All	[borderBase]	PROFILE	VXLAN	false	2019-06-03...	
ext_fabric_intf	All		PROFILE	VXLAN	false	2019-06-03...	
ext_fabric_multisite_intf_11_1	All		PROFILE	VXLAN	false	2019-06-03...	
ext_multisite_overlay_setup_11_1	All	[multiSiteO...	PROFILE	VXLAN	false	2019-06-03...	
ext_multisite_rs_base_feature	NSK,N7K	[multiSiteO...	PROFILE	VXLAN	false	2019-06-03...	s...
ext_multisite_rs_base_setup	NSK	[multiSiteO...	PROFILE	VXLAN	false	2019-06-03...	s...

DCNM のネットワークと VRF ワークフローを介してオーバーレイ構成を適用する方法の詳細については、「ネットワークと VRF の作成と展開」セクションを参照してください。

### 補足事項

ポリシーまたはプロファイルテンプレートが適用されると、テンプレートのアプリケーションごとにインスタンスが作成されます。これに使用される一般的な用語は、ポリシーテンプレートインスタンスまたは PTI です。PTI は、実質的にポリシーまたはプロファイルテンプレート + 置換後の特定のインスタンスを与える名前と値のペアです。デバイス用に作成された PTI は、ファブリックビルダのそのデバイスの [ポリシーの表示/編集 (View/Edit policies)] オプションで表示できます。表形式のビューでは、[ポリシーの表示/編集 (View/Edit policies)] ボタンを使用して、ファブリック全体のデバイスのサブセット全体でポリシーの選択と一括作成/削除を行うことができます。詳細については、「ポリシーの表示と編集」セクションを参照してください。

## ポリシーの表示、編集、および追加

[ポリシーの表示/編集 (View/Edit Policies)] ウィンドウに移動するには、[ファブリックビルダ (Fabric Builder)] ウィンドウでデバイスを右クリックし、[ポリシーの表示/編集 (View/edit policies)] を選択します。



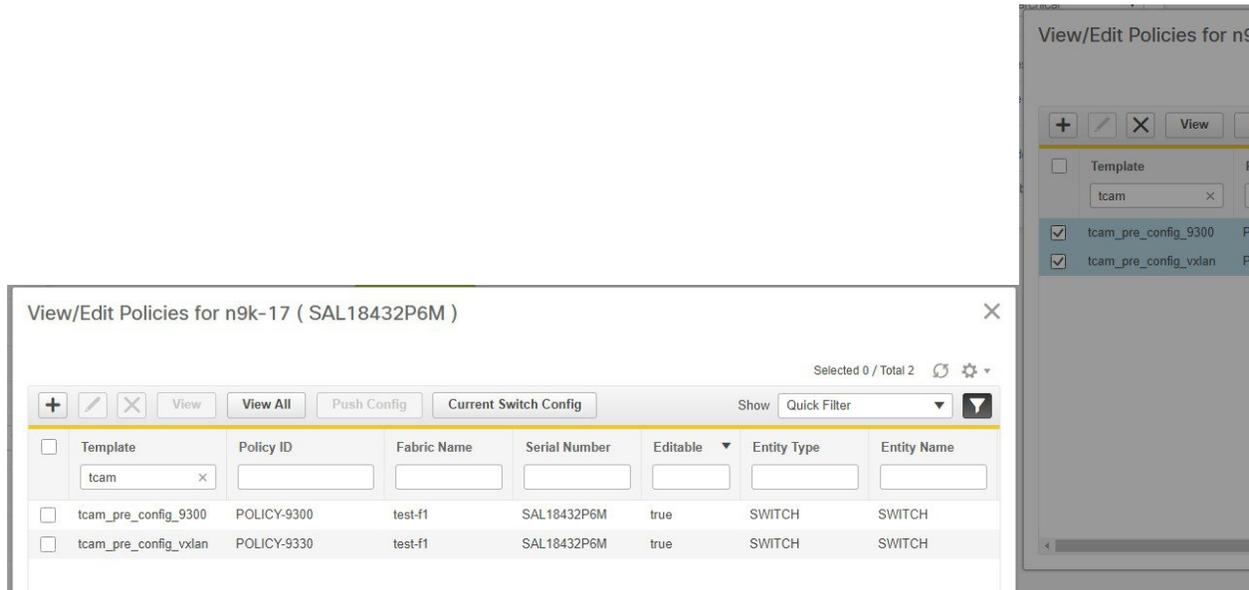
[ポリシーの表示/編集 (View/Edit Policies)] ウィンドウを使用して、デバイスのポリシーを表示、編集、または作成できます。インターフェイスポリシーは表示のみが可能で、[ポリシーの表示/編集 (View/Edit Policies)] ウィンドウから編集/作成することはできないことに注意してください。インターフェイスは、[インターフェイス (Interfaces)] ウィンドウからのみ編集、作成、または削除できます。

