

# 瞭解Catalyst Center上的模板

## 簡介

本文檔介紹Cisco Catalyst Center以及針對三層或摺疊核心園區架構的配置模板的體驗。

## 背景資訊

本文檔面向具備對Cisco Catalyst中心的基礎知識和配置模板相關經驗的企業專業人員。對於已經或計畫使用三層或摺疊核心園區架構的人來說，這一點尤其重要。

主要目標是幫助讀者使用Cisco Catalyst Center中的模板實施並自動化配置和管理解決方案。通過介紹高級見解、實用技術和實際示例，本文檔為那些希望通過自動化和基於模板的管理來增強其LAN基礎架構技能並最佳化工作流程的使用者提供了一個實用資源。

### 執行摘要

隨著企業網路的不斷發展，對可擴展、一致和自動化管理的需求從未像現在這樣迫切。Cisco Catalyst Center提供基於意圖的集中式平台，可簡化整個園區網路的配置、調配和保證。本白皮書探討了網路專業人員如何利用Cisco Catalyst Center的CLI模板編輯器和自動化功能來簡化網路操作、減少配置錯誤，以及加快跨三層核心架構和摺疊核心架構的部署。它詳細介紹了以下方面的最佳做法：設計基於Jinja2的模組化模板，將自動化整合到第0天和第N天工作流程中，以及跨核心、分佈層和接入層實現操作一致性。通過採用本文檔中介紹的策略，您可以將傳統的手動網路管理轉變為與思科的基於意圖的網路願景相一致的靈活、標準化和自動化驅動模式。

## 園區網路面臨的挑戰

隨著園區網路不斷發展以滿足現代組織的需求，它們面臨著以下幾個關鍵挑戰：

### 2a. 網路管理的複雜性

許多網路功能仍由人工管理，增加了人為錯誤的風險。這不僅會增加維護工作量，還會使IT資源緊張，特別是在靜態或有限的預算下。

## 2b. 部署和自動化挑戰

為有線和無線網路安裝新裝置通常既耗時又複雜，導致部署延遲和管理開銷增加。

## 2c. 軟體映像管理

在整個網路中保持一致的「黃金映像」具有挑戰性。許多網路最終會為有線和無線裝置配備多個作業系統，導致效率低下和管理困難。

## 2d. 網路配置不一致

網路配置的變化可能導致合規性問題和操作效率低下，使維護可靠安全的網路更加困難。

## 2e. 使用者期望不斷提高

使用者要求無中斷的連線和無縫的應用體驗，無論其位置或裝置如何。滿足這些期望要求網路具有恢復力、智慧性，並且能夠適應即時變化。

除了這些挑戰外，現代區域網基礎設施還面臨著各種其他複雜性。

## 使用Cisco Catalyst Center簡化園區網路

Cisco Catalyst Center是園區網路的集中式網路管理解決方案，支援總部、分支機構、有線和無線連線以及IT/OT環境。它提供靈活的部署選項，包括物理裝置、VMware ESXi伺服器或AWS雲。Catalyst Center具有全面的功能，可簡化操作、提高效能並增強安全性。

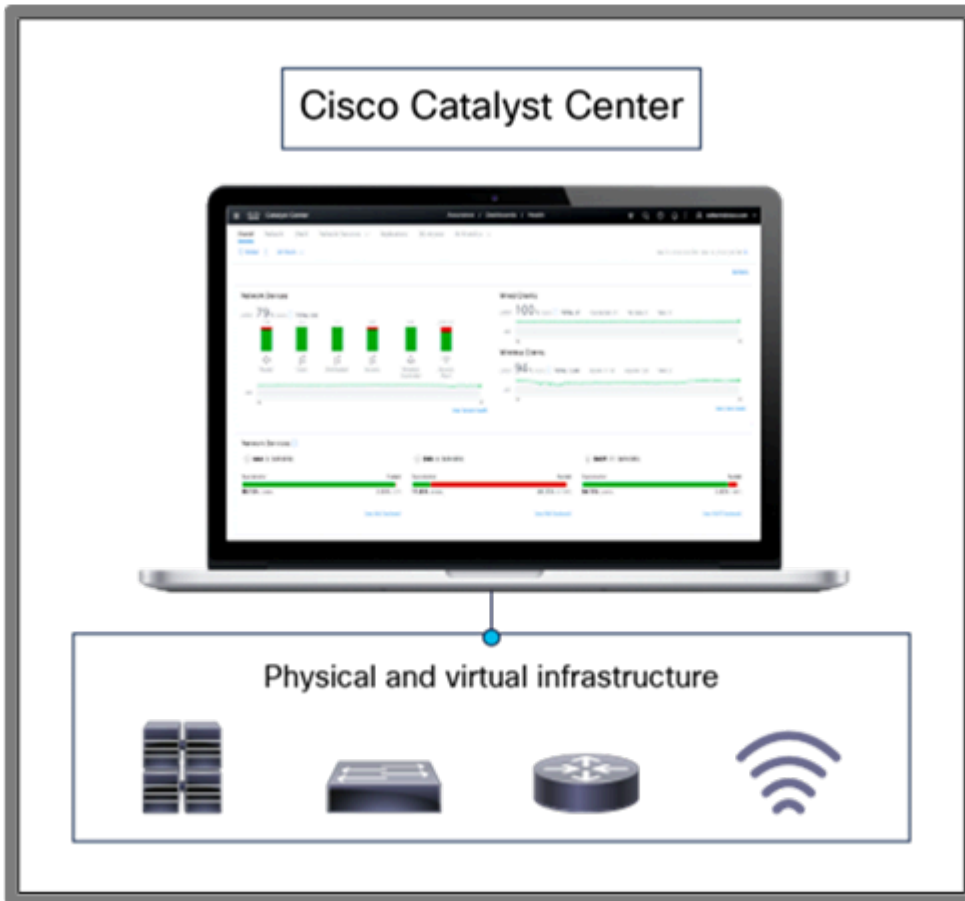


圖 1:使用Cisco Catalyst Center管理基礎設施

## 主要功能和優勢

Cisco Catalyst Center(CC)提供簡化網路管理和自動化的高級功能：

零接觸布建(ZTP):自動註冊裝置，減少手動工作和部署時間。

軟體映像管理(SWIM):通過升級前和升級後檢查確保跨裝置使用一致的軟體版本以防止出現問題。

意圖型自動化:通過將網路意圖轉換為有線和無線網路的裝置配置來簡化部署。

LAN自動化:自動執行第3層IP編址和路由，以建立端到端拓撲。

無線網路自動化:即插即用(PnP)等功能可實現無線接入點的快速調配。

分層網路管理:允許站點特定的配置檔案（例如，SSID、RF引數、VLAN）跨位置進行一致的部署。

CLI模板：Catalyst Center模板編輯器使管理員能夠輕鬆建立和管理基於CLI的配置模板，從而跨裝置實現一致且高效的部署。

保證：保證允許通過CC對受管裝置進行集中監控。

除了這些功能外，Cisco Catalyst Center還提供許多超出本文檔範圍的功能。本文重點介紹如何使用Catalyst Center設計CLI模板。

採用Catalyst Center的LAN園區架構高級概述

傳統LAN園區網路構成了企業連線的骨幹，確保有線和無線裝置可靠且可擴展的通訊。這些網路通常使用3層架構或摺疊核心架構設計，具體取決於組織的規模和複雜性。

### 三層體系結構

三層架構是一個基礎網路設計模型，由核心層、分佈層和接入層組成而成。此架構提供可擴充性、高效能和高效的流量管理。請參見每層的概述。

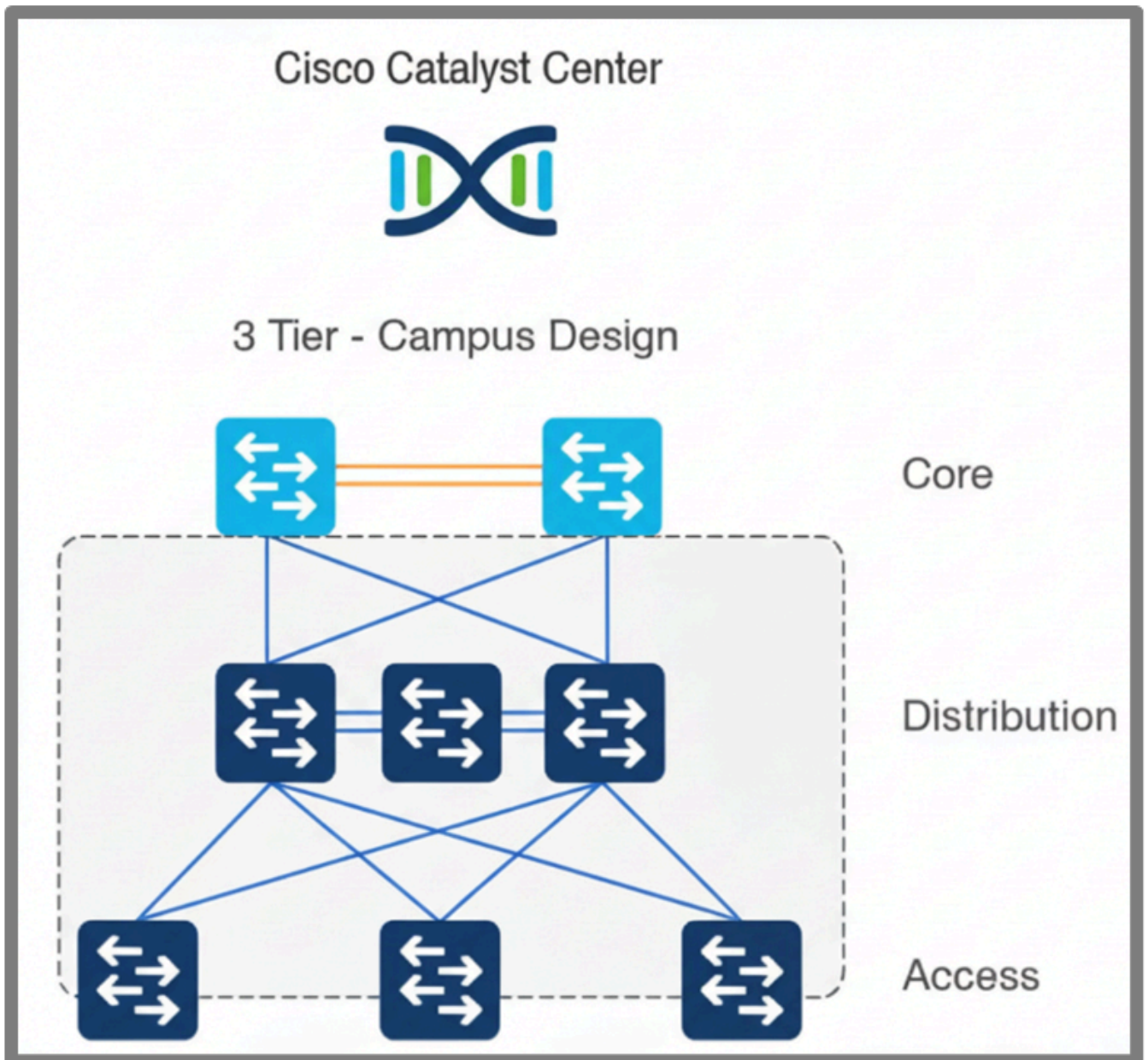


圖 2:三層園區架構

### 核心層

核心層作為網路的骨幹，提供高速連線性和可擴充性。關鍵配置包括北向和南向路由協定（如 OSPF 和 BGP）、路由策略、下行鏈路和上行鏈路介面配置、安全加固等

### 分佈層

分佈層橋接核心層和接入層，處理流量聚合、策略實施和冗餘。關鍵配置包括用於冗餘的 HSRP/VRRP、用於環路預防的 STP、第 2 層和第 3 層 VLAN、上行鏈路和下行鏈路介面配置、用於安全的 ACL 以及安全強化。

### 接入層

接入層將終端連線到網路，實現安全可靠的接入。關鍵配置包括接入介面配置、上行鏈路介面配置、第2層VLAN、用於限制對裝置的訪問的ACL以及安全強化。

## 摺疊核心架構

摺疊式核心體系結構將核心層和分佈層合併到單個層，在保持效能和可擴充性的同時降低了複雜性和成本。這種方法非常適合於不需要單獨核心層的中小型網路。請參閱此架構中的層概述。

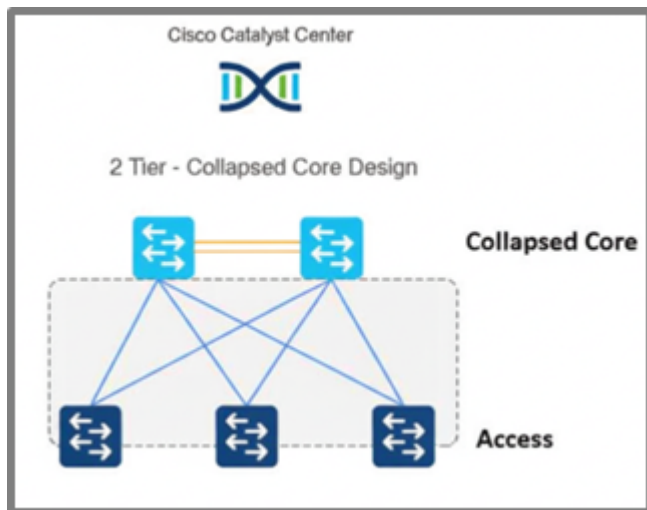


圖 3:摺疊核心園區架構

## 摺疊核心層

摺疊核心層將核心層和分佈層的功能相結合，提供主干連線、流量聚合和策略實施。關鍵配置包括北向和南向路由協定（如OSPF和BGP）、路由策略、下行鏈路和上行鏈路介面配置、用於故障檢測的BFD、使用SVI的VLAN間路由、用於網關冗餘的HSRP/VRRP、用於環路預防的STP以及安全強化。利用Cisco Catalyst Center中的模板，可以自動執行這些配置，確保部署一致且高效。

## 接入層

如前所述，接入層將終端連線到網路，從而實現安全可靠的接入。關鍵配置包括接入介面配置、上行鏈路介面配置、第2層VLAN、用於限制對裝置的訪問的ACL以及安全強化。

## 模板設計注意事項

本節概述如何在Cisco Catalyst Center中設計用於生成裝置配置的模板。模板編輯器通過支援建立可重複使用的CLI模板並支援動態部署為您的網路定製的配置來簡化資源調配。Catalyst Center支援兩種模板語言：Jinja2和Velocity。這些語言有助於裝置的配置管理。

Jinja是一種流行的、設計者友好的模板語言，主要用於Python生成動態內容，如HTML、XML或其他基於文本的格式。它允許在模板內嵌入變數和控制結構（如循環和條件）以建立動態輸出。

Apache Velocity是一個基於Java的模板引擎，它使用Velocity模板語言(VTL)在各種文檔（包括網頁、XML甚至原始碼）中啟用動態內容。它將來自Java對象的資料與模板合併，以生成最終輸出。

本文檔僅涉及Jinja2模板。

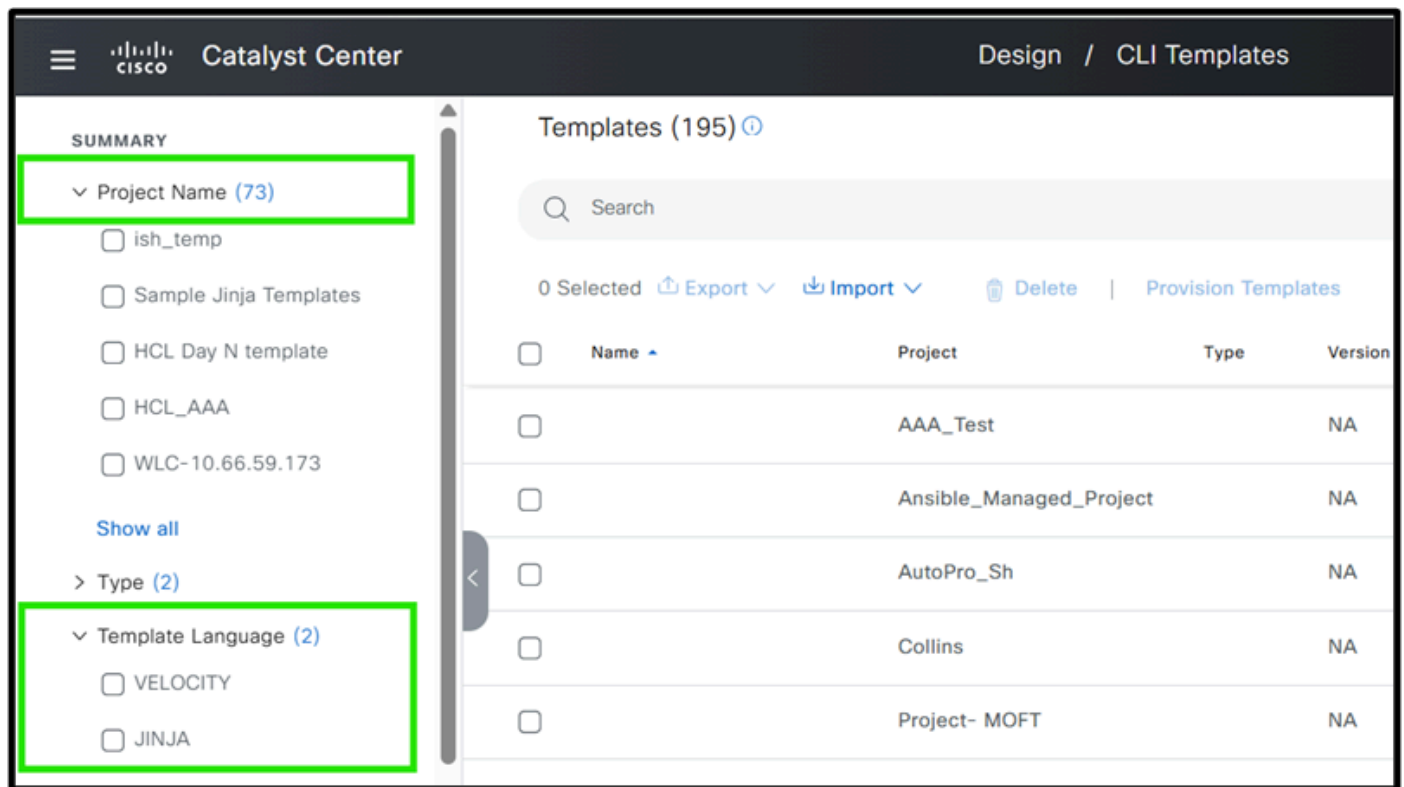


圖 4: Cisco Catalyst中心範本編輯器

在本文中，我們使用Jinja2是因為它的靈活性。重點在於模板設計的實際應用，而不是對Jinja2進行深入的探索。有關Catalyst Center中Jinja2模板的詳細資訊，請參閱連結：

<https://ciscolearning.github.io/cisco-learning-codelabs/posts/cat-center-j2-part-1/#0>

在深入瞭解思科園區網路的模板設計策略之前，使用模板時利用關鍵的最佳實踐來確保效率和可管理性非常重要。

## 模板結構和最佳實踐/最佳策略指南

使用Cisco Catalyst Center自動配置網路裝置時，必須採用結構化策略和最佳實踐。這些步驟有助

於確保整個網路基礎設施的一致性、可擴充性和易管理性。

## 按裝置角色劃分配置

首先根據裝置在網路拓撲中的角色對其進行分類。常見角色包括：

核心

分佈

存取

範例：作為核心交換機的裝置與接入交換機相比，其配置要求必須不同。

## 將配置劃分成模組塊

在每個裝置角色中，通過將相似的功能或配置組合在一起，將配置分解為模組塊。此模組化方法簡化了自動化、故障排除和未來更新。

核心裝置示例：

OSPF配置塊

BGP組態塊

QoS策略塊

## 確定與角色無關的配置塊

某些配置塊在所有裝置角色中普遍適用。確定這些模組並將其標準化，可確保整個網路的最佳實踐和一致性。

常見的角色無關配置塊：

基本配置：主機名、登入標語

管理協定:DHCP、DNS、NTP、SNMP

## 訪問策略：標準安全配置

這些模組可重新用於核心、分發和接入裝置，從而簡化自動化流程。

Use architecture-based configuration segregation to build templates using a modular template methodology		
<p><b>Step1: CLI template project</b> Gives you control to combine similar config and templatzize based on variables</p>	<p><b>Step2: Network Profile</b> Gives you control to map single CLI template to 1 or more sites</p>	<p><b>Step3: Device Tag</b> Control over human error, Ability to mandate review of the tag/config before change</p>
<p><b>Strategy:</b> Use Modular approach to breakdown the configuration by functional area</p>	<p><b>Strategy:</b> Create functional network profile to combine the sites with similar architecture and configuration</p>	<p><b>Strategy:</b> Tag devices only during Change implementation. Remove the tag as soon as change is successful</p>
<p><b>Example:</b></p> <ul style="list-style-type: none"> <li>• Base template for each Core, Distribution, Access devices.</li> <li>• Add on templates for L2/L3, BP, Routing, VLAN, uplinks, etc.</li> <li>• Do not forget to create the tags</li> </ul>	<p><b>Example:</b></p> <ul style="list-style-type: none"> <li>• All sites with 3 Tier Architecture, dual exit routes, similar L2/L3 can be placed under 1 Network profile</li> <li>• All site with Server farm/TOR switch can be in 1 Network profile</li> </ul>	<p><b>Example:</b></p> <ul style="list-style-type: none"> <li>• If New Access switch configurations are needs to be pushed, tag the access switch only during MW.</li> </ul>

圖 1:最佳做法與示例

## Collection of 11 template that can automate entire collapsed core site with 1 single network profile

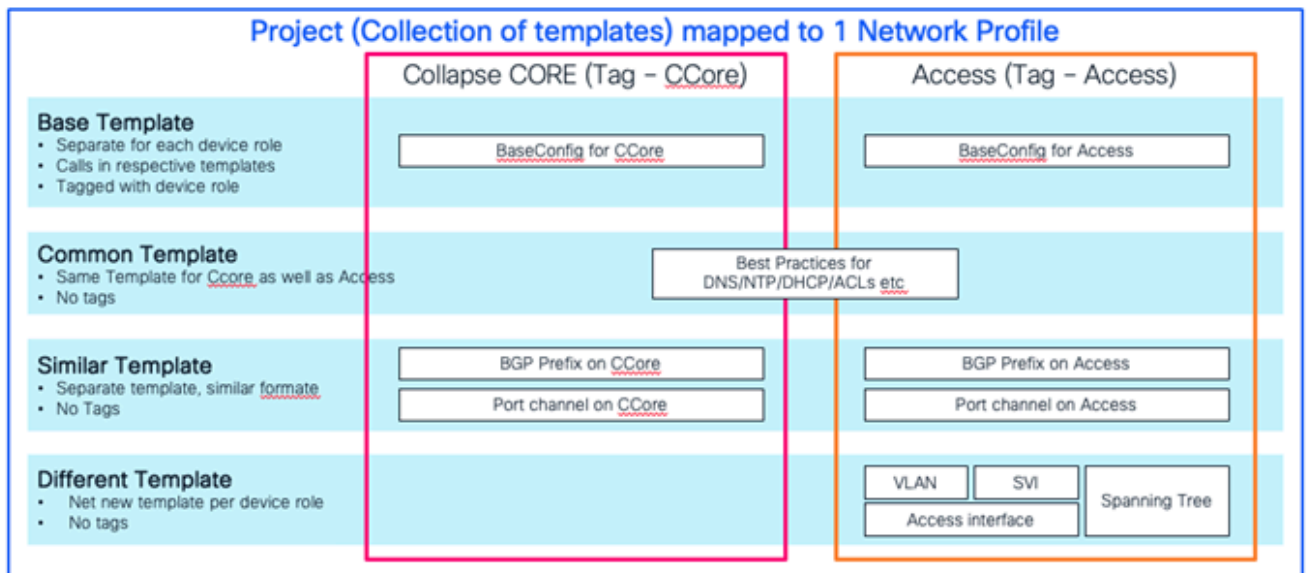


圖 2:摺疊核心模板示例

## 使用模板的最佳實踐

### 用於自動配置的模組化模板設計

在Cisco Catalyst Center中自動執行裝置配置時，請避免將所有配置嵌入到單個單一模板中。相反，應採用模組化方法：

建立引用較小、特定於用途的模板（模組）的基本模板：

將配置分解為邏輯模組（例如，介面設定、路由協定、安全功能）。

此結構使更新更有效 — 對特定模組的更改會在使用該模組的任何位置自動反映出來，從而大大減少了錯誤和複雜性。

範例：分支裝置的模組化配置

假設您要自動配置分支裝置。

基本模板：

包括對關鍵配置區域的模組模板的引用。

根據需要將變數傳遞給每個模組進行自定義。

模組模板：

interface\_settings:管理介面配置。

routing\_protocols:包含OSPF、EIGRP或BGP設定。

security\_features:定義ACL、防火牆規則或其他安全策略。

```
{% include "Branch/Interface Configuration" %}
{% include "Branch/Routing Protocol Configuration" %}
{% include "Branch/Security Configuration" %}

{{ Branch_Interface_Configuration(branch_id) }}
{{ Branch_Routing_Protocol_Configuration(branch_id, ospf_area) }}
{{ Branch_Security_Configuration(branch_id) }}
```

基本模板結構示例：

在此結構中，對路由或安全配置的任何更改只需在各自的模組中進行，而無論使用基本模板在哪裡，這些更改都會立即反映出來。這使您的配置在所有分支路由器上更易於管理且更一致。

這裡的專案名稱為Branch，另外在project下定義了3個不同的模組。所有這些都組合在基本模板中。

### 最小化模板中的變數

將模板中的變數數保持為最小，以減少複雜性和錯誤。更少的變數可簡化部署，尤其是在大型網路中，使流程更高效且一致。

### 對模板使用裝置標籤

利用Cisco Catalyst Center中的裝置標籤（如位置、角色或站點）來建立動態且可擴展的Jinja2模板。這些標籤啟用條件邏輯，確保將正確的配置應用到適當的裝置。這種方法可最大限度地減少錯誤，並簡化不同網路環境中的模板管理。

### 儘可能對靜態值進行硬編碼

對靜態值進行硬編碼可以簡化模板並提高部署效率。常見示例包括DNS、NTP或Syslog伺服器的IP地址，這些地址通常在裝置之間保持一致。同樣，在接入交換機上使用標準VLAN ID可以對這些值進行硬編碼，從而減少可變性並加快部署。

### 採用兩階段方法：第0天和第N天模板

當使用思科即插即用等服務自註冊裝置時，請使用兩階段模板策略：

第0天模板：推送基本配置，確保裝置可以與Cisco Catalyst Center通訊。

N日模板：一旦裝置可訪問，就部署高級功能和配置。

最佳實踐可實現高效且可擴展的模板，從而簡化思科園區網路部署。

### Jinja模板宏中的空白控制元件

使用Jinja語言建立模板時，必須小心處理空格和新行，尤其是在宏中呈現動態內容時。累積的空格或無意中的新行可能會導致生成的輸出中出現格式設定問題，這必定會導致下游處理中的錯誤解釋或錯誤。為了解決這一問題，Jinja提供了控制空白的語法：將減號(-)直接放在分隔符（`{{- ... -}}`或`{%- ... -%}`）中，會刪除表達式周圍的任何前導或尾部空格。例如，將`{{item[1]}}`替

換為`{{- item[1] -}}`可確保宏渲染時刪除任何多餘的空格或換行。這種做法在迭代清單或生成配置檔案時特別有用，如模板片段所示。我們建議始終在這種情況下應用空白控制，以保持乾淨和可預測的輸出。

示例 ( 建議使用 ) :

```
{% for wildcard_list %}中的項
  {% if item[0] == prefix -%}
    {{-item[1] -}}
  {%- endif %}
{% - end for %}
```

## 三層體系結構

本白皮書從開發接入交換機到核心交換機的模板開始，概述了每一層的要求。

### 存取層交換器

接入交換機使用即插即用方式登入，並且必須使用0天模板。有關Catalyst Center中的即插即用過程的詳細資訊，請參閱連結：

[https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/catalyst-center/2-3-7/user\\_guide/b\\_cisco\\_catalyst\\_center\\_user\\_guide\\_237/m\\_onboard-and-provision-devices-with-plug-and-play.html](https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/catalyst-center/2-3-7/user_guide/b_cisco_catalyst_center_user_guide_237/m_onboard-and-provision-devices-with-plug-and-play.html)

如前所述，Catalyst Center支援Velocity和Jinja2模板語言。本文利用Jinja2來說明模板結構，因為它具有靈活性。可使用第0天和第N天模板部署接入層交換機配置。

可以構建基本的Day 0模板，請參閱步驟1:

### 步驟 1:定義模板

```

username admin privilege 15 password SamplePass123
!
enable secret EnableSecret123
!
ip routing
!
vlan {{ branch_number * 100 + 13 }}
 name SW_MGMT
!
interface vlan {{ branch_number * 100 + 13 }}
 ip address {{ ip_address }} 255.255.255.128
 no ip redirects
 no ip unreachable
 no ip proxy-arp
!
ip route 0.0.0.0 0.0.0.0 {{ nexthop }} name Default-Gateway
!
interface range Te1/1/1 - 2
 switchport
 switchport mode trunk
 no shutdown
!

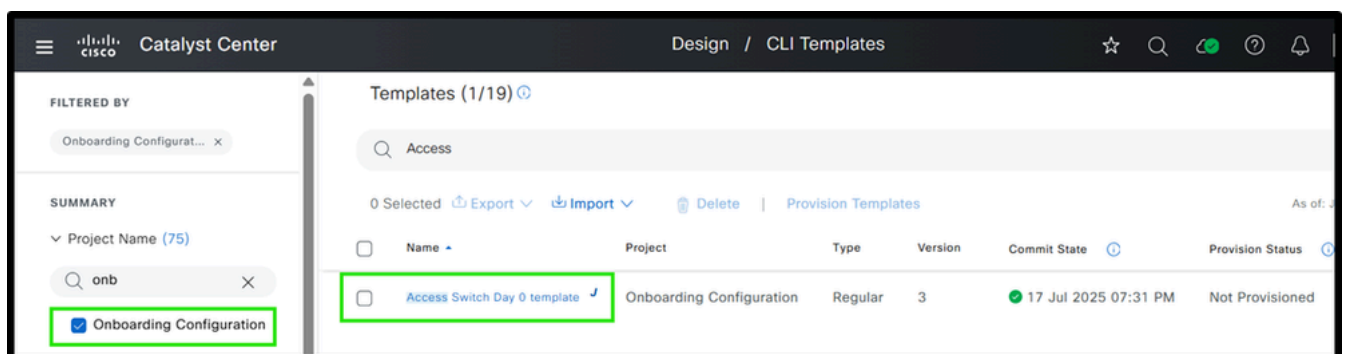
```

步驟 1:定義模板

模板通過硬編碼常數（如使用者名稱、密碼、使能加密和子網掩碼）來簡化配置，因為分支中的所有交換機共用相同的管理VLAN子網掩碼。但是，管理IP地址對於每個交換機都是唯一的，並且被定義為變數。必須在N天模板中提供全面的模板結構，該模板將利用此Day 0模板。在N天模板中，接入交換機的每個功能都由專用模組管理，例如，一個模組處理第2層VLAN，獨立的模組管理上行鏈路和下行鏈路接入介面，另一個模組側重於加強安全性，等等。

雖然應優先使用一致的VLAN ID，但可以使用基於分支編號的公式動態生成不同的ID(例如，分支1 = VLAN 113，分支2 = VLAN 213)。這將使模板在分支之間可重複使用。同樣，下一跳IP是一個變數，因為根據連線的分佈集群，每個分支必須有所不同。

步驟 2:執行模擬並提供變數

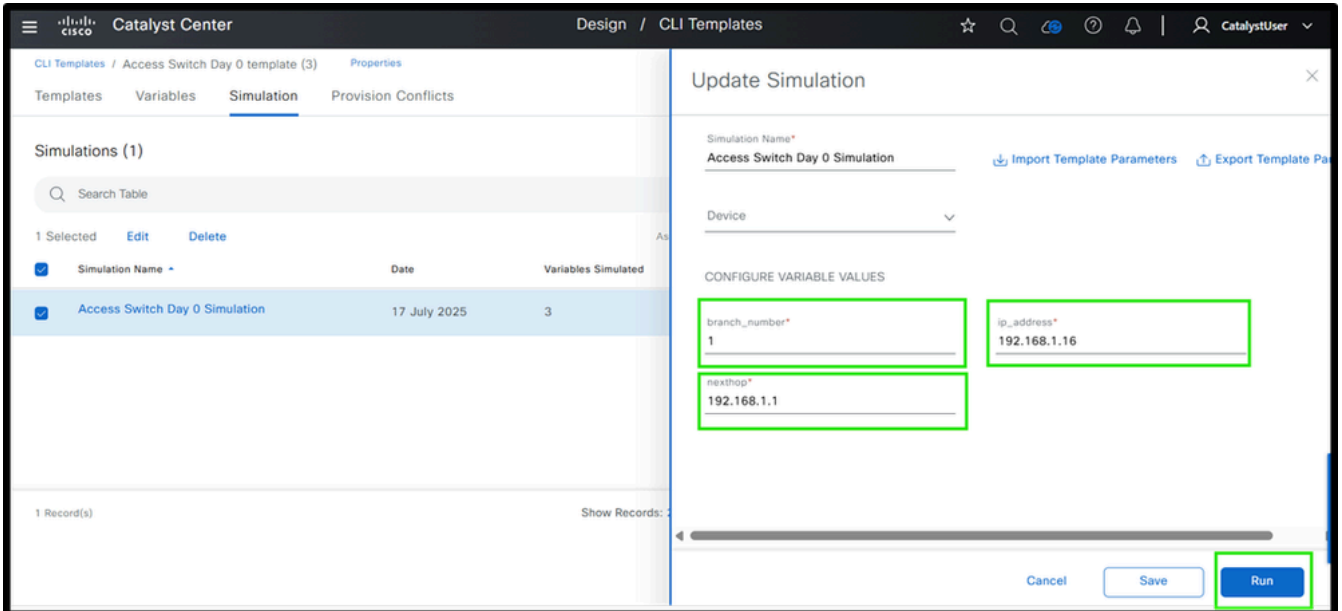


The screenshot shows the Cisco Catalyst Center interface for managing CLI templates. The main area displays a table of templates with the following data:

Name	Project	Type	Version	Commit State	Provision Status
Access Switch Day 0 template	Onboarding Configuration	Regular	3	17 Jul 2025 07:31 PM	Not Provisioned

Additional interface details include a search bar with 'Access', a filter for 'Onboarding Configuration', and a sidebar with 'Project Name (75)' and 'Onboarding Configuration' selected.

具有模擬輸入和輸出的接入交換機Day 0模板結構



例如模擬輸入

始終建議在部署之前模擬模板。螢幕截圖顯示了輸入變數後的最終配置。



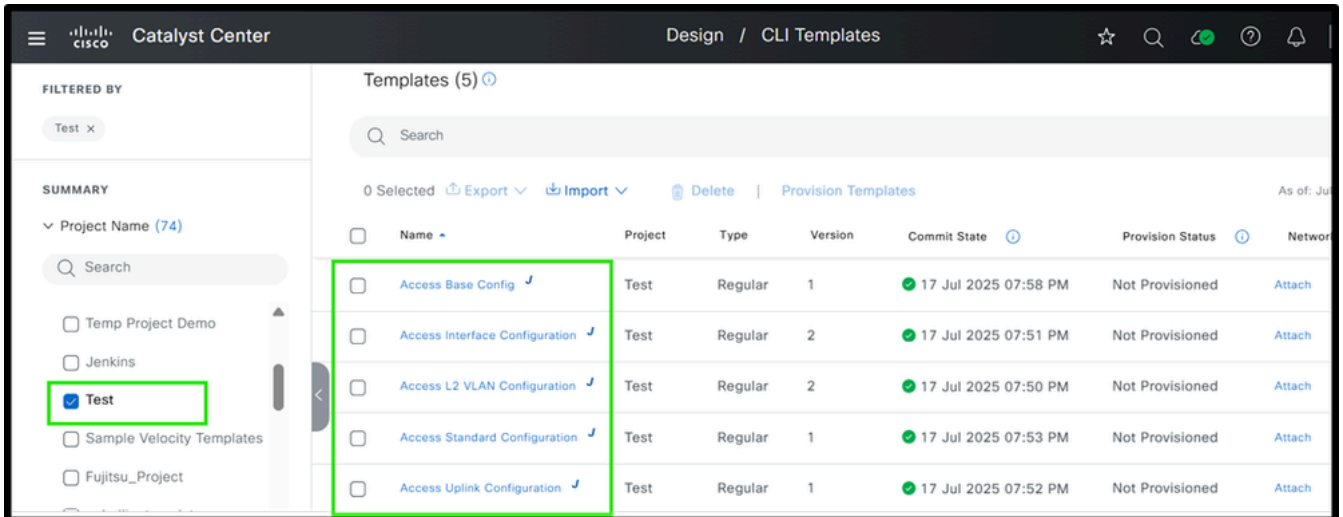
輸入值後的內部配置

現在，讓我們看一下如何建立模組化N日模板。

接入交換機配置可以分為各種模組，所有這些模組都可以組合在一個基本模組中。訪問交換機的基本模板結構如圖所示。

基本模板及其模組均在Cisco Catalyst Center中名為「測試」的專案中建立。

## 步驟 1: 定義各種模板，包括基本模板



訪問交換機第N天模板結構

## 步驟 2: 定義各種模組

Access Base Config :

螢幕截圖顯示了基本配置的示例。

```
{% include "Test/Access L2 VLAN Configuration" %}
{% include "Test/Access Interface Configuration" %}
{% include "Test/Access Uplink Configuration" %}
{% include "Test/Access Standard Configuration" %}

{{ Access_L2_VLAN_Configuration(branch_number, is_poe) }}
{{ Access_Uplink_Configuration(branch_number, is_poe)}}
{{ Access_Interface_Configuration(branch_number, is_poe) }}
{{ Access_Standard_Configuration(branch_number) }}
```

Access Base配置

此模組化配置模板包括四個部分：VLAN配置、上行鏈路介面配置、接入介面配置和標準配置。它僅使用兩個變數：branch\_number和is\_poe，使其易於管理。

branch\_number會計算特定於分支機構的VLAN ID（如0天模板所示），並且is\_poe會確定接入交換機是PoE還是非PoE交換機。這些變數在調配過程中提供，並傳遞給模組以建立正確的配置，從而減少工作量，提高效率。

現在，讓我們檢視每個模組，瞭解它們如何促進生成整體配置的特定部分。

## 訪問L2 VLAN配置

```
{% macro Access_L2_VLAN_Configuration (branch_number, is_poe) %}
!
vlan {{ 100 * branch_number + 11 }}
  name DATA_VLAN
!
vlan {{ 100 * branch_number + 12 }}
  name VOICE_VLAN
!
{% if is_poe == 'Yes' %}
vlan {{ 100 * branch_number + 14 }}
  name AP_Mgmt
{% endif %}
!
{% endmacro %}
```

## 訪問L2 VLAN配置

本模組根據分支編號建立VLAN，如前所述。所有交換機（無論是否支援PoE）都建立了資料和語音VLAN。AP管理VLAN（例如，分支1的114）僅在將is\_poe設定為Yes（是）（表示交換機支援PoE）時建立。如果is\_poe為「否」，則會跳過AP管理VLAN，因為非PoE交換機不能支援接入點。這使用if條件進行管理。

```

{% macro common_access_settings() %}
switchport port-security maximum 2
switchport port-security
switchport port-security violation shutdown
spanning-tree portfast
spanning-tree bpduguard enable
storm-control broadcast level 2.00
storm-control multicast level 2.00
storm-control unknown-unicast 2.00
{% endmacro %}

{% macro Access_Interface_Configuration(branch_number, is_poe) %}
!
interface range Gi1/0/1 - 6
{% if is_poe == 'Yes' %}
description *** AP ***
switchport mode access
switchport access vlan {{ 100 * branch_number + 14 }}
{% else %}
description *** User Ports ***
switchport mode access
switchport access vlan {{ 100 * branch_number + 11 }}
switchport voice vlan {{ 100 * branch_number + 12 }}
{% endif %}
{{ common_access_settings() }}
!
interface range Gi1/0/7 - 24
description *** User Ports ***
switchport mode access
switchport access vlan {{ 100 * branch_number + 11 }}
switchport voice vlan {{ 100 * branch_number + 12 }}
{{ common_access_settings() }}
!
{% endmacro %}

```

存取介面組態

此模組處理訪問介面配置並使用與前面介紹的PoE交換機相同的方法。如果is\_poe變數為Yes，"表示交換機是PoE交換機，則前六個埠(1-6)必須配置有AP管理VLAN。否則，前六個埠必須設定為使用者訪問埠。

假設交換機是24埠型號，則無論交換機是否為PoE，其餘埠(7-24)始終配置為使用者接入埠。

介面範圍已經過標準化，不再作為輸入變數使用，這被認為是使模板中變數數量最小的最佳實踐。此外，該模組包括一個名為common\_access\_settings的宏，該宏通過合併重複配置來最小化模板大小。此宏在介面設定內被簡單呼叫，無需多次指定它們。



附註：此模板對所有訪問介面應用相同的描述。如果每個介面都需要唯一的描述，建議使用單獨的Python指令碼或類似的自動化工具來推送描述。

檢視生成上行鏈路介面配置的模組。

```
{% macro Access_Uplink_Configuration(branch_number, is_poe) %}
{% if is_poe == 'Yes' %}
!
interface range Te 1/1/1 - 2
switchport
switchport mode trunk
switchport trunk allowed vlan {{ branch_number * 100 + 11 }},{{ branch_number * 100 + 12 }},{{ branch_number * 100 + 13 }},{{
branch_number * 100 + 14 }}
no shutdown
!
{% else %}
!
interface range Te 1/1/1 - 2
switchport
switchport mode trunk
switchport trunk allowed vlan {{ branch_number * 100 + 11 }},{{ branch_number * 100 + 12 }},{{ branch_number * 100 + 13 }}
no shutdown
!
{% endif %}
{% endmacro %}
```

接入上行鏈路配置

此模組生成上行鏈路介面的配置並處理VLAN修剪。如果交換器支援PoE，則AP管理VLAN會包括在允許的VLAN清單中；否則會將其排除。如前所述，使用代碼中的if條件來管理此邏輯。

檢視最終模組，該模組演示了標準配置，包括最佳做法和安全強化。



注意：請注意，這僅用於說明目的，不能用作實際網路設定的參考，因為配置可能會因特定要求而異

---

```

{% macro Access_Standard_Configuration (branch_number) %}
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
vtp mode off
no errdisable recovery cause all
crypto key generate rsa modulus 2048
!
ip ssh version 2
ip ssh time-out 120
ip ssh source-interface vlan {{ branch_number * 100 + 13 }}
no ip http server
no ip http secure-server
ip http client source-interface vlan {{ branch_number * 100 + 13 }}
!
logging buffered informational
logging host 192.168.1.10
logging host 192.168.2.20
logging source-interface vlan {{ branch_number * 100 + 13 }}
!
ntp authentication
ntp authentication-key 10 md5 NetwOrkAuthKey
ntp source vlan {{ branch_number * 100 + 13 }}
ntp server 192.168.3.1 key 10
ntp server 192.168.3.2 key 10
!
snmp-server enable traps
snmp-server trap-source vlan {{ branch_number * 100 + 13 }}
snmp-server group NMSNWDEVICE v3 priv access SNMPHOST
snmp-server user netadmin NMSNWDEVICE v3 auth sha AuthKey123 priv aes 128 PrivKey123
!
ip access-list standard SNMPHOST
permit 192.168.4.0 0.0.0.255
!
ip access-list standard VTYACL
permit 192.168.5.10

```

第1部分：訪問標準配置

```

permit 192.168.5.11
!
aaa new-model
ip tacacs source-interface vlan {{ branch_number * 100 + 13 }}
tacacs server TACACS_1
  address ipv4 192.168.6.1
  key TACACSKey123
  timeout 4
tacacs server TACACS_2
  address ipv4 192.168.6.2
  key TACACSKey123
  timeout 4
aaa group server tacacs+ TACACS-SERVER
  server name TACACS_1
  server name TACACS_2
!
aaa authentication login default group TACACS-SERVER local
aaa authorization exec default group TACACS-SERVER local
aaa accounting exec default start-stop group TACACS-SERVER
!
line console 0
login authentication default
exec-timeout 5 0
!
line vty 0 15
login authentication default
access-class VTYACL in
exec-timeout 5 0
!
banner login ^
***** WARNING *****
All systems/network should be used/accessed by authorized persons only
  If you are not authorized to do so, you should log off immediately
  Access to and usage of this system /network may be monitored
  All users must comply with information security policies
  Any Violation may lead to disciplinary action.
*****^
{% endmacro %}

```

## 第2部分：訪問標準配置

此模組生成一個標準配置，該配置結合了最佳實踐、安全加固和安全裝置管理的主要功能。除 `branch_number` 外，大多數值都進行了硬編碼，以確保各分支機構的一致性。`branch_number` 用於計算每個分支機構中交換機的管理 VLAN，並且用作多個配置的源介面。

步驟 3: 在配置交換機之前執行模擬。僅必須模擬基本配置。

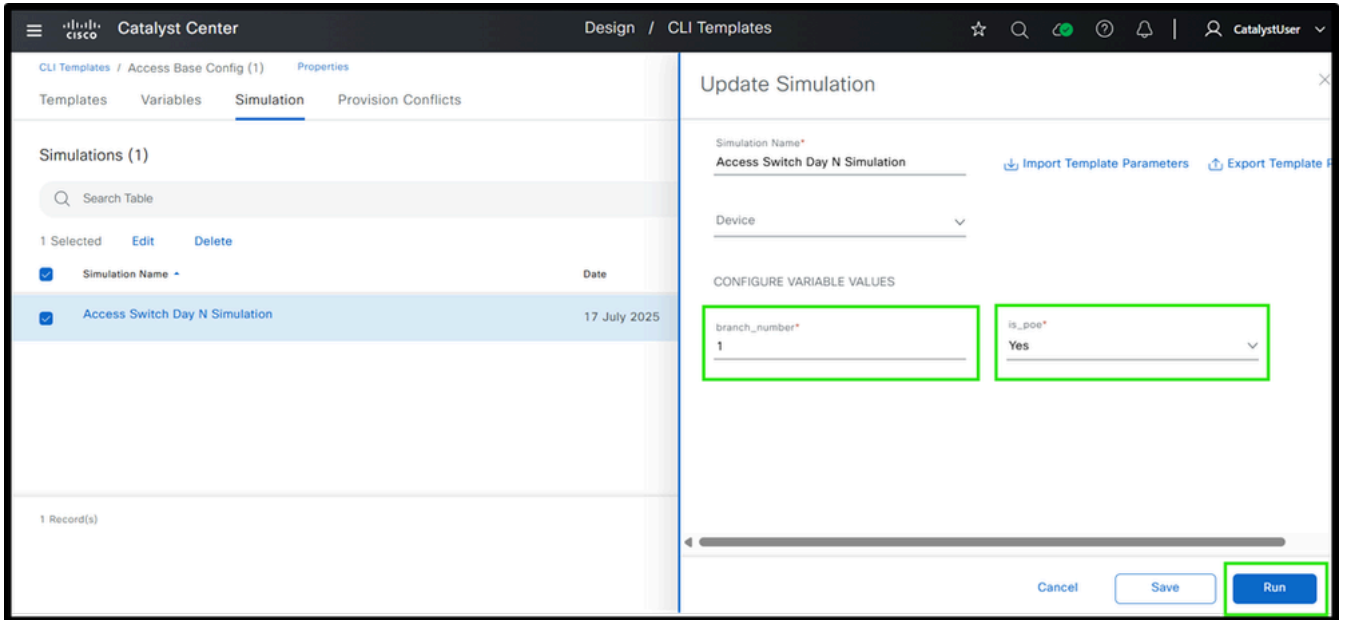


圖 7: 接入交換機第N天模板模擬輸入和輸出

```
Catalyst Center Design / CLI Templates
Simulation - Access Switch Day N Simulation
Variables Simulated: 2 Status:
3 !
4 vlan 111
5   name DATA_VLAN
6 !
7 vlan 112
8   name VOICE_VLAN
9 !
10 vlan 114
11   name AP_Mgmt
12 !
13 !
14 !
15 interface range Te 1/1/1 - 2
16   switchport
17   switchport mode trunk
18   switchport trunk allowed vlan 111,112,113,114
19   no shutdown
20 !
21 !
22 !
23 interface range Gi1/0/1 - 6
24   description *** AP ***
25   switchport mode access
```

```
Catalyst Center Design / CLI Templates
Simulation - Access Switch Day N Simulation
Variables Simulated: 2 Status:
22 !
23 interface range Gi1/0/1 - 6
24   description *** AP ***
25   switchport mode access
26   switchport access vlan 114
27   switchport port-security maximum 2
28   switchport port-security
29   switchport port-security violation shutdown
30   spanning-tree portfast
31   spanning-tree bpduguard enable
32   storm-control broadcast level 2.00
33   storm-control multicast level 2.00
34   storm-control unknown-unicast 2.00
35 !
36 !
37 interface range Gi1/0/7 - 24
38   description *** User Ports ***
39   switchport mode access
40   switchport access vlan 111
41   switchport voice vlan 112
42   switchport port-security maximum 2
43   switchport port-security
44   switchport port-security violation shutdown
```

模擬

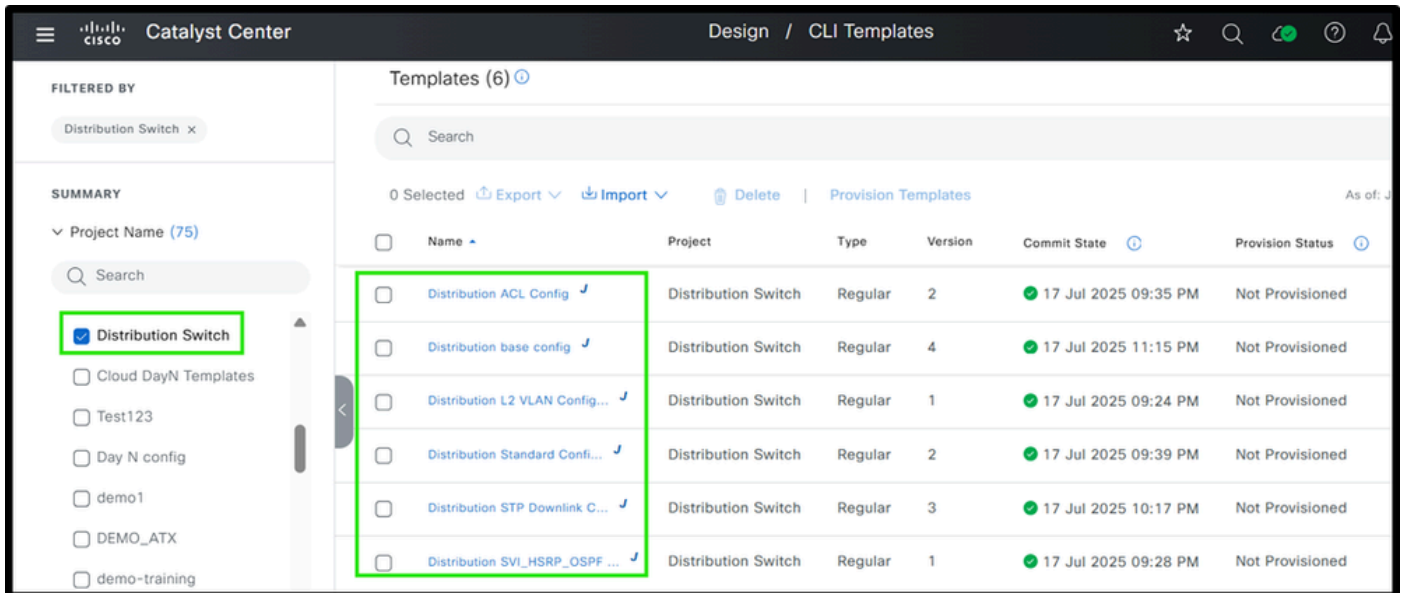
這就是在接入層使用模板生成配置的方法。

現在，讓我們看一下分佈層裝置，看看如何對它們應用模板。

## 分佈層交換器

現在，為分佈層交換機設計一個模組化模板。基本模板及其模組是Cisco Catalyst Center「分佈交換機」專案的一部分。

### 步驟 1: 配電開關模板結構



例如發行模板

## 第2步： 定義每個模組

提供的基本配置定義了每個模組，並引用了所有模組。

```
{% include "Distribution Switch/Distribution L2 VLAN Configuration" %}
{% include "Distribution Switch/Distribution STP Downlink Config" %}
{% include "Distribution Switch/Distribution SVI_HSRP_OSPF Config" %}
{% include "Distribution Switch/Distribution ACL Config" %}
{% include "Distribution Switch/Distribution Standard Configuration" %}

{{ Distribution_L2_VLAN_Configuration(branch_number, is_poe) }}}
{{ Distribution_STP_Downlink_Config(branch_number, is_poe, distribution_number) }}
{{ Distribution_SVI_HSRP_OSPF_Config(branch_number, is_poe, distribution_number) }}
{{ Distribution_ACL_Config(branch_number) }}
{{ Distribution_Standard_Config() }}
```

例如通訊基地範本模組

與接入交換機類似，所有模板都建立在「分佈交換機」項目中，並在基本模板中引用。雖然某些模板與用於接入交換機的模板相同，但本節將說明特定於分佈層交換機的差異。模組「分佈層L2 VLAN配置」與前面介紹的接入交換機模組相同。請檢查提供此資訊的[Access L2 VLAN Configuration](#)模組。它根據為變數提供的輸入值生成所需的VLAN。

現在請檢視分散式STP下行鏈路配置模組，該模組處理分散式交換機的生成樹和上行鏈路配置。

```

{% macro Distribution_STP_Downlink_Config (branch_number, is_poe, distribution_number) %}
!
spanning-tree mode rapid-pvst

{% set base_vlan = branch_number * 100 %}
{% set vlans = [base_vlan + 11, base_vlan + 12, base_vlan + 13] %}

{% if is_poe == 'Yes' %}
    {% set vlans = vlans + [base_vlan + 14] %}
{% endif %}

{% if distribution_number == 1 %}
    {% set stp_priority = 4096 %}
{% else %}
    {% set stp_priority = 8192 %}
{% endif %}

spanning-tree vlan {{ vlans | join(',') }} priority {{ stp_priority }}
!
interface range TWE 1/0/1 - 2
    switchport
    switchport mode trunk
    switchport trunk allowed vlan {{ vlans | join(',') }}
    no shutdown
!
{% endmacro %}

```

分佈STP下行鏈路配置

此處使用了Jinja2宏功能，在基於模組中會引用該功能。這種結構有助於構建模組化方法。

此模組根據branch\_number和交換機是否啟用PoE來配置生成樹協定(STP)和下行鏈路介面。"branch\_number"變數用於為每個分支機構生成唯一的基本VLAN，從而確保不同的VLAN，類似於已為接入交換機突出顯示的方法。如果交換機已啟用PoE("is\_poe" == 'Yes')，則會向清單中新增一個額外的VLAN，例如AP管理VLAN。"distribution\_number"變數確定STP優先順序，為分佈1設定4096（使其成為首選根網橋），為輔助分佈交換機設定8192。最後，將適當的VLAN應用到中繼介面，確保根據交換機是否啟用PoE，僅允許相關的VLAN。

現在請檢視分佈SVI\_HSRP\_OSPF配置模組，該模組重點介紹SVI、HSRP和OSPF的設定，以實現高效的網路路由和冗餘。

```

{% macro Distribution_SVI_HSRP_OSPF_Config (branch_number, is_poe, distribution_number) %}
!
interface loopback0
ip address {{ loopback_ip }} 255.255.255.255
!
router ospf 1
router-id {{ loopback_ip }}
!
key chain HSRP_KEY
key 0
key-string cisco@7875
!
interface vlan {{ 100 * branch_number + 11 }}
description Data_Endpoints
ip address 172.17.{{ (branch_number - 1) * 16 }}.{{ distribution_number + 1 }} 255.255.240.0
standby bfd
standby version 2
standby {{ 100 * branch_number + 11 }} ip 172.17.{{ (branch_number - 1) * 16 }}.1
{% if distribution_number == 1 %}
standby {{ 100 * branch_number + 11 }} priority 255
{% else %}
standby {{ 100 * branch_number + 11 }} priority 250
{% endif %}
standby {{ 100 * branch_number + 11 }} authentication md5 key-chain HSRP_KEY
standby {{ 100 * branch_number + 11 }} preempt delay minimum 120
no ip redirects
no ip unreachable
no ip proxy-arp
ip ospf 1 area 0
bfd interval 100 min_rx 100 multiplier 3
!
! uplink interfaces
interface TWE1/1/1
no switchport
ip address {{ twe1_1_1_ip }} 255.255.255.0
ip ospf 1 area 0
no shutdown
!
interface TWE1/1/2
no switchport
ip address {{ twe1_1_2_ip }} 255.255.255.0
ip ospf 1 area 0
no shutdown
!
{% endmacro %}

```

分佈SVI\_HSRP\_OSPF Config

此模組Distribution\_SVI\_HSRP\_OSPF\_Config可幫助配置分佈交換機的SVI、HSRP、OSPF和上行鏈路介面。在本例中，我們將重點放在資料子網的SVI上，但同樣的方法也可用於其他SVI，如語音或管理。

如果已完成資料子網的IP地址規劃，則可以根據branch\_number和distribution\_number變數為每個SVI自動計算IP地址。例如，如果branch 1的子網為172.17.0.0/20,branch 2為172.17.16.0/20，而branch 3為172.17.32.0/20，則網關IP為172.17.x.1（其中x是分支編號）。第一分佈層交換機的第二個IP是172.17.x.2，第二分佈層交換機的第三個IP是172.17.x.3。這樣可以自動計算IP地址，減少錯誤，簡化流程。

為環回介面分配了變數loopback\_ip的IP，該變數用作OSPF路由器ID，以確保整個網路路由穩定一致。在OSPF配置中，此環回IP用作路由器ID，相關介面將新增到OSPF區域0。對於HSRP，將設定優先順序值：第一台分佈交換機為255，第二台為250，確保正確的故障轉移。此外，HSRP身份驗證使用金鑰鍵(HSRP\_KEY)進行配置以增強安全性。

為了保持配置乾淨且可管理，需要對某些值進行硬編碼。例如，所有分支的子網掩碼(255.255.240.0)和某些HSRP設定（如版本和BFD）相同，從而減少了變數數量。這樣可簡化組態、簡化組態、降低錯誤發生的可能性。最後，使用IP配置上行鏈路介面，並將其新增到OSPF區域0，以便在交換機之間進行正確路由。這種方法使配置過程更易於管理且不易出錯，同時對於不同的分支機構也是靈活的。

現在請檢視「分佈ACL配置」模組，該模組在分佈層提供分段。

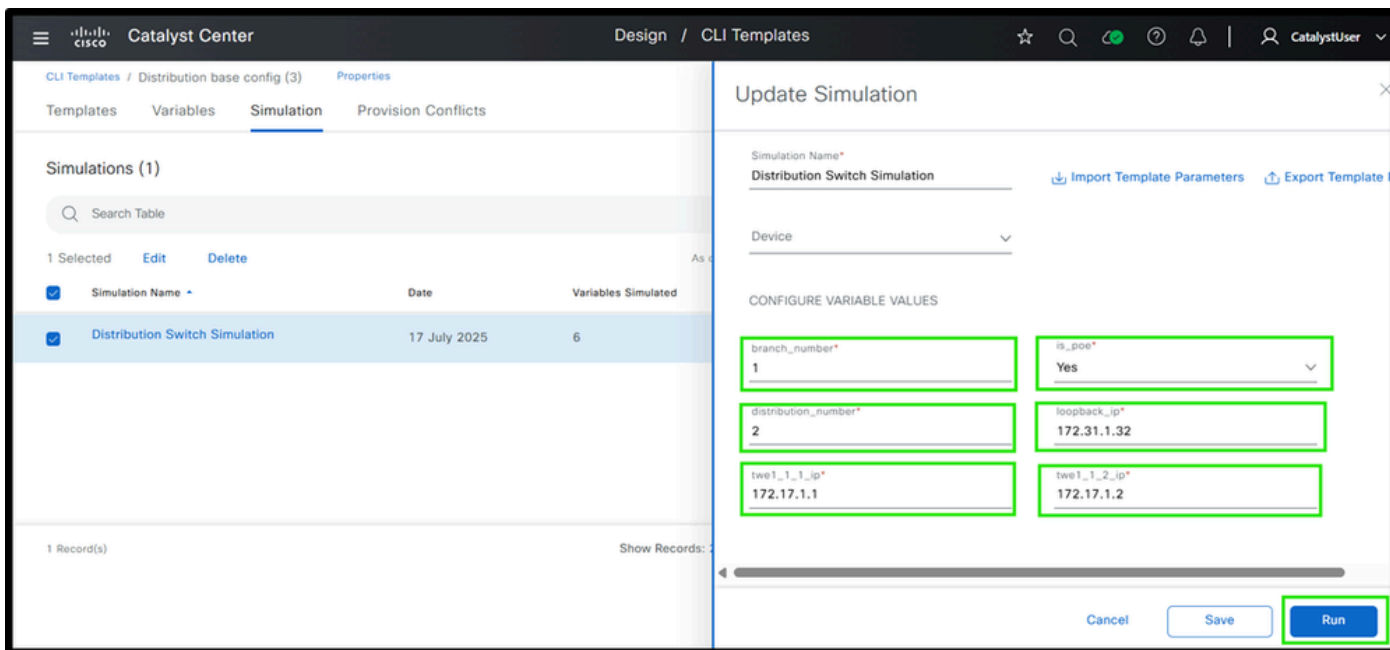
```
{% macro Distribution_ACL_Config (branch_number) %}
!
ip access-list extended BLOCK_BRANCH
deny ip 172.17.{{ 16 * (branch_number - 1) }}.0 0.0.15.255 172.16.{{ 16 * (branch_number - 1) }}.0 0.0.15.255
deny ip any host 239.255.255.250
permit ip any any
!
interface vlan {{ 100 * branch_number + 11 }}
ip access-group BLOCK_BRANCH in
!
{% endmacro %}
```

分佈ACL配置

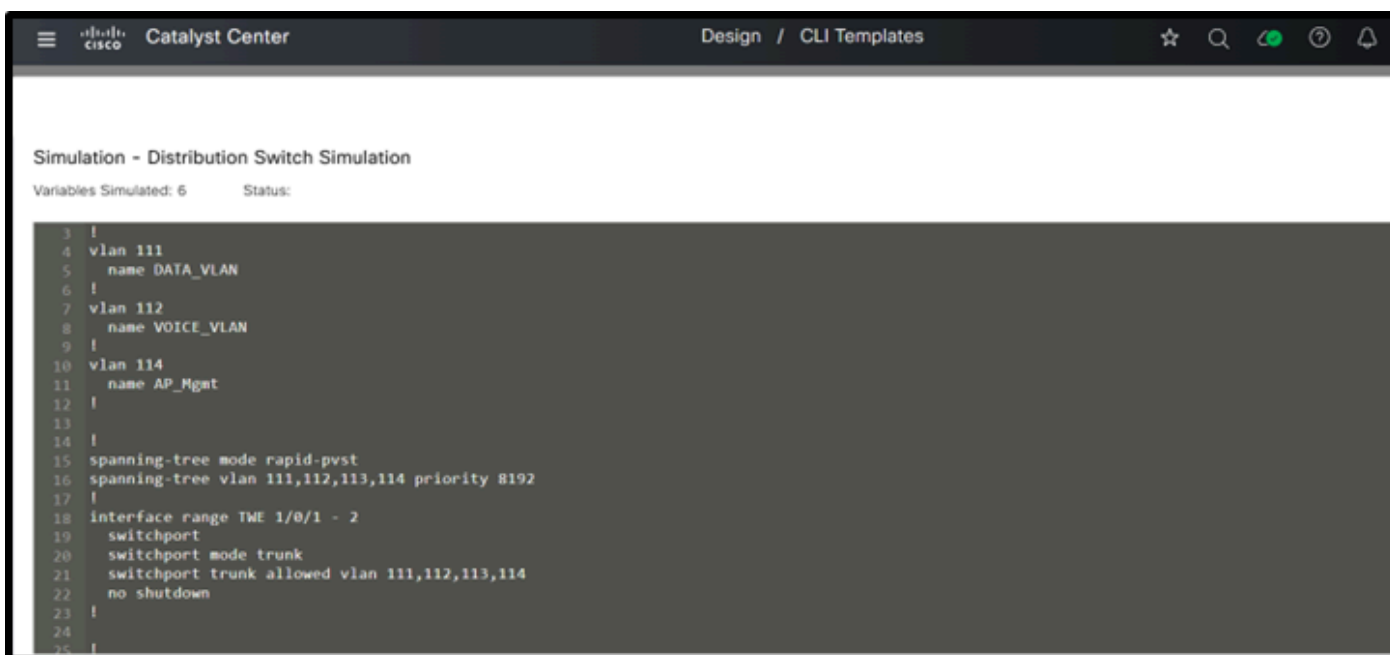
此模組演示使用Jinja2模板在分佈層進行分段。它利用branch\_number變數動態計算元網地址，從而實現自動化和可擴展的ACL配置。對於每個分支，ACL通過拒絕這些範圍之間的IP流量來阻止子網1(172.17.X.0)和子網2(172.16.X.0)之間的通訊。它還拒絕發往組播地址239.255.255.250的流量，但允許所有其他流量。VLAN介面根據分支編號動態分配，且ACL應用於該接口的入站。這種自動化方法可確保每個分支機構的有效分段，減少手動配置錯誤，並簡化網路策略實施。

最後，最後一個模組「分佈標準配置」與[訪問標準配置](#)模組中描述的模組（有關詳細資訊，請參閱該部分）幾乎完全相同。它包括最佳實踐、安全加固和安全裝置管理的主要功能。唯一的區別在於來源介面:在接入交換機模板中，它被定義為VLAN {{ branch\_number \* 100 + 13 }}，而在分佈交換機配置中，它可以硬編碼為Loopback0。

步驟 3:在部署配置之前執行模擬。



(1)配電開關模版模擬輸入輸出



(2)配電開關模版模擬輸入輸出

The screenshot shows the Catalyst Center interface with the title 'Design / CLI Templates'. Below the title, it says 'Simulation - Distribution Switch Simulation' and 'Variables Simulated: 6 Status:'. The main content is a code block with the following configuration:

```
26 interface loopback0
27 ip address 172.31.1.32 255.255.255.255
28 !
29 router ospf 1
30 router-id 172.31.1.32
31 !
32 key chain HSRP_KEY
33 key 0
34 key-string cisco@7875
35 !
36 interface vlan 111
37 description Data_Endpoints
38 ip address 172.17.0.21 255.255.240.0
39 standby bfd
40 standby version 2
41 standby 111 ip 172.17.0.1
42 standby 111 priority 250
43 standby 111 authentication md5 key-chain HSRP_KEY
44 standby 111 preempt delay minimum 120
45 no ip redirects
46 no ip unreachable
47 no ip proxy-arp
```

(3)配電開關模版模擬輸入輸出

The screenshot shows the Catalyst Center interface with the title 'Design / CLI Templates'. Below the title, it says 'Simulation - Distribution Switch Simulation' and 'Variables Simulated: 6 Status:'. The main content is a code block with the following configuration:

```
50 !
51 ! uplink interfaces
52 interface TWE1/1/1
53 no switchport
54 ip address 172.17.1.1 255.255.255.0
55 ip ospf 1 area 0
56 no shutdown
57 !
58 interface TWE1/1/2
59 no switchport
60 ip address 172.17.1.2 255.255.255.0
61 ip ospf 1 area 0
62 no shutdown
63 !
64 !
65 !
66 ip access-list extended BLOCK_BRANCH
67 deny ip 172.17.0.0 0.0.15.255 172.16.0.0 0.0.15.255
68 deny ip any host 239.255.255.250
69 permit ip any any
70 !
71 interface vlan 111
72 ip access-group BLOCK_BRANCH in
```

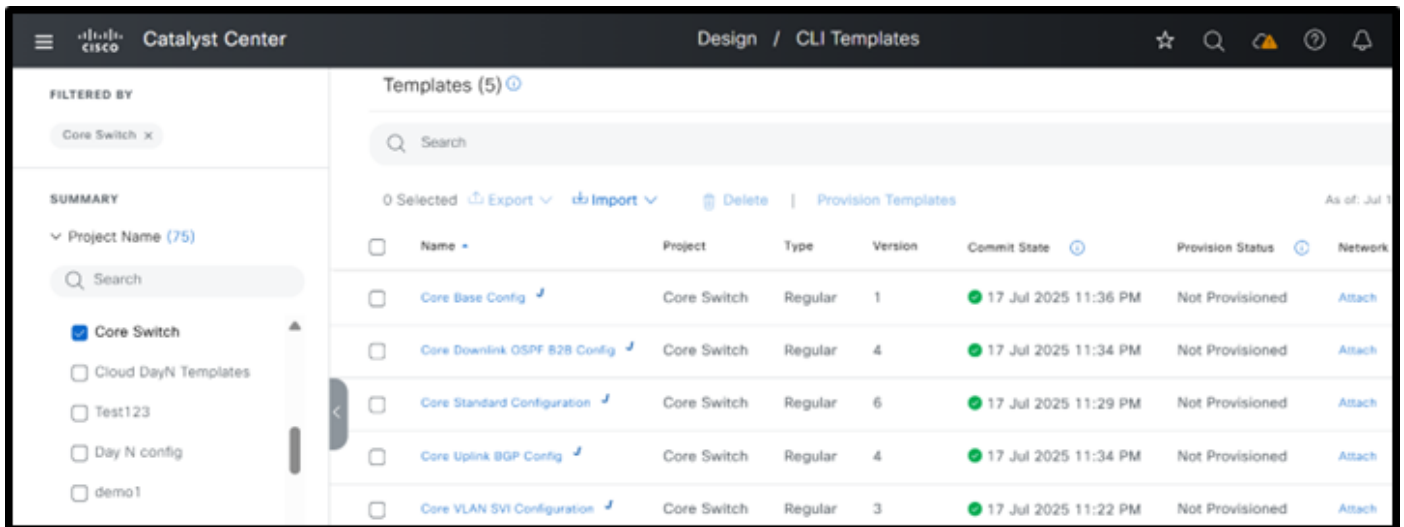
(4)配電開關模版模擬輸入輸出

這就是在分佈層使用模板生成配置的方法。現在，讓我們看一下核心層裝置，看看模板如何應用於其中。

### 核心層交換器

現在，為核心交換機設計一個模組化模板。基本模板及其模組是Cisco Catalyst Center「核心交換機」專案的一部分。請參閱步驟1中的基本模板。

## 步驟 1: 定義各種核心交換機結構



核心交換器範本結構

## 步驟 2: 定義各種模組

```
{% include "Core Switch/Core VLAN SVI Configuration" %}
{% include "Core Switch/Core Downlink OSPF B2B Config" %}
{% include "Core Switch/Core Uplink BGP Config" %}
{% include "Core Switch/Core Standard Configuration" %}

{{ Core_VLAN_SVI_Configuration () }}
{{ Core_Downlink_OSPF_B2B_Config () }}
{{ Core_Uplink_BGP_Config () }}
{{ Core_Standard_Config () }}
```

核心基本配置

大多數核心交換機配置在所有分支中都是類似的，因此可以對常用值進行硬編碼。通常，只有IP地址會更改，可以使用變數設定這些地址。由於每個分支機構通常只有兩台核心交換機，因此管理這些變數非常簡單。即使某些分支機構擁有更多的核心交換機，其數量仍少於接入或分佈交換機數量。這就是為什麼，作為最佳實踐，將接入和分佈交換機的變數降至最低更為重要，因為它們數量較多，而且變數過多會增加配置時間。

現在從第一個模組開始："核心VLAN SVI配置。" 在本範例中，核心交換器位於防火牆之後，且必須建立與防火牆對等的eBGP。此模組負責生成eBGP對等和OSPF鄰居關係所需的VLAN和相應的SVI。假設防火牆以主用/備用配置運行。

```

{% macro Core_VLAN_SVI_Configuration () %}
!
vlan 2001
  name eBGP_peering_to_FW
!
vlan 2002
  name OSPF_neighborship
!
interface vlan 2001
  description eBGP Peering to Firewall
  ip address {{ VLAN2001_IP }} 255.255.255.248
  bfd interval 100 min_rx 100 multiplier 3
  no ip redirects
  no ip unreachable
  no ip proxy-arp
!
interface vlan 2002
  description OSPF neighborship to Core SW 2
  ip address {{ VLAN2002_IP }} 255.255.255.248
  bfd interval 100 min_rx 100 multiplier 3
  ip ospf 1 a 0
  no ip redirects
  no ip unreachable
  no ip proxy-arp
!
{% endmacro %}

```

#### 核心VLAN SVI配置

如前所述，此模組建立建立OSPF和BGP鄰居關係所需的VLAN和關聯的SVI。除SVI IP地址以外的所有引數均採用硬編碼，包括與IP編址計劃一致的子網掩碼。此方法有助於限制變數並減少出現配置錯誤的可能性。

現在，我們來看看「核心下行鏈路OSPF B2B配置」模塊，該模組為核心交換機1和核心交換機2之間的下行鏈路介面、OSPF和背對背鏈路生成配置。

```

{% macro Core_Downlink_OSPF_B2B_Config () %}
!
interface loopback0
ip address {{ loopback_ip }} 255.255.255.255
!
router ospf 1
router-id {{ loopback_ip }}
default-information originate
!
! downlink interfaces
interface TWE1/0/1
ip address {{ twe1_0_1_ip }} 255.255.255.0
ip ospf 1 area 0
no shutdown
!
interface TWE1/0/2
ip address {{ twe1_0_2_ip }} 255.255.255.0
ip ospf 1 area 0
no shutdown
!
interface TWE1/0/24
description Towards_Core_SW
switchport mode trunk
switchport trunk allowed vlan 2001,2002
channel-group 10 mode active
spanning-tree portfast trunk
no shutdown
!
interface TWE1/0/48
description Towards_Core_SW
switchport mode trunk
switchport trunk allowed vlan 2001,2002
channel-group 10 mode active
spanning-tree portfast trunk
no shutdown
!
interface Port-channel10
description Towards_Core_SW
switchport mode trunk
switchport trunk allowed vlan 2001,2002
spanning-tree portfast trunk
no shutdown
!
{% endmacro %}

```

核心下行鏈路OSPF B2B配置

與先前的模組類似，此模組中的大多數值都經過硬編碼，以最小化變數數。只有環回介面和下行鏈路介面的IP地址是可變的。此外，背對背埠通道和VLAN在不同分支機構中實現了標準化。

現在，我們來看一下「核心上行鏈路BGP配置」模組，它生成BGP配置並管理連線到防火牆的上行鏈路。

```

{% macro Core_Uplink_BGP_Config () %}
!
router bgp {{ AS_Number }}
  bgp log-neighbor-changes
  bgp router-id interface Loopback0
  bgp graceful-restart
!
! eBGP Peering with Firewall
neighbor {{ BGP_NEIGHBOR }} remote-as {{ REMOTE_AS }}
neighbor {{ BGP_NEIGHBOR }} description eBGP Peering with Firewall
neighbor {{ BGP_NEIGHBOR }} update-source vlan 2001
neighbor {{ BGP_NEIGHBOR }} fall-over bfd
aggregate-address {{ AGGREGATE_IP }} {{ AGGREGATE_MASK }} summary-only
!
address-family ipv4
  network {{ loopback_ip }} mask 255.255.255.255
  neighbor {{ BGP_NEIGHBOR }} activate
exit-address-family
!
! Redistribute OSPF into BGP
redistribute ospf
!
! Uplink interfaces
interface TWE1/0/23
  description Towards...Firewall
  switchport mode access
  switchport access vlan 2001
  channel-group 10 mode active
  spanning-tree portfast
  no shutdown
!
interface TWE1/0/47
  description Towards...Firewall
  switchport mode access
  switchport access vlan 2001
  channel-group 10 mode active
  spanning-tree portfast
  no shutdown
!
interface Port-channel10
  description Towards...Firewall
  switchport mode access
  switchport access vlan 2001
  spanning-tree portfast
  no shutdown
!
{% endmacro %}

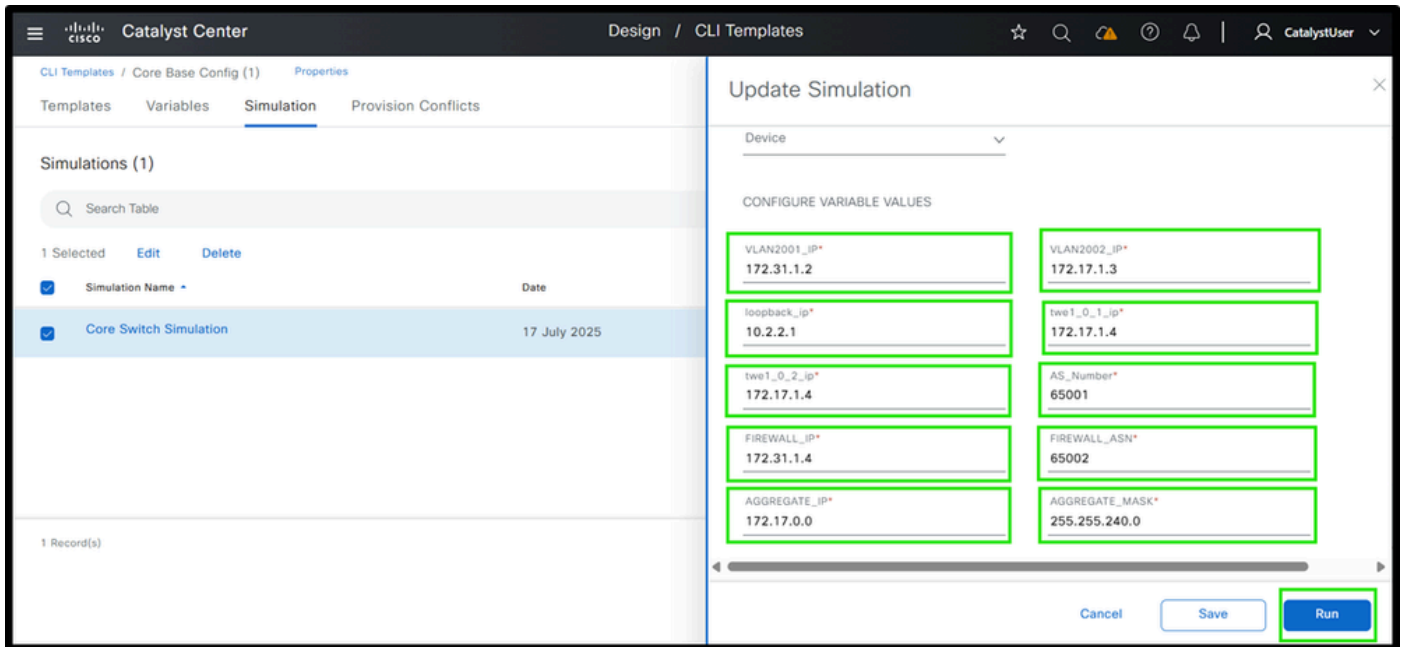
```

#### 核心上行鏈路BGP配置

此模組生成與防火牆建立eBGP鄰居關係所需的BGP配置。如上所示，大多數值是硬編碼的，因為它們在不同分支之間保持一致。只有IP地址和AS編號（因每個分支而異）會作為輸入變數並用於生成必要的配置。大多數其他設定已標準化，以儘量減少變數的數量。連線到防火牆的上行鏈路介面與用於eBGP鄰居的VLAN一起指定，後者由上一個模組生成。

最後，最後一個模組「核心標準配置」與訪問標準配置中介紹的模組幾乎完全相同（有關詳細資訊，請參閱該部分）。它包括最佳實踐、安全加固和安全裝置管理的主要功能。與分佈層交換機配置一致，源介面也可在此模組中設定為loopback0，並且此值可以採用硬編碼。

### 步驟 3:執行模擬



(1)核心交換機模板模擬輸入輸出



(2)核心交換機模板模擬輸入和輸出

The screenshot shows the Catalyst Center interface with the title "Design / CLI Templates". Below the header, it says "Simulation - Core Switch Simulation" and "Variables Simulated: 10 Status:". A terminal window displays the following configuration:

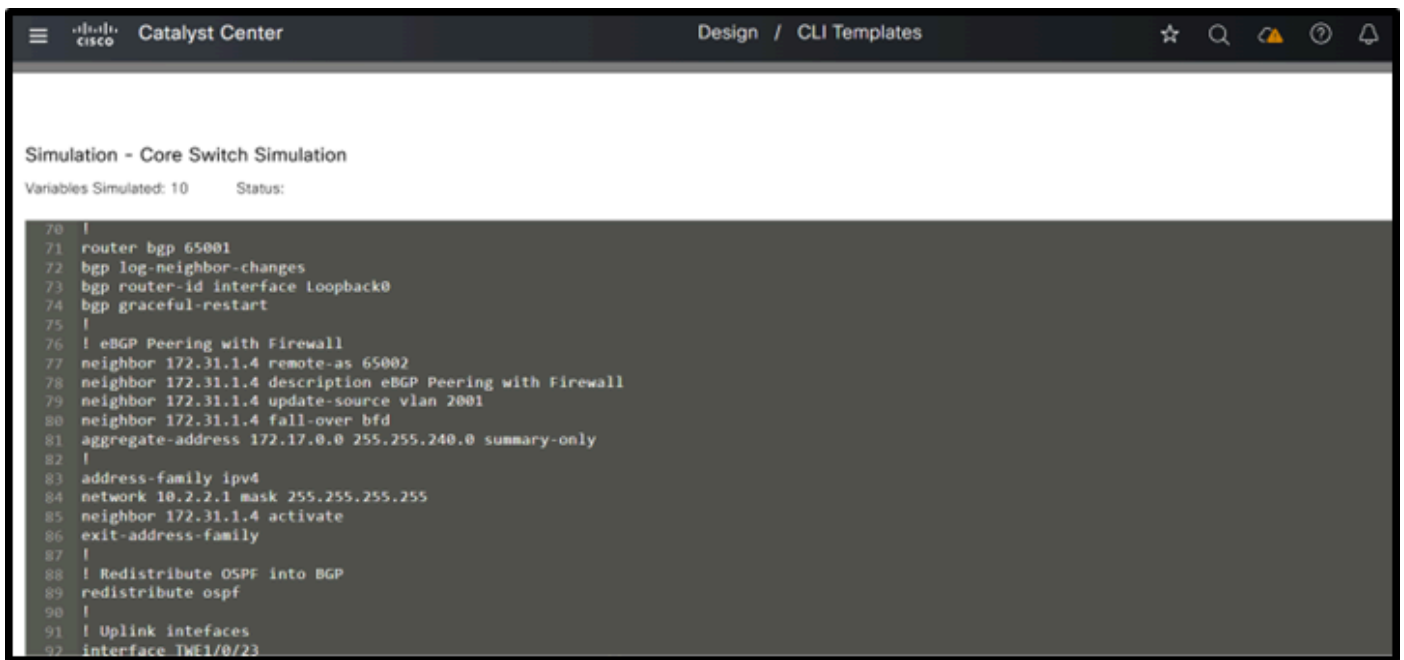
```
26
27 |
28 interface loopback0
29 ip address 10.2.2.1 255.255.255.255
30 |
31 router ospf 1
32 router-id 10.2.2.1
33 default-information originate
34 |
35 | downlink interfaces
36 interface TWE1/0/1
37 ip address 172.17.1.4 255.255.255.0
38 ip ospf 1 area 0
39 no shutdown
40 |
41 interface TWE1/0/2
42 ip address 172.17.1.4 255.255.255.0
43 ip ospf 1 area 0
44 no shutdown
45 |
46 interface TWE1/0/24
47 description Towards_Core_SW
48 switchport mode trunk
```

(3)核心交換機模板模擬輸入輸出

The screenshot shows the Catalyst Center interface with the title "Design / CLI Templates". Below the header, it says "Simulation - Core Switch Simulation" and "Variables Simulated: 10 Status:". A terminal window displays the following configuration:

```
39 no shutdown
40 |
41 interface TWE1/0/2
42 ip address 172.17.1.4 255.255.255.0
43 ip ospf 1 area 0
44 no shutdown
45 |
46 interface TWE1/0/24
47 description Towards_Core_SW
48 switchport mode trunk
49 switchport trunk allowed vlan 2001,2002
50 channel-group 10 mode active
51 spanning-tree portfast trunk
52 no shutdown
53 |
54 interface TWE1/0/48
55 description Towards_Core_SW
56 switchport mode trunk
57 switchport trunk allowed vlan 2001,2002
58 channel-group 10 mode active
59 spanning-tree portfast trunk
60 no shutdown
61 |
```

(4)核心交換機模板模擬輸入輸出



The screenshot shows the Catalyst Center interface with a CLI template for a Core Switch Simulation. The configuration includes BGP settings, eBGP peering with a firewall, address-family configuration, and OSPF redistribution.

```
70 |
71 | router bgp 65001
72 |   bgp log-neighbor-changes
73 |   bgp router-id interface Loopback0
74 |   bgp graceful-restart
75 |
76 | ! eBGP Peering with Firewall
77 | neighbor 172.31.1.4 remote-as 65002
78 | neighbor 172.31.1.4 description eBGP Peering with Firewall
79 | neighbor 172.31.1.4 update-source vlan 2001
80 | neighbor 172.31.1.4 fall-over bfd
81 | aggregate-address 172.17.0.0 255.255.240.0 summary-only
82 |
83 | address-family ipv4
84 |   network 10.2.2.1 mask 255.255.255.255
85 |   neighbor 172.31.1.4 activate
86 | exit-address-family
87 |
88 | ! Redistribute OSPF into BGP
89 | redistribute ospf
90 |
91 | ! Uplink interfaces
92 | interface TWE1/0/23
```

(5)核心交換機模板模擬輸入輸出

這樣就完成了有關設計三層體系結構的模板的詳細說明，概述了每個模組的結構和配置。

所有這些模組都採用了前面介紹的最佳做法。



附註：在設計摺疊核心體系結構的模板時，請參閱為三層體系結構提供的說明。模板結構保持不變；但是，以前在核心層和分佈層單獨實現的特性現在在摺疊的核心層合併。這裡也可以使用相同的模組化模板方法，方法是建立一個基礎模板並引用其中的相關模組。

## 摘要

傳統的三層園區架構通常依賴於跨核心、分佈層和接入層的大量手動配置。這種方法不僅耗時，而且容易發生人為錯誤。缺乏自動化和集中化管理顯著增加了運營開銷，使得難以有效擴展和管理現代的動態園區網路。通過Catalyst Center CLI模板，可以為傳統LAN網路自動配置功能。在調配裝置時，必須使用模組化方法。模組可以基於不同層使用的各種功能。最後將這些模組繫結到基礎模組。

## 行動號召

我們邀請企業採用本白皮書提供的模組化模板方法，將其作為在三層核心架構和摺疊核心架構中標準化交換機配置和最佳化網路操作的最佳實踐。

- 通過實施模組化模板，網路團隊可以：
- 通過一致、可重複的配置做法提高操作效率。
- 最大程度減少人為錯誤並減少故障排除時間。
- 實現更高的可擴充性，以支援增長和不斷發展的業務需求。
- 確保不同環境的配置一致性。

此方法不僅可簡化日常管理，還可實現更快的部署、簡化更新週期以及增強與安全性和合規性要求的協調。採用模組化模板可使您的網路在不斷變化的IT環境中保持靈活性、可復原性和長期成功。

如需實用演示，請瞭解有關模板的更多資訊，請參閱YouTube系列

1如何在Catalyst Center中建立和管理模板

<https://youtu.be/SyUqEEcwy0>

2如何在Catalyst Center的CLI模板中使用系統繫結變數

<https://youtu.be/gV1QBuHYJdo>

## 作者

思科客戶體驗客戶交付架構師Naveen Kumar

Risabh Mishra，思科客戶體驗諮詢工程師

## 關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。