

Configuring and Troubleshooting Catalyst Center Templates

cisco Live !

Alan Gardner

CEO, Current Technologies CLC, @alangardnerCT

3xCCIE #22758

<https://www.linkedin.com/in/alangard/>

Cisco Webex App

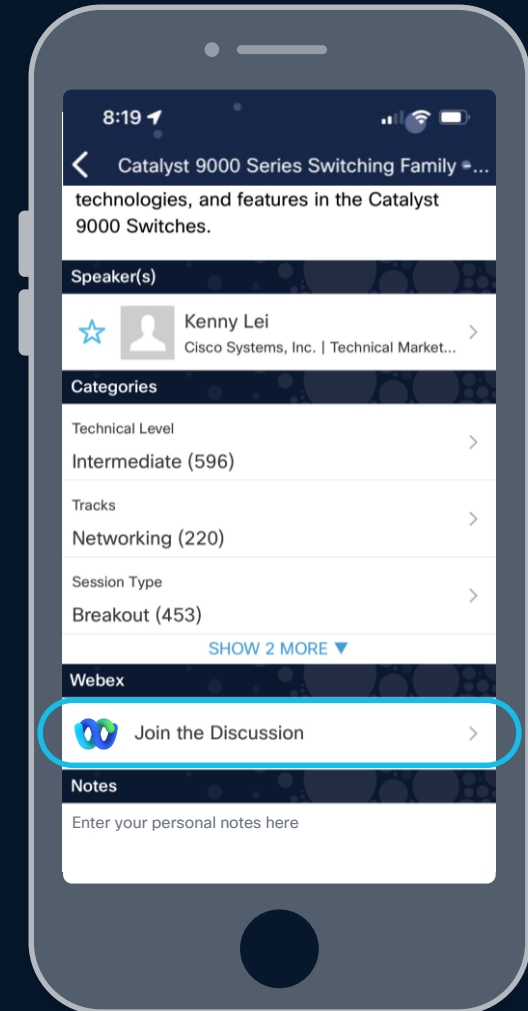
Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 13, 2025.