## ポリシーの表示

デバイスの特定のポリシーを表示するには、各フィールドの下にある空のボックスに検索条件を指定することにより、フィルタを使用できます。ポリシーが見つかったら、複数のポリシーを選択して [表示 (View)] ボタンをクリックすると、コンテンツを表示できます。以下は、フィルタの使用方法和、ポリシーインスタンスに関連付けられた構成を表示する方法を示す例です。

## 例：デバイスのポリシーの表示

検索フィールドに **tcam** と入力してテンプレートをフィルタリングし、表示するテンプレートを選択し、[表示 (View)] ボタンをクリックして、デバイス用に作成された TCAM ポリシーを表示します。



View/Edit Policies for n9k-17 (SAL18432P6M)

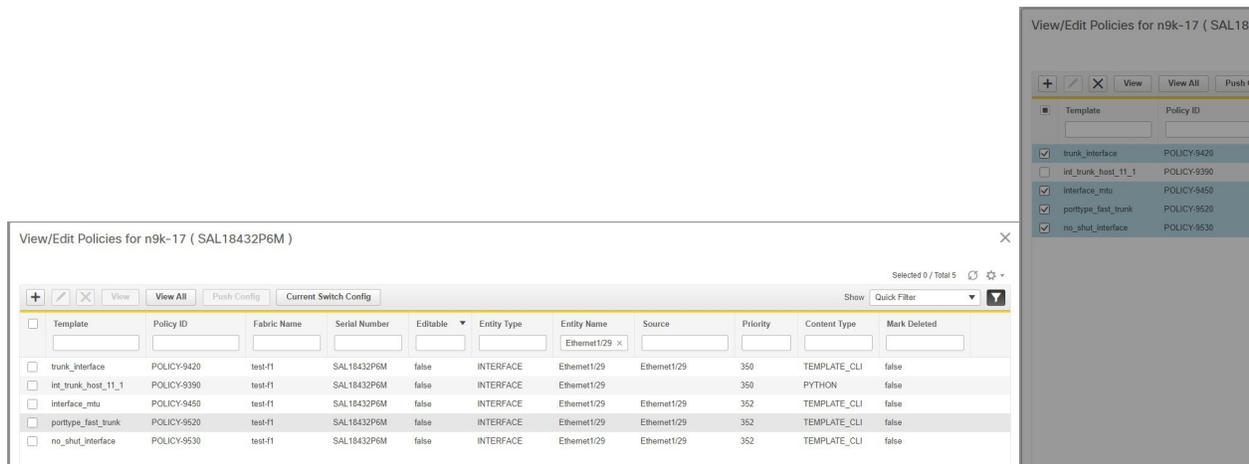
Selected 0 / Total 2

Buttons: +, View, View All, Push Config, Current Switch Config, Show, Quick Filter

Template	Policy ID	Fabric Name	Serial Number	Editable	Entity Type	Entity Name
tcam						
<input type="checkbox"/> tcam_pre_config_9300	POLICY-9300	test-f1	SAL18432P6M	true	SWITCH	SWITCH
<input type="checkbox"/> tcam_pre_config_vxlan	POLICY-9330	test-f1	SAL18432P6M	true	SWITCH	SWITCH

## 例：インターフェイスのポリシーの表示

エンティティ名の下にある検索フィールドにインターフェイス名を入力して、インターフェイスをフィルタ処理します。インターフェイスを選択し、[表示 (View)] ボタンをクリックして、インターフェイス用に作成されたポリシーを表示します。



View/Edit Policies for n9k-17 (SAL18432P6M)

Selected 0 / Total 5

Buttons: +, View, View All, Push Config, Current Switch Config, Show, Quick Filter

Template	Policy ID	Fabric Name	Serial Number	Editable	Entity Type	Entity Name	Source	Priority	Content Type	Mark Deleted
						Ethernet1/29				
<input type="checkbox"/> trunk_interface	POLICY-9420	test-f1	SAL18432P6M	false	INTERFACE	Ethernet1/29	Ethernet1/29	350	TEMPLATE_CLI	false
<input type="checkbox"/> int_trunk_host_11_1	POLICY-9390	test-f1	SAL18432P6M	false	INTERFACE	Ethernet1/29		350	PYTHON	false
<input type="checkbox"/> interface_mtu	POLICY-9450	test-f1	SAL18432P6M	false	INTERFACE	Ethernet1/29	Ethernet1/29	352	TEMPLATE_CLI	false
<input type="checkbox"/> porttype_fast_trunk	POLICY-9520	test-f1	SAL18432P6M	false	INTERFACE	Ethernet1/29	Ethernet1/29	352	TEMPLATE_CLI	false
<input type="checkbox"/> no_shut_interface	POLICY-9530	test-f1	SAL18432P6M	false	INTERFACE	Ethernet1/29	Ethernet1/29	352	TEMPLATE_CLI	false



- (注)
- 各インターフェイスは、1つのインターフェイス `lython` ポリシー テンプレートに関連付ける必要があります。
  - インターフェイス `lython` ポリシー テンプレートには、そのコンテンツに CLI が含まれていませんが、CLI ポリシー テンプレートの PTI が作成されます。これらすべての PTI が組み合わせられて、インターフェイスに関連付けられた完全な構成が生成されます。

## ポリシーの編集

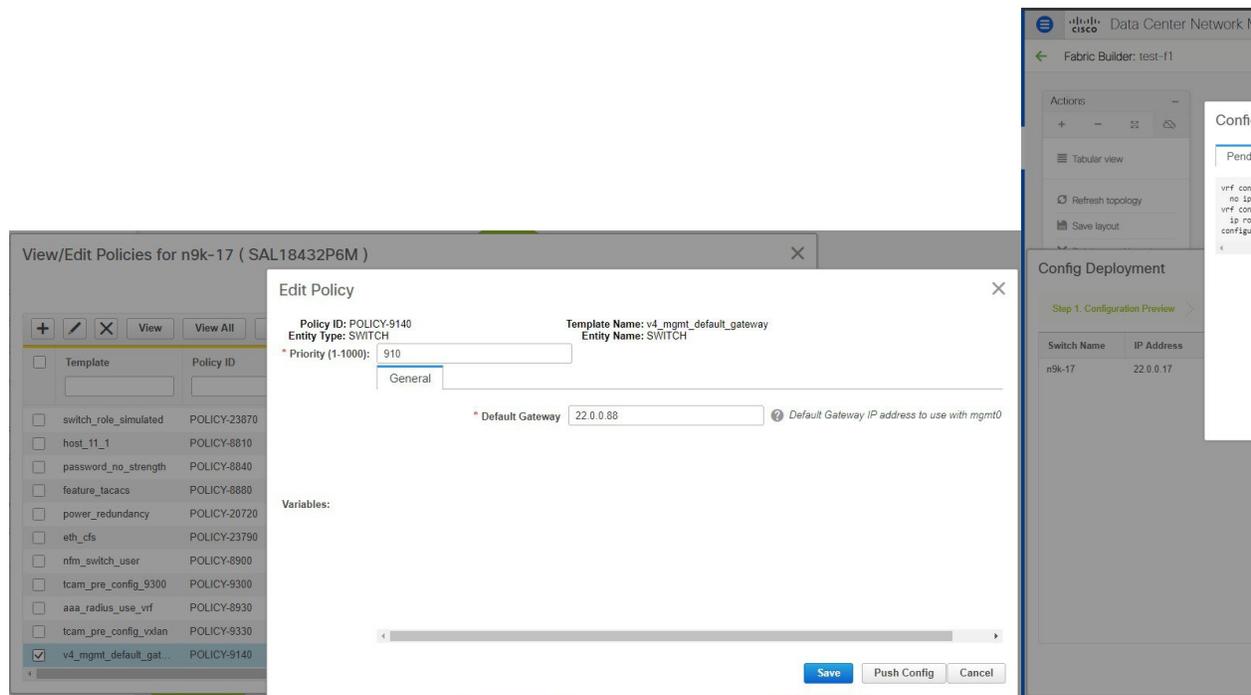
すべてのデバイスポリシーを [ポリシーの表示/編集 (View/Edit policies)] ウィンドウから編集できるわけではありません。空の送信元で作成され、フラグ `Editable = true` を持つポリシーのみを編集できます。

### Procedure

- ステップ 1** デバイスポリシーを編集するには、既存のポリシーを選択し、編集または [鉛筆 (Pencil)] ボタンをクリックします。[ポリシーの編集 (Edit Policy)] ウィンドウが開きます。
- ステップ 2** 1つ以上の名前と値のペアを変更したら、[保存 (Save)] ボタンを押して [ポリシーの編集 (Edit Policy)] ウィンドウで変更を保存します。
- ステップ 3** 変更した構成を展開するには、[ファブリックビルダ (Fabric Builder)] ウィンドウに戻り、デバイスを右クリックして [構成の展開 (Deploy Config)] を選択します。  
これにより、構成コンプライアンスが呼び出され、デバイスの保留中の構成が生成されます。保留中の構成は、スイッチの現在の構成と新しいインテント構成の差分です。
- ステップ 4** 保留中の構成が正しい場合は、[構成の展開 (Deploy Config)] をクリックして保留中の構成をスイッチにプッシュします。

### 例：ポリシーの編集

この例は、IPv4 管理デフォルトゲートウェイを変更する方法を示しています。



## ポリシーの追加

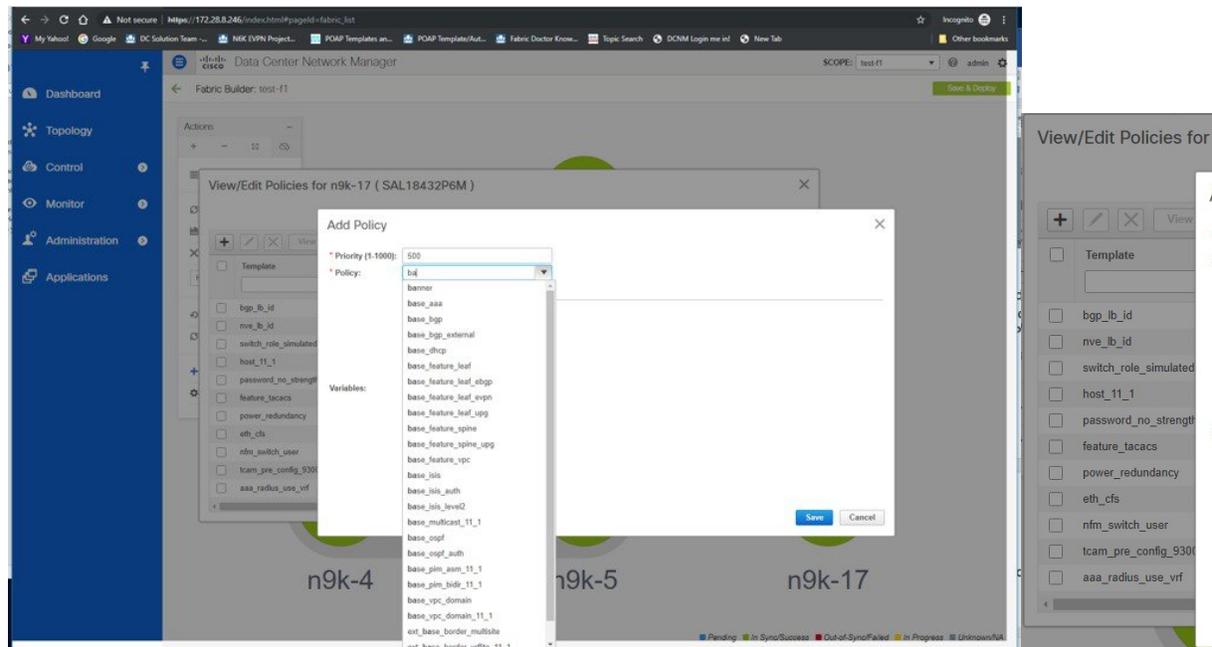
### Procedure

- ステップ 1** ポリシーをデバイスに追加するには、[ポリシーの表示/編集 (View/Edit Policies)] ページで [ + ] ボタンをクリックします。
- [ポリシーの追加 (Add Policy)] ウィンドウが開きます。
- ステップ 2** [ポリシー (Policy)] ドロップダウンリストから、デバイスに追加するポリシーを選択します。
- ステップ 3** ポリシーの優先順位を設定し、必須フィールドに入力します。
- ステップ 4** [保存 (Save)] ボタンをクリックして保存し、ポリシーの追加を完了します。

**Note** ポリシーの優先順位は、構成がスイッチに適用される順序を決定するために使用されます。望ましい構成またはインテントでは、優先順位の低い PTI が優先順位の高い PTI の前に配置されます。これが、展開モジュールを介して構成がプッシュされる順序になります。デフォルトの優先順位は 500 です。

### バナー ポリシーの追加

この例は、バナー ポリシーをデバイスに追加する方法を示しています。



## 新しい構成の展開

新しい構成を展開するには、次の2つの方法があります。

1. [ファブリックビルダ (Fabric Builder)] ウィンドウに移動し、デバイスを右クリックして [構成の展開 (Deploy Config)] を選択します (この方法が推奨されています)。
2. [ポリシーの表示/編集 (View/Edit Policies)] ウィンドウで、新しく追加したポリシーを選択し、[表示 (View)] をクリックして構成を確認します。新しい構成が適切に見える場合は、[構成のプッシュ (Push Config)] ボタンをクリックして、新しい構成をデバイスにプッシュします。[構成のプッシュ (Push Config)] は構成コンプライアンスをバイパスすることに注意してください。このオプションは、新しいユーザまたはSNMPユーザをスイッチに追加する必要がある場合などの例外シナリオにのみ使用してください。

## switch\_freeform テンプレートの使用

[switch\_freeform] は、ユーザーがデバイスに任意の自由形式構成を指定できるようにする特別なポリシーテンプレートです。テンプレートの使用方法は次のとおりです。

- [スイッチ自由形式構成 (Switch Freeform Config)] パラメータでスイッチレベルの構成を指定します。
- 指定された構成は、大文字と小文字と改行に関して [show run] 出力と一致する必要があります。不一致があると、展開中に予期しない差分が発生します。
- 指定された構成に対して、内部の [switch\_freeform\_config] CLI ポリシーが作成されます。

例 : `switch_freeform` ポリシーの作成

- 現在、SVI インターフェイスはインターフェイス ページで構成できないため、SVI インターフェイス以外のインターフェイス構成にはこのテンプレートを使用しないでください。
- ユーザーは、さまざまな構成に対して多くの `[switch_freeform]` ポリシーを作成できます。
- `[switch_freeform]` PTI は、ポリシーの優先順位に基づいて他の PTI と一緒に並べ替えられます。
- `[switch_freeform]` ポリシーは、構成の展開前または展開後に編集できます。
- 構成コンテンツに変更がある場合、以前に作成された内部 `[switch_freeform_config]` ポリシー優先順位が正の数から負の数に変更され、新しい構成に対して新しい内部ポリシーが作成されます。
- [負 (negative) ] の優先順位 PTI は、PTI 内の CLI を削除する必要があることを意味します。[構成コンプライアンス (Configuration Compliance) ] は、それに応じて `[no]` コマンドを生成します。
- `[switch_freeform]` ポリシーを削除すると、その内部ポリシーの PTI 優先順位が負の数に変更されます。

次のセクションでは、`[switch_freeform]` ポリシーを作成し、ポリシーを展開し、その後、更新されたポリシーを編集して再展開する方法を示します。

## 例 : `switch_freeform` ポリシーの作成

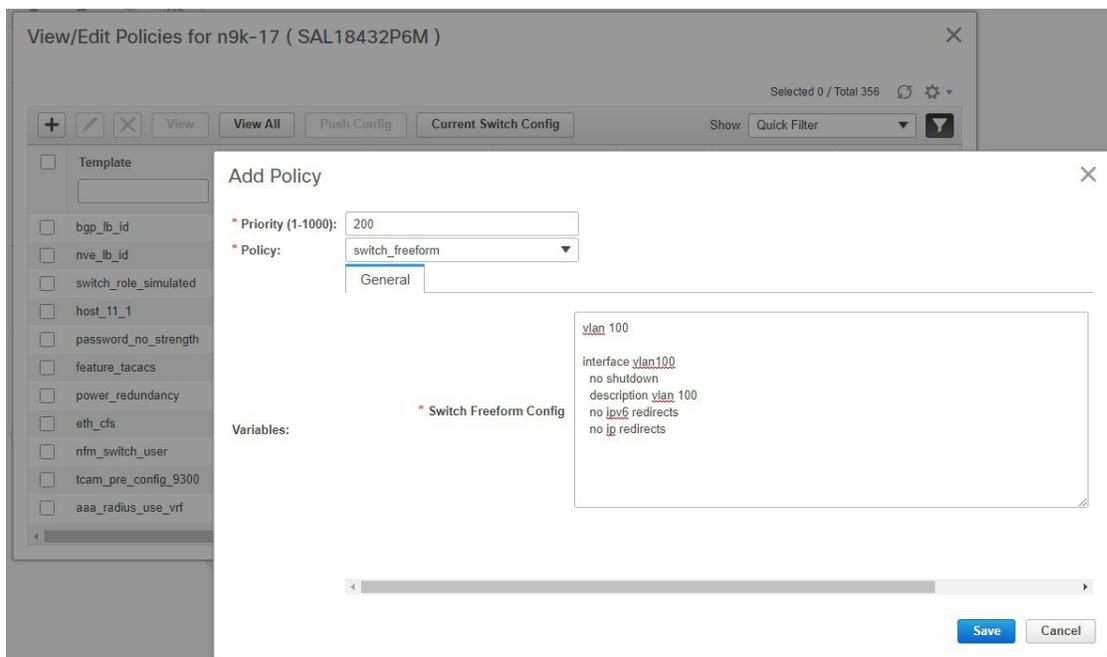
`[switch_freeform]` ポリシーを作成するには、次の手順を実行します。

### Procedure

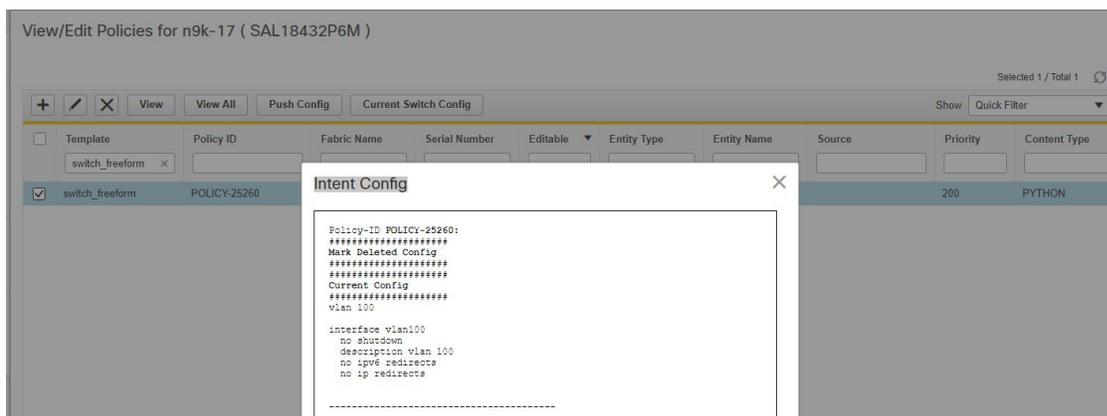
---

**ステップ 1** [ポリシーの追加 (Add Policy) ] 画面のポリシー リストから、`[switch_freeform]` テンプレートを選択します。

優先順位を設定し、自由形式構成を切り替えます。ポリシーを保存します。



ステップ2 [switch\_freeform] ポリシーのインテント構成を表示します。

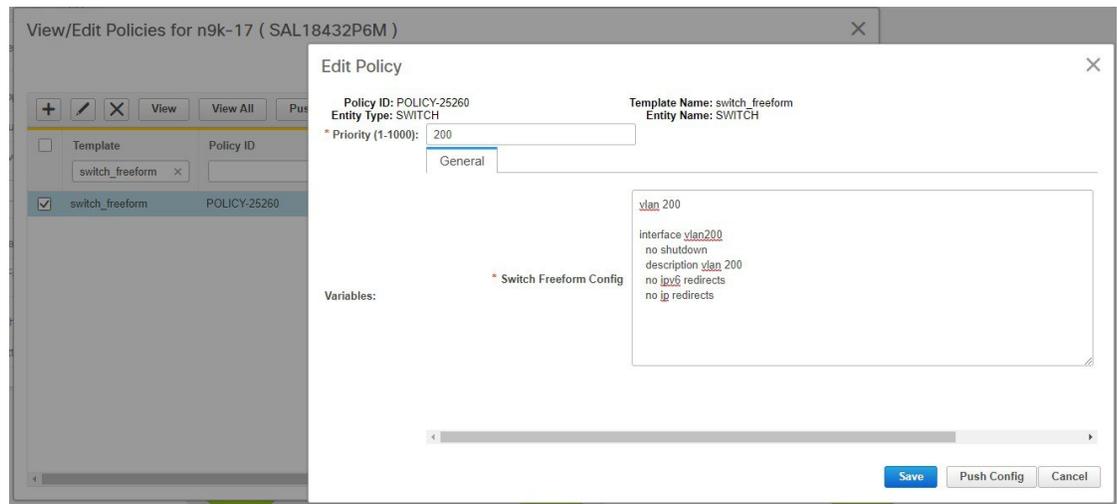


ステップ3 ファブリック ビルダから switch\_freeform ポリシーを展開します。

ステップ4 [ポリシーの表示/編集 (View/Edit Policies)] ウィンドウから、switch\_freeform ポリシーを編集します。

構成を変更します。

## 例 : switch\_freeform ポリシーの作成

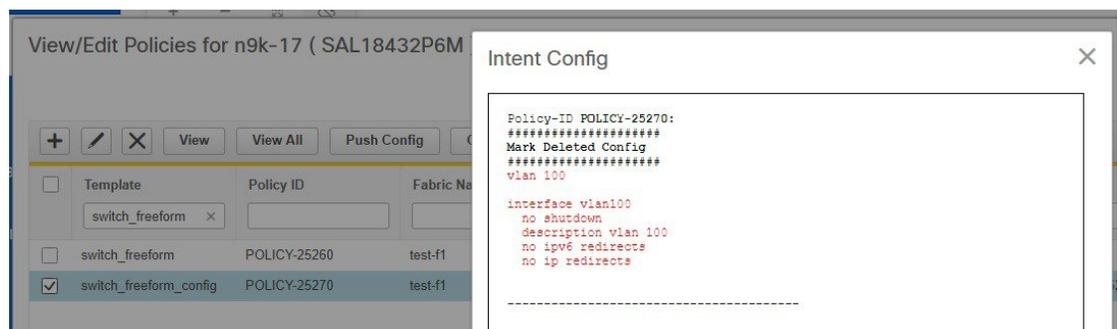


ステップ5 変更を保存します。

以下に示すように、以前に作成された内部 [switch\_freeform\_config] ポリシーの優先順位が負の数 (-200) に変更され、[削除済みとマークする (Mark Deleted)] フラグが true に設定されています。ただし、設計により、新しく作成された内部 [switch\_freeform\_config] ポリシーは表示されません。

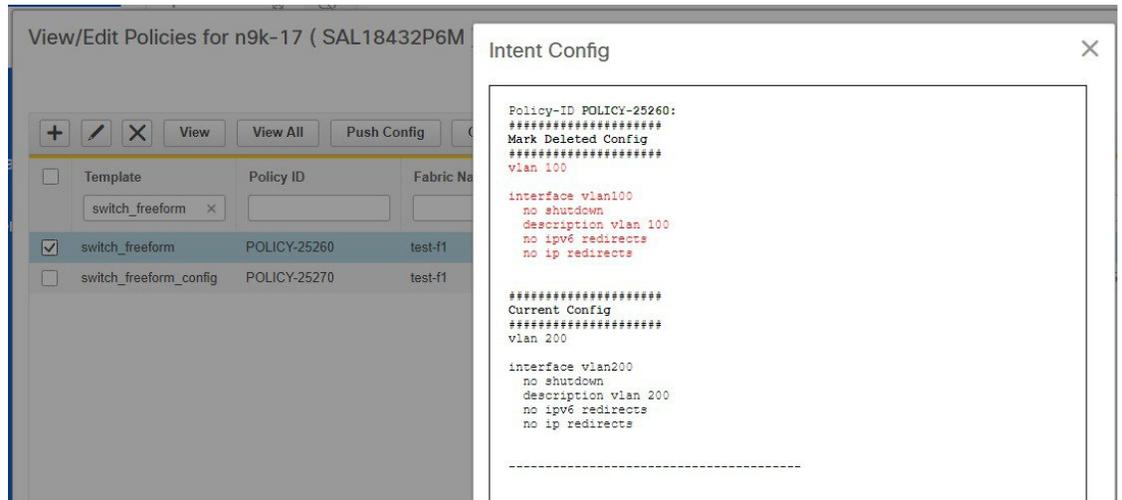
Template	Policy ID	Fabric Name	Serial Number	Editable	Entity Type	Entity Name	Source	Priority	Content Type	Mark Deleted
switch_freeform	POLICY-25260	test-f1	SAL18432P6M	true	SWITCH	SWITCH		200	PYTHON	false
switch_freeform_config	POLICY-25270	test-f1	SAL18432P6M	false	SWITCH	SWITCH	POLICY-25260	-200	TEMPLATE_CLI	true

ステップ6 [削除済みとマークする (mark deleted)] 内部ポリシーのインテント構成を表示します。

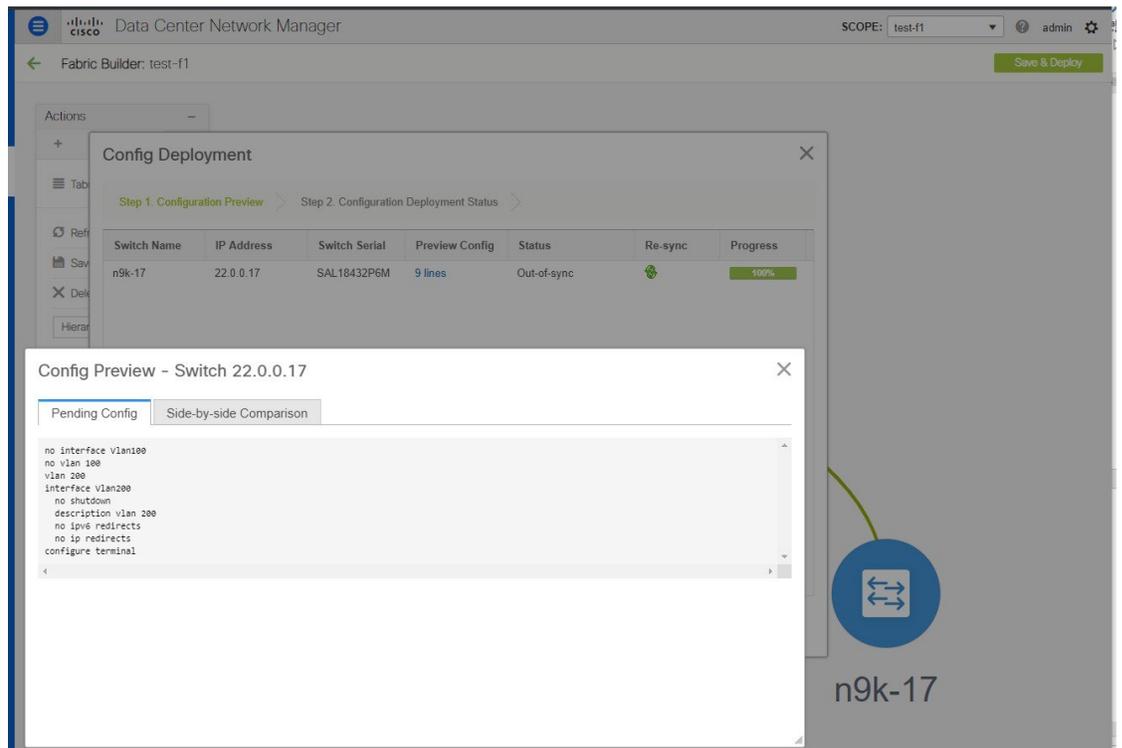


ステップ7 展開の前に、[変更された (changed)] switch\_freeform ポリシーのインテント構成を表示します。

[mark-deleted] 構成と[現在の構成 (current configs)] の両方が表示されることに注意してください。



**ステップ 8** 変更した構成をファブリック ビルダから展開します。



## 使用中テンプレートのコンテンツの変更

一般に、テンプレートは、ポリシー、ファブリック、またはプロファイルテンプレートのいずれであっても、インスタンス化されると変更できません。ただし、テンプレートのバグを修正したり、すでに展開されている構成を変更したりするなど、テンプレートのコンテンツを編集

したい場合もあります。これは、[管理 (Administration)] > [サーバー プロパティ (Server Properties)] タブで [template.in\_use.check] オプションを切り替えることで実現できます。

## Procedure

**ステップ 1** [template.in\_use.check] を [true (デフォルト)] から [false] に変更します。

**ステップ 2** 右上角にある [変更を適用 (Apply Changes)] をクリックします。

DCNM の再起動が必要であることを示す警告がポップアップ表示されます。

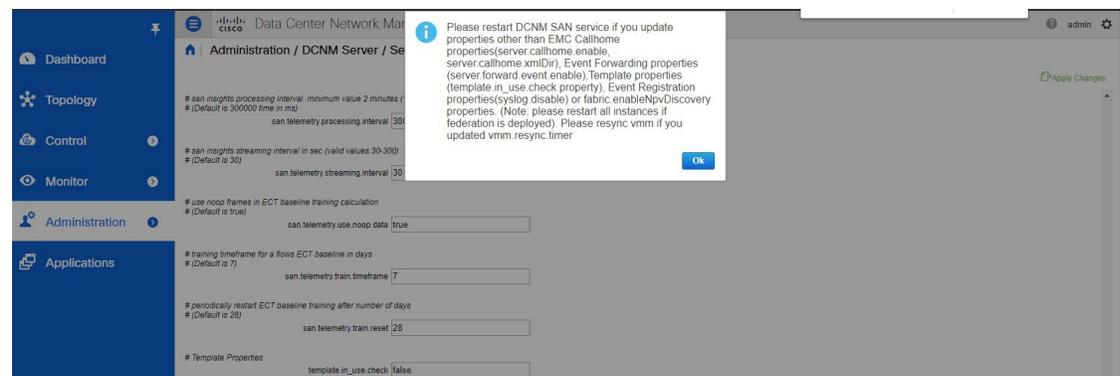
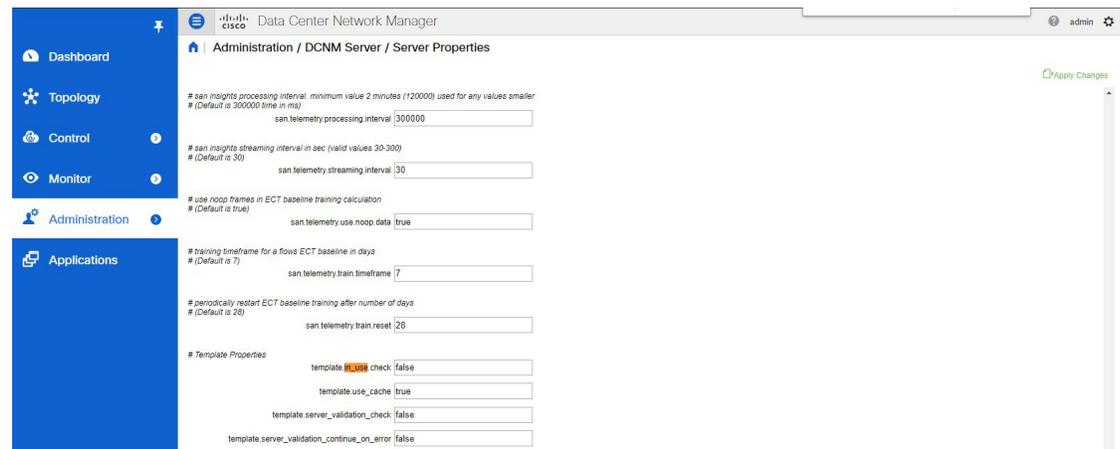
in\_use フラグを有効にするために再起動する必要はないため、この警告は無視してください。

**ステップ 3** 目的のテンプレートを編集します。

**ステップ 4** ファブリック ビルダのページに移動し、ファブリック全体に対して [保存と展開 (Save & Deploy)] をクリックします。

これにより PTI が再生成され、更新されたコンテンツが取得され、望む構成 (またはインテン ト) に使用されます。

**ステップ 5** コンテンツが再生成されて展開されたら、パフォーマンスの問題を回避するために、[template.in\_use.check] を [true] に戻します。



## 翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。