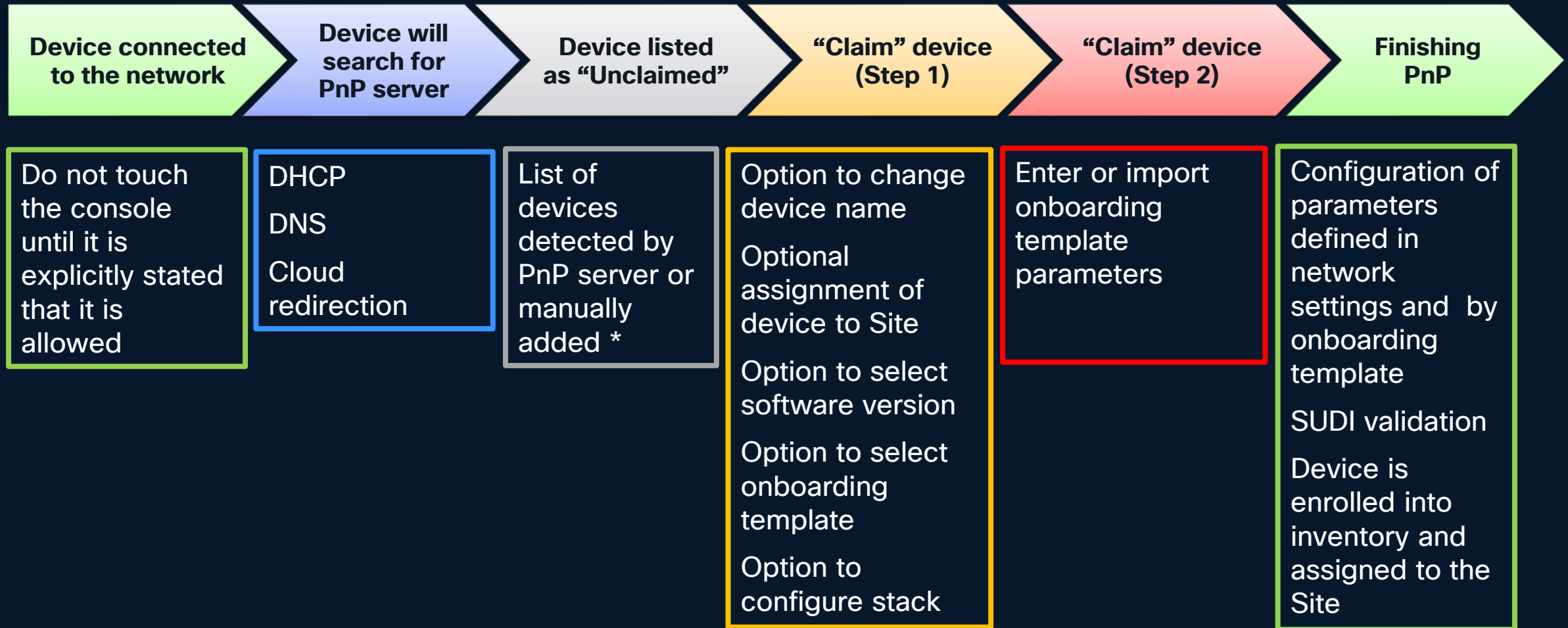


<https://ciscolive.ciscoevents.com/ciscolivebot/#CISCOU-3004>

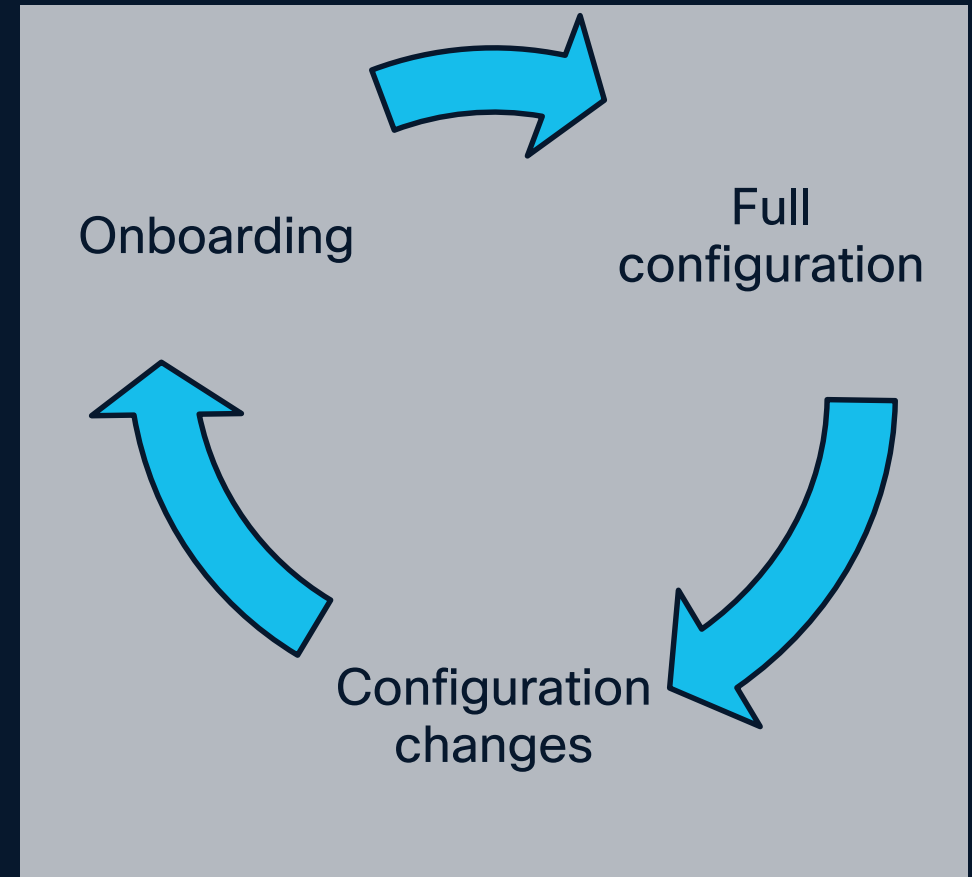
Agenda

- 01 Device Onboarding
- 02 Catalyst Center Templates
- 03 Projects and Templates
- 04 Day-0 and Day-N Templates
- 05 CLI Template Languages
- 06 Network Profiles
- 07 Troubleshooting Best Practices
- 08 Troubleshooting Template Deployments
- 09 Template Compliance Checks

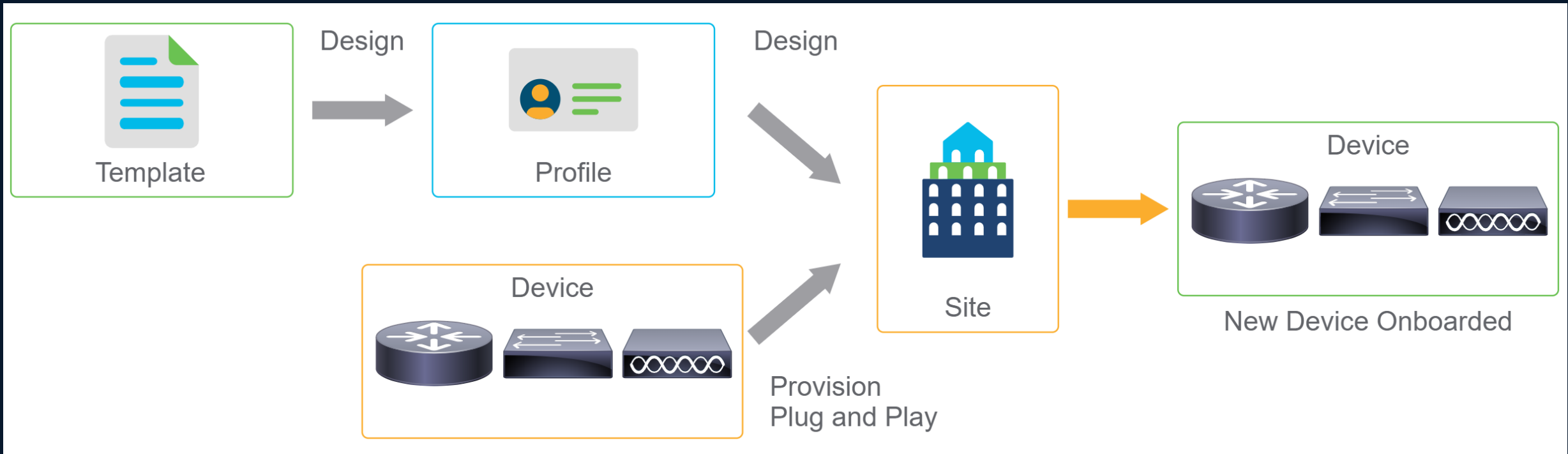
Device Onboarding - Provision Workflow



Use of Catalyst Center **CLI templates** can significantly simplify device configuration through its entire life cycle and helps to maintain configuration consistency.



Catalyst Center Templates



Configuration Templates

TemplatesVariablesSimulationProvision Conflicts

(x) System Variables AssistantTemplate HistoryAttach to Network ProfileShow Design Conflicts

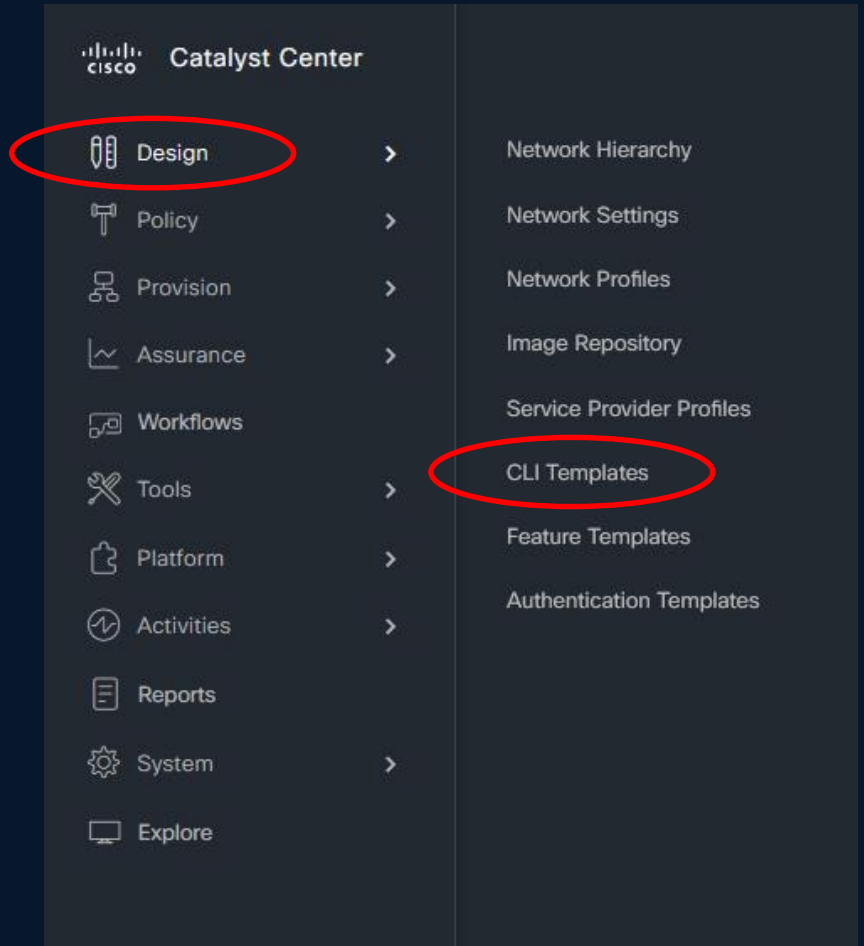
Auto saved: 12:43 PM (every 5 mins)SearchCloneExportDeleteCLI Template Best Practices

```
1 ###
2 ### This template works for DNAC 2.2.3.4, workaround for CSCvz52730 works for DNAC 2.2.3.x. For older DNAC versions it might be necessary to change variable definitions
3 ###
4 ### Interfaces with Closed Authentication
5 ###
6 ### !!! Interfaces with Closed Authentication
7 ###
8 !
9 #foreach($interface in ${Closed_Authentication_Interfaces})
10 ### Original if statement - CSCvz52730 must be fixed
11 #if ($interface.portType == "Ethernet Port" &&
12     !$interface.description.contains("Uplink"))
13 )
14 ### Workaround for CSCvz52730
15 ###if($interface.portType == "Ethernet Port"
16 ###)
17 ###!!!
18 ###!!! $interface.portType , $interface.portName , $interface.description
19 ###!!!
20 ! @start-ignore-compliance
21     default interface $interface.portName
22 ! @end-ignore-compliance
23     interface $interface.portName
24         description Access -lan- Closed authentication
25         switchport mode access
26         device-tracking attach-policy IPDT_POLICY
27     ### ip flow monitor dnacmonitor input
28     ### ip flow monitor dnacmonitor output
29     ### ip flow monitor dnacmonitor_dns input
30     ### ip flow monitor dnacmonitor_dns output
31     dot1x timeout tx-period 7
32     dot1x max-reauth-req 3
33     source template DefaultWiredDot1xClosedAuth
34     spanning-tree portfast
35     spanning-tree bpduguard enable
36     service-policy input DNA-MARKING_IN
37     service-policy output DNA-dscp#APIC_QOS_Q_OUT
38     ip nbar protocol-discovery
39 !
40 #end
41 #end
42 ###
43 ### Interfaces with Open Authentication
44 ###
45 ###!!! Interfaces with Open Authentication
```

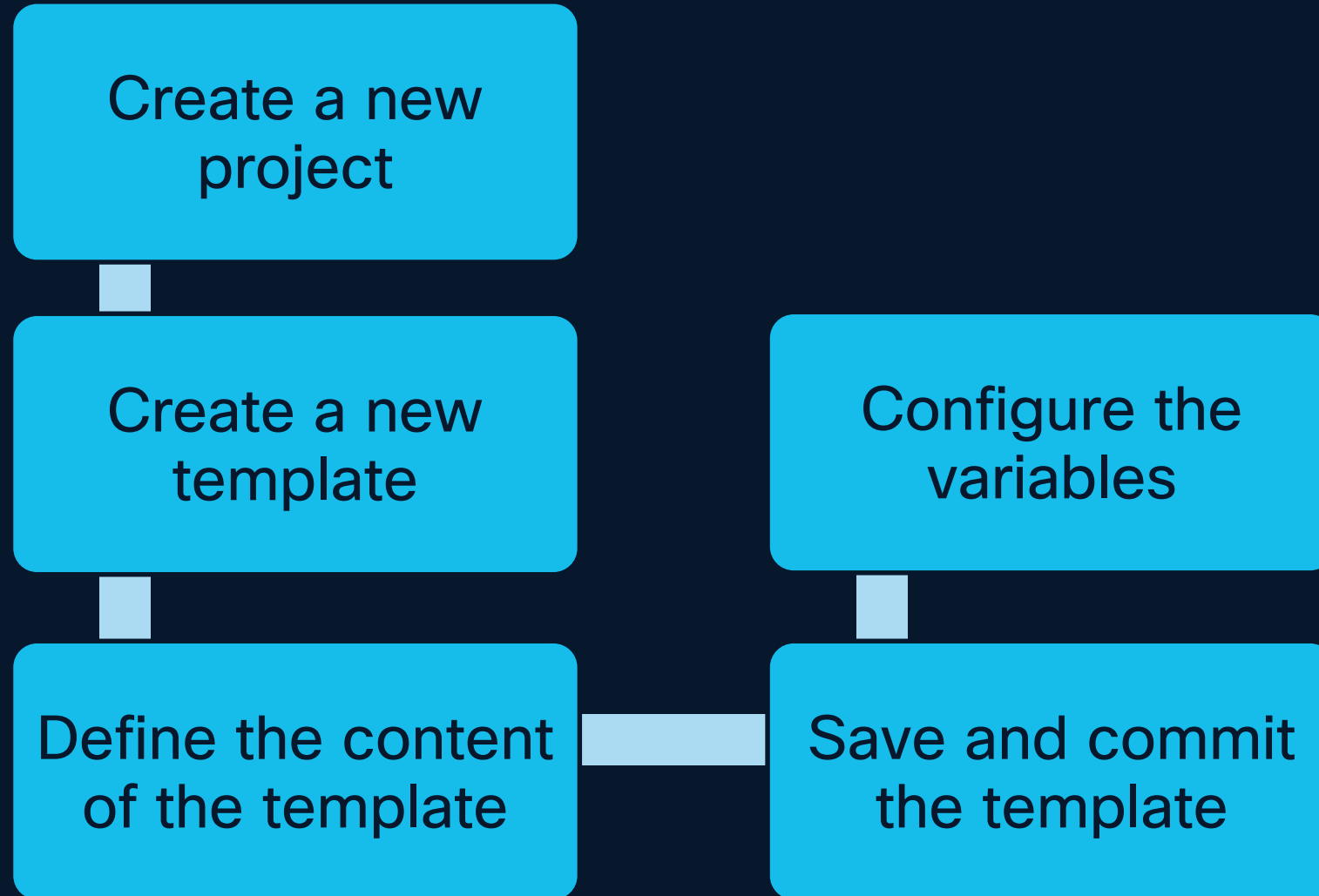
- Segments of code
- Support variables
- Provides basic programming constructs
- Flexible

Template Basics

- Currently under the Design menu
 - **Design -> CLI templates**
- In previous versions
 - **Tools -> Template Hub**
 - **Tools -> Template Editor**
- Cisco Catalyst Center provides an interactive template hub to author CLI templates.
- Easily design with predefined configurations by using parameterized elements or variables.
- Use the template to deploy your devices in one or more sites that are configured anywhere in the network.



Template Creation Workflow



Projects and Templates

Catalyst Center Design / CLI Templates

Templates (204)

0 Selected Export Import Delete Provision Templates

As of: Dec 9, 2024 6:15 PM

Name	Project	Type	Version	Commit State	Provision Status	Network Profiles	Actions
Base Config	Sample Jinja Templates	Regular	Not Committed	Not Committed	Not Provisioned	NA	...
BTS05-INFRA-AP_Pool-138_32	SDA DHCP Server	Regular	2	19 Nov 2021 04:00 PM	Not Provisioned	Attach	...
BTS05-INFRA-IoT_Pool-138_160	SDA DHCP Server	Regular	1	19 Nov 2021 03:32 PM	Not Provisioned	Attach	...
BTS05-VN-Campus-End-Users-1...	SDA DHCP Server	Regular	1	19 Nov 2021 03:35 PM	Not Provisioned	Attach	...

Catalyst Center

SUMMARY

Project Name (31)

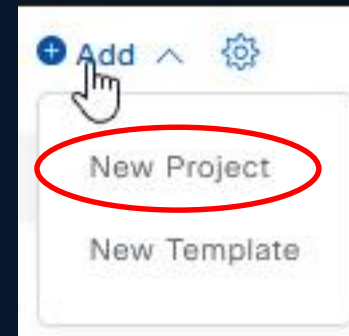
Search

- Silent Trustsec Branch
- C3650-PEN
- Wake_on_LAN
- Wireless-tests
- L2 NAT
- Smart Licensing
- C9200_C9200L_CTS
- test-proj
- CCIE
- Utility

Show less

Projects:

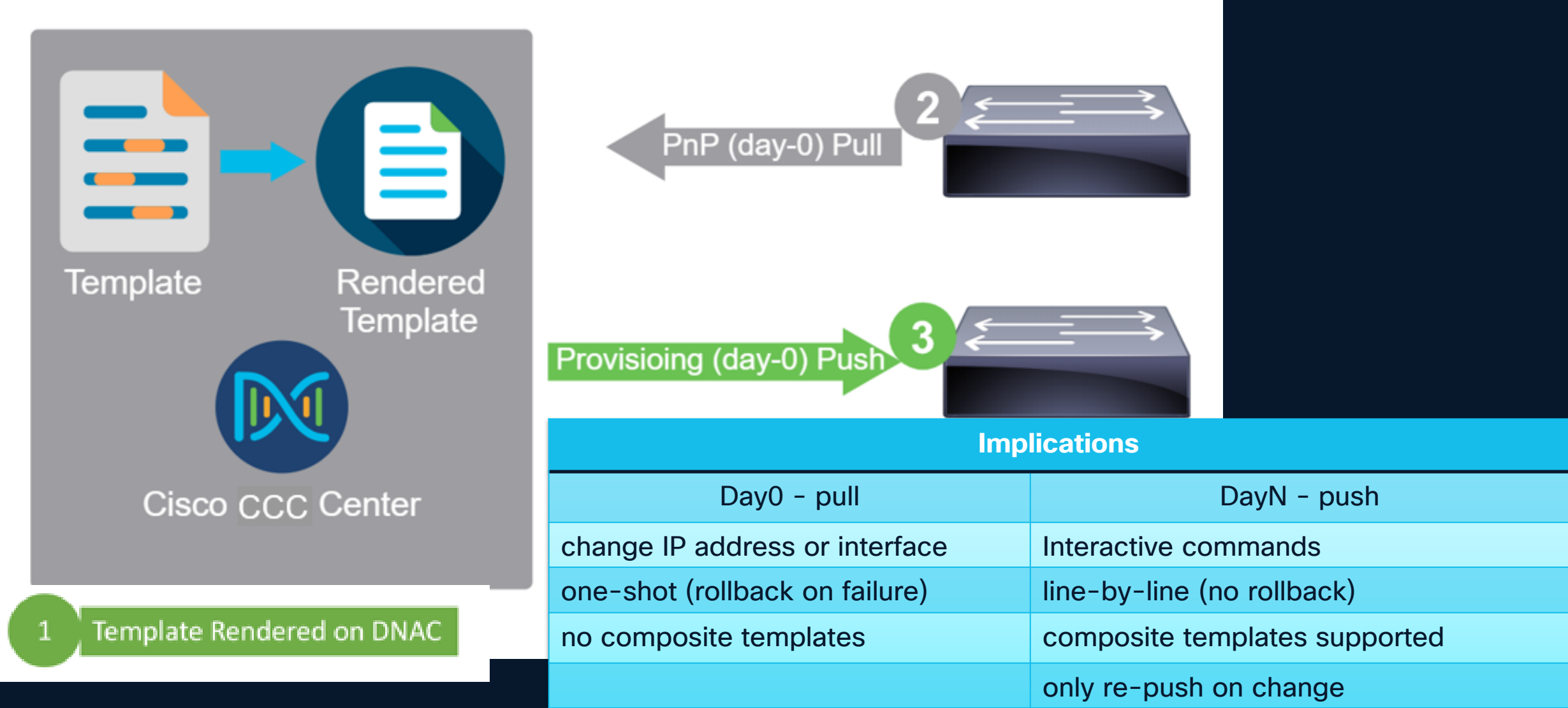
- “Folders” where templates are organized in structure
- Can be exported and imported



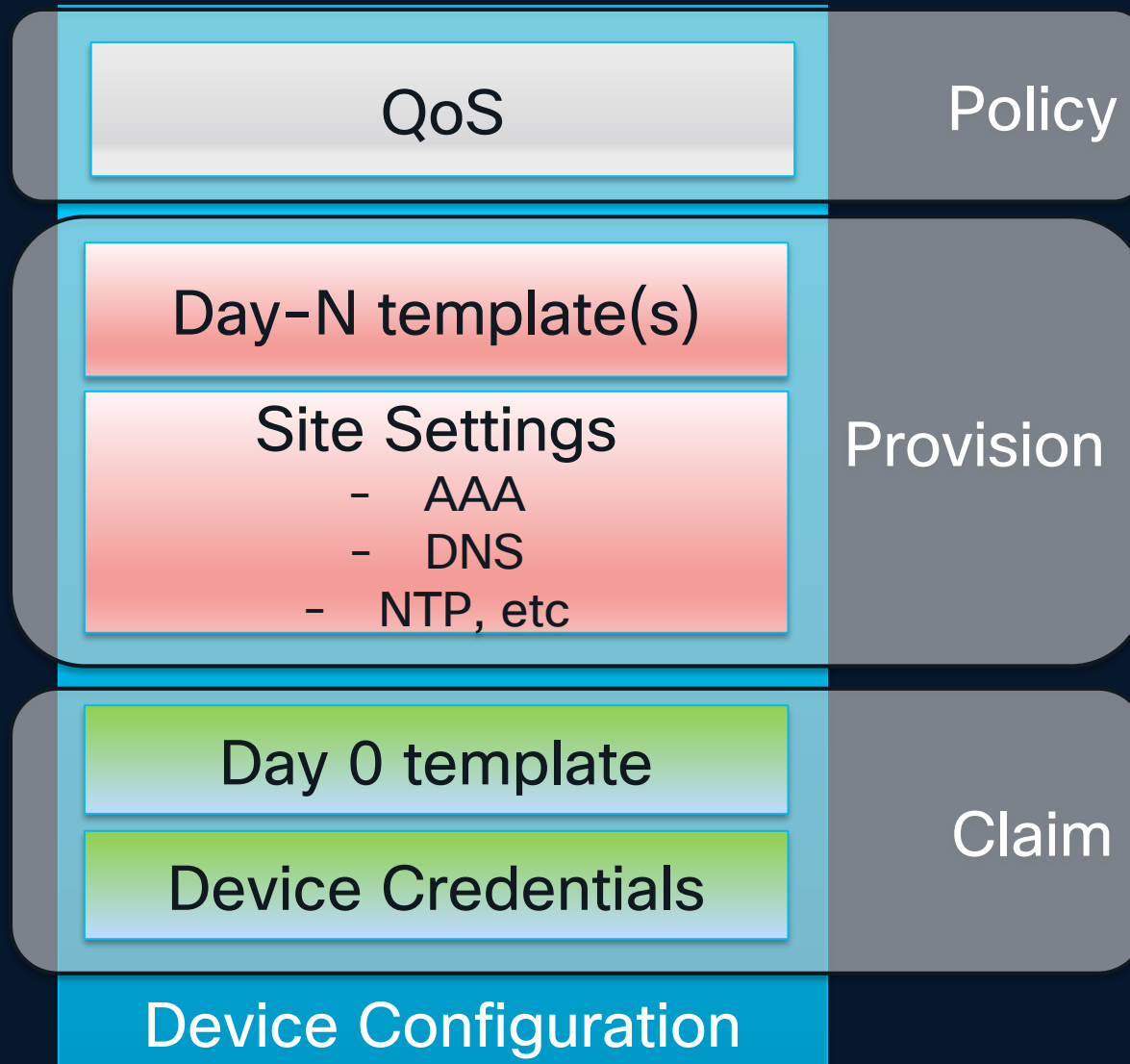
Adding Projects:

- Name
- Description (optional)

Day-0 and Day-N Templates



Day-0 and Day-N Templates



- In “**Day-0 Configuration**”, Cisco Catalyst Center generates a CLI configuration, including device credentials and enabling SSH for users.
- For “Site Settings” and Day-N Templates, need “Provision”

CLI Template Languages

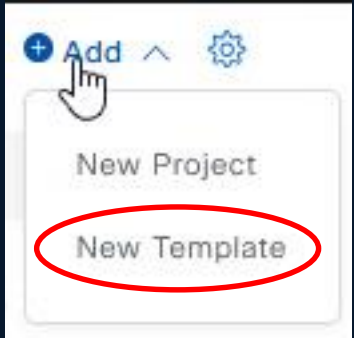
Apache Velocity Language

- Java-based template engine
- Allows CatC to expose variables
- Allows scripting logic to be included in templates
- Provides:
 - Source binding
 - Manipulation of variables
 - Conditional branches
 - Loops
 - Methods

Jinja

- Template engine written in Python
- Inherits many Python's capabilities
- Allows CatC to expose variables
- Allows scripting logic to be included in templates
- Provides:
 - Source binding
 - Manipulation of variables
 - Conditional branches
 - Loops
 - Methods and filters

Adding CLI Template



Add New Template

Template Details

Define the properties for the template.

Template Name*
Test-template

Project Name*
test-proj

Template Type
☒ Regular Template ☐ Composite Sequence

Template Language
☐ JINJA ☒ VELOCITY

Software Type*
IOS-XE

Device Type Details

Add the types of devices you want to associate with the template

DEVICE DETAILS*

[Edit Device Details](#)

Device Family
Switches and Hubs

Devices
Cisco Catalyst IE-3400-8P2S Rugged Switch
Cisco Catalyst IE-3400-8T2S Rugged Switch

Additional Details

Device Tags
Device Tags

Software Version

Template Description

Cancel Continue

Adding template (parameters):

- Template Name
- Project Name (selected from available)
- Template Type (Regular or Composite)
- Template Language (Jinja or Velocity)
- Software Type (IOS, IOS-XE, ...)
- Device Family (selected from available)
- Devices (optional, selected from available families or models)
- Device Tags (optional, selected from available, used to provide better granularity of device selection)
- Software Version (optional)
- Template Description (optional)

← Mandatory

← Optional

CLI Template Types – Regular vs. Composite

Regular

- Generates real configuration segments
- Can use variables
- Can use:
 - Source binding
 - Manipulation of variables
 - Conditional branches
 - Loops
 - Methods

Composite

- Sequence of regular templates
- Sequence can be reordered
- Sequence can be aborted if deployment fails

CLI Template Types – Regular vs. Composite

Regular

CLI Templates / Two VLAN L2 NAT and Profinet (10) Properties

Templates Variables Simulation Provision Conflicts

(x) System Variables Assistant Template History Attach to N

```
1 ###
2 ###
3 ### Create Machine and External VLANs
4 !
5 vlan $vlanid_l2nat
6 name External_VLAN
7 vlan $vlanid_profinet
8 name Machine_VLAN
9 ###
10 ### Create SVI Interfaces
11 !
12 interface Vlan$vlanid_profinet
13 ip address $int_gw_addr $int_gw_mask
14 interface Vlan$vlanid_l2nat
15 ip address $out_gw_addr $out_gw_mask
16 ###
17 ### Create L2NAT instance
18 ! @start-ignore-compliance
19 !
20 l2nat instance $l2nat_instance_name
21 !
22 exit
23 !
24 default l2nat instance $l2nat_instance_name
25 !
26 l2nat instance $l2nat_instance_name
27 !
28 ! @end-ignore-compliance
29 ### Following command will be deprecated
30 ### instance-id 1
```

Composite

CLI Templates / IE3400-2VLAN (3) Properties

Templates Simulation Provision Conflicts

Add applicable templates to create your composite template's sequence.

☐ Abort sequence if deployment fails

Search Table

Reorder More Actions

Template Name

Miscellaneous configuration_IE3400

Authentication templates_IE3400

VLAN interfaces - IE3400

Access interfaces and trunks

Two VLAN L2 NAT and Profinet

CCV sensor prerequisites

CLI Template Editor

The screenshot displays the Cisco Catalyst Center CLI Template Editor. The interface includes a top navigation bar with the Cisco logo and 'Catalyst Center' text. Below this, a breadcrumb trail shows 'Design / CLI Templates'. A sidebar on the left contains tabs for 'Templates', 'Variables', 'Simulation', and 'Provision Conflicts'. The main area is a code editor showing a CLI template for interfaces with closed authentication. A text box on the right lists three items: 'Template simulation', 'Variable definition', and 'Template code creation', with red arrows pointing to the 'Simulation' tab, the 'Variables' tab, and the code editor respectively. At the bottom right, three buttons are circled in red: 'Discard Changes', 'Save', and 'Commit'. Above these buttons, two tabs are visible: 'Editor View' and 'Version Compare View', with red arrows pointing to them.

Editor View Version Compare View

Templates Variables Simulation Provision Conflicts

(x) System Variables Assistant Template History Attach to Network Profile Show Design Conflicts

Auto saved: 3:50 PM (every 5 mins) Search Clone Export Delete CLI Template Best Practices

```
1  ###
2  ### This template works for DNAC 2.2.3.4, workaround for CSCvz52730 works for DNAC 2.2.3.x. For older DNAC versions it might be necessary to change variable definitions
3  ###
4  ### Interfaces with Closed Authentication
5  ###
6  ### !!! Interfaces with Closed Authentication
7  ###
8  !
9  #foreach($interface in ${Closed_Authentication_Interfaces})
10 ### Original if statement - CSCvz52730 must be fixed
11 -#if ($interface.portType == "Ethernet Port") &&
12     $interface.description.contains("Uplink")
13 )
14 ### Workaround for CSCvz52730
15 ###if($interface.portType == "Ethernet Port"
16 ###)
17 ###!!!
18 ###!!! $interface.portType , $interface.portName , $interface.description
19 !!! $interface
20 ###!!!
21 ! @start-ignore-compliance
22 default interface $interface.portName
23 ! @end-ignore-compliance
24 interface $interface.portName
25 description Access -lan- Closed authentication
26 switchport mode access
27 device-tracking attach-policy IPDT_POLICY
28 ### ip flow monitor dnacmonitor input
29 ### ip flow monitor dnacmonitor output
30 ### ip flow monitor dnacmonitor_dns input
31 -### ip flow monitor dnacmonitor_dns output
32 dot1x timeout tx-period 7
33 dot1x max-reauth-req 3
34 source template DefaultWiredDot1xClosedAuth
35 spanning-tree portfast
36 spanning-tree bpduguard enable
37 service-policy input DNA-MARKING_IN
38 service-policy output DNA-dscp#APIC_QOS_Q_OUT
39 ip nbar protocol-discovery
40 !
41 #end
42 #end
43 ###
44 ### Interfaces with Open Authentication
45 ###
```

Template simulation
Variable definition
Template code creation

Discard Changes Save Commit

CLI Template Editor - History

CLI Templates / Configure access interfaces and trunks (9) Properties

Templates Variables Simulation Provision Conflicts

(x) System Variables Assistant **Template History** Attach to Network Profile Show Design Conflicts

Version 8 - 19 November 2024, 10:38 AM Search **Save as latest** Latest

```
1 ###
2 ### This template works for DNAC 2.2.3.4, workaround for CSCvz52730 works for DNAC 2.2.3.x. For older DNAC versions
   it might be necessary to change variable definitions
3 ###
4 ### Interfaces with Closed Authentication
5 ###
6 ### !!! Interfaces with Closed Authentication
7 ###
8 !
9 #foreach($interface in ${Closed_Authentication_Interfaces})
10 ### Original if statement - CSCvz52730 must be fixed
11 #if ($interface.portType == "Ethernet Port" &&
12     !$interface.description.contains("Uplink"))
13 )
14 ### Workaround for CSCvz52730
15 ###if($interface.portType == "Ethernet Port"
16 ###)
17 #####
18 ##### $interface.portType , $interface.portName , $interface.description
19 - #####
20 default interface $interface.portName
21 interface $interface.portName
22 description Access -lan- Closed authentication
23 switchport mode access
24 device tracking attach-policy IPDT_POLICY
25 ### ip flow monitor dnacmonitor input
26 ### ip flow monitor dnacmonitor output
27 dot1x timeout tx period 7
28 dot1x max-reauth req 3
29 source template DefaultWiredDot1xClosedAuth
30 spanning tree portfast
31 spanning tree bpduguard enable
32 ### service-policy input DNA-MARKING_IN
33 ### service-policy output DNA-dscp#APIC_QOS_Q_OUT
34 ### ip nbar protocol-discovery
35 !
36 #end
37 #end
38 ###
39 ### Interfaces with Open Authentication
40 ###
41 ##### Interfaces with Open Authentication
42 #foreach($interface in ${Open_Authentication_Interfaces})
43 ### Original if statement - CSCvz52730 must be fixed
44 #if ($interface.portType == "Ethernet Port" &&
```

```
1 ###
2 ### This template works for DNAC 2.2.3.4, workaround for CSCvz52730 works for DNAC 2.2.3.x. For older DNAC versions
   it might be necessary to change variable definitions
3 ###
4 ### Interfaces with Closed Authentication
5 ###
6 ### !!! Interfaces with Closed Authentication
7 ###
8 !
9 #foreach($interface in ${Closed_Authentication_Interfaces})
10 ### Original if statement - CSCvz52730 must be fixed
11 #if ($interface.portType == "Ethernet Port" &&
12     !$interface.description.contains("Uplink"))
13 )
14 ### Workaround for CSCvz52730
15 ###if($interface.portType == "Ethernet Port"
16 ###)
17 #####
18 ##### $interface.portType , $interface.portName , $interface.description
19 - #####
20- @start ignore-compliance
21 default interface $interface.portName
22- @end ignore-compliance
23 interface $interface.portName
24 description Access -lan- Closed authentication
25 switchport mode access
26 device tracking attach-policy IPDT_POLICY
27 ### ip flow monitor dnacmonitor input
28 ### ip flow monitor dnacmonitor output
29 dot1x timeout tx period 7
30 dot1x max-reauth req 3
31 source template DefaultWiredDot1xClosedAuth
32 spanning tree portfast
33 spanning tree bpduguard enable
34 ### service-policy input DNA-MARKING_IN
35 ### service-policy output DNA-dscp#APIC_QOS_Q_OUT
36 ### ip nbar protocol-discovery
37 !
38 #end
39 #end
40 ###
41 ### Interfaces with Open Authentication
42 ###
43 ##### Interfaces with Open Authentication
44 #foreach($interface in ${Open_Authentication_Interfaces})
```

Save as latest

Highlighted differences

Template History

Search

Latest Content
Author: jpilar

Version 9
Author: jpilar | 11 December 2024, 3:35 PM
Ignore compliance check for default interface c
ommands
Compare View

Version 8
Author: jpilar | 19 November 2024, 10:38 AM
Compare View

Version 7
Author: jpilar | 22 June 2024, 1:22 PM

Version 6
Author: jpilar | 21 June 2024, 4:56 PM

Version 5
Author: jpilar | 11 June 2024, 1:09 PM

Version 4
Author: jpilar | 11 June 2024, 1:08 PM

Version 3
Author: jpilar | 12 January 2024, 2:52 PM

Version 2
Author: jpilar | 10 January 2024, 2:21 PM

Version 1
Author: jpilar | 10 January 2024, 2:04 PM

Compare versions

© 2025 Cisco and/or its affiliates. All rights reserved.

CISCOU-3004

18

Network Profiles

- Template must be attached to the network profile before it is used to provision the device
- Different network profile types– switching, routing, wireless, ...
- Each profile contains a section for onboarding templates and Day-N templates
- Device-to-template mapping is managed by device series/model selection and/or tags
- Template can be assigned to a network profile in the Design->Network profile section or in the Design->CLI templates section
- Catalyst Center supports up to 50 network profiles

Day 0 (Onboarding) and Day N (Provisioning) Templates

Day 0 (Onboarding) templates

- Must be in “Onboarding Configuration” project (available by default)
- Regular or Composite templates
- Can use Velocity or Jinja language
- Import, export and clone options
- Used for **initial device configuration** (usually during plug-and-play process)
- Some **restrictions** apply
- Entire configuration **pushed at once**

Day N (Provision) templates

- Can be in any project under existing Projects
- Regular or Composite templates
- Can use Velocity or Jinja language
- Import, export, and clone options
- Used to build **complete device configuration**
- Provides flexibility
- Configuration **pushed line-by-line**

Enable and Interactive Mode Template Commands

Using enable mode commands in CLI templates

```
1 #MODE_ENABLE
2 write memory
3 #MODE_END_ENABLE
```

Beginning of enable mode block

End of enable mode block

Using interactive commands in CLI templates

```
1 #MODE_ENABLE
2 write memory
3 #INTERACTIVE
4 reload in 1<IQ>confirm<R>y
5 #ENDS_INTERACTIVE
6 #MODE_END_ENABLE
```

Beginning of interactive command

End of interactive command

- An interactive command contains the input that must be entered following the execution of a command
- To enter an interactive command in the CLI content area, use the following syntax
CLI_command<IQ>interactive_question_1<R>response_1<IQ>interactive_question_2<R>response_2
- <IQ> and <R> tags are case-sensitive and must be entered as uppercase

How to Use CLI Template for Device Configuration

- Option 1 – Provisioning device from the Inventory page
 - Provision->Inventory->Actions->Provision->Provision Device
 - Template is pushed as part of complete device provisioning as advanced configuration (Network intent, Advanced configuration, Device controllability)
 - In Tasks it is shown as “Provision Device” task
- Option 2 – Push template from the CLI template page
 - Design->CLI Templates, select Project and Template (can be regular or composite)
 - Template must be attached to Network Profile(s) (also true for Option 1)
 - Only the configuration generated by the template is pushed
 - In Tasks, it is shown as “Provision Templates” task (if not modified)

Troubleshooting Best Practices

Template Not Deployed on Target Device

- To understand where in the template we are experiencing an issue

```
event manager applet CLI_COMMANDS-->  
  event cli pattern ".*" sync no skip no  
  action 1 syslog msg "$_cli_msg"  
!  
term mon
```

- Log the lines to a file on the flash drive of the device

```
event manager applet CLI_COMMANDS-->  
  event cli pattern ".*" sync no skip no  
  action 1.0 syslog msg "$_cli_msg"  
  action 2.0 file open FH bootflash:eem_cli_commands.txt a+  
  action 2.1 file puts FH "$_event_pub_time %HA_EM-6-LOG: CLI_COMMANDS-->: $_cli_msg"  
  action 2.2 file close FH
```

- Logs to reflect the changes deployed line by line from Catalyst Center and is granular and clear where the template stopped

Using Template Simulation Capability

Available for both regular and composite templates

Catalyst Center

Design / CLI Templates

☆ 🔍 🔄 ⓘ ⚙️ | 👤 Jerry ▾

CLI Templates / Silent Trustsec Branch Switch (3) Properties

Templates **Simulation** Provision Conflicts

Simulations (1)

🔍 Search Table

0 Selected Edit Delete

☐	Simulation Name ▾	Date	Variables Simulated	Report
☐	Test1	03 September 2024	43	View

+ Add Simulation

of: Dec 17, 2024 2:49 PM 🔄

Existing simulations list

Result of last run

Creates new simulation

Using Template Simulation Capability

Select real device to run simulation against (device configuration will not be modified)

Enter/select values for template(s) variables

Save or run simulation

Using Template Simulation Capability

The screenshot displays the Cisco Catalyst Center interface for managing CLI templates. The top navigation bar shows 'Design / CLI Templates'. Below this, the 'Simulation' tab is active, showing a table of simulations. A callout box points to the 'Test1' simulation, indicating that the configuration shown on the right is generated for review and can contain debug prints.

Simulations (1)

Simulation Name	Date	Variables Simulated	Report
Test1	03 September 2024	44	View

Simulation - Test1

Variables Simulated: 44 Status:

```
786 no autostate
787 !
788 vlan 1630
789 name VN_Hospital_WAN_link
790 !
791 interface Vlan1630
792 description vrf interface to External router
793 vrf forwarding Hospital
794 ip address 192.168.112.22 255.255.255.252
795 no ip redirects
796 ip route-cache same-interface
797 !
798 router bgp 60050
799 address-family ipv4 vrf Hospital
800 redistribute connected
801 neighbor 192.168.112.21 remote-as 65535
802 neighbor 192.168.112.21 update-source Vlan1630
803 neighbor 192.168.112.21 activate
804 exit-address-family
805 !
806 !
807 !
808 !
809 vrf definition IoT
810 rd 1:4100
811 !
812 address-family ipv4
813 route-target export 1:4100
814 route-target import 1:4100
815 exit-address-family
816 !
817 vlan 1029
818 name IoT-Profiled
819 !
820 vlan 1030
821 name IoT_Quarantined
822 !
823 interface Vlan1029
824 description VN-IoT-Profiled_Devices
825 vrf forwarding IoT
826 ip address 192.168.113.184 255.255.255.248
827 ip helper-address 192.168.111.16
828 no ip redirects
829 ip route-cache same-interface
830 no autostate
831 !
832 interface Vlan1030
833 description VN-IoT-Quarantined_Devices
```

Dealing with Existing Configuration Overlap

In some cases, pushing already existing commands leads to failure

```
16  ###
17  ### Create L2NAT instance
18  !
19  l2nat instance $l2nat_instance_name
20  !
21  permit unmatched
22  ###
23  !
24  fixup $fixup_protocol
25  !
26  ###
27  ### Configuration
28  ###
```

Provision Templates 02 Jan 2025 9 00 PM

Task

PROVISION

Completed

Failed

Start: Jan 2, 2025 9:00 PM End: Jan 2, 2025 9:00 PM

As of: 9:00:59 PM [Refresh](#)

Search

IE3400-PLZ01.enprglab.local

Two VLAN L2 NAT and Profinet

NCTP10214: Provisioning failed for the template Two VLAN L2 NAT and Profinet.

Message: Unable to push the invalid CLI to the device 192.168.138.155 using protocol ssh2. Invalid CLI - 'outside from host 192.168.148.155'.

Command Output : Error. The current translated address overlaps with the translated address already configured.

Dealing with Existing Configuration Overlap

'default command' for configured entity can often help

```
16  ###
17  ### Create L2NAT instance
18  ! @start-ignore-compliance
19  !
20  l2nat instance $l2nat_instance_name
21  !
22  exit
23  !
24  default l2nat instance $l2nat_instance_name
25  !
26  ! @end-ignore-compliance
27  !
28  l2nat instance $l2nat_instance_name
29  permit unmatched in
30  permit unmatched out
31  fixup $fixup_protocol
32  ###
33  ### Configure NAT entries
34  ###
```

Provision Templates 02 Jan 2025 9 19 PM

Task • PROVISION

Completed • Success

Start: Jan 2, 2025 9:20 PM

End: Jan 2, 2025 9:20 PM

Search

IE3400-PLZ01.enprglab.local

Two VLAN L2 NAT and Profinet

```
1 !
2 vlan 1027
3 name External_VLAN
4 vlan 507
5 name Machine_VLAN
6 !
7 interface Vlan507
8 ip address 192.168.0.254 255.255.255.0
9 interface Vlan1027
10 ip address 192.168.139.61 255.255.255.224
11 ! @start-ignore-compliance
12 !
13 l2nat instance Sova-House
14 !
15 exit
16 !
17 default l2nat instance Sova-House
18 !
19 l2nat instance Sova-House
20 !
21 ! @end-ignore-compliance
22 !
23 permit unmatched
24 !
```

Removing Part of the Existing Configuration

- Creation of a new version of the template without the given command is not enough
- Several possible options for solving it:
 - Use the default (if possible) and recreate the configuration from scratch
 - Use the “no” command to remove the desired line of configuration
 - Create a section of the template to remove or swap part of configuration as needed

```
! @start-ignore-compliance
! default interface $interface.portName
! @end-ignore-compliance
interface $interface.portName
description Access -lan- Closed authentication
switchport mode access
```

```
#elseif ($Create_VN_Hospital == 'Remove')
router bgp $Branch_AS_number
no address-family ipv4 vrf Hospital
!
no interface Vlan$Hospital_users_VLAN_number
no interface Vlan$Hospital_resources_VLAN_number
no interface Vlan$Hospital_WAN_link_VLAN_number
no vlan $Hospital_users_VLAN_number
no vlan $Hospital_resources_VLAN_number
no vlan $Hospital_WAN_link_VLAN_number
no vrf definition Hospital
#end
```

Troubleshooting Template Deployments

Catalyst Center Template Common Issues

- **Undefined variables not mapped at deployment time**
 - Templates in Catalyst Center often include variable placeholders that must be defined before the template is pushed to a device. If these variables are not mapped with actual values during the deployment phase, the system cannot render the final configuration correctly, leading to errors.
- **Syntax errors or invalid commands for specific platforms:**
 - Not all Cisco devices interpret commands in the same way, and a command valid on one platform may not be recognized by another. For example, the command `ip route 0.0.0.0 0.0.0.0 Ethernet1` might work on a traditional IOS device but fail on an IOS-XE device that expects an interface to be named `GigabitEthernet1`
- **Logical errors in conditional statements (e.g., incorrect interface type)**
 - Logical errors usually stem from incorrect assumptions or misused logic in the template's Jinja2 syntax. For example, a conditional statement like `{% if interface_type == "GigabitEthernet" %}` might fail if the actual variable value is `GigE`, causing the template to skip configuration steps or apply incorrect ones
- **Overlapping or conflicting configurations from multiple templates**
 - Can create problems, especially in environments where multiple teams or workflows are managing devices. For example, one template may configure a static IP address on an interface, while another template attempts to enable DHCP on the same interface.

Catalyst Center Template Validation Tools

- **Use “Preview Template” to render configuration with test variables**

- Before deploying a template, the "Preview Template" feature allows administrators to substitute test variables and view the resulting configuration. This helps ensure that all variables are correctly defined and that the final configuration will generate as expected. For example, if a template includes a variable like `{{interface_name}}`, previewing with a value like `GigabitEthernet1` will show how the actual CLI will look on the device.

- **Validate against target device type before deploying**

- Different Cisco platforms interpret configurations differently, so it's important to validate the rendered template against the actual device model and OS version. For instance, a command like `switchport trunk allowed vlan add` may be valid on Catalyst 9300 switches but not on older Catalyst 3850 models. Validation ensures compatibility with the target device and avoids platform-specific syntax errors.

- **Catch errors in logic, missing variables, and unsupported CLI commands**

- During validation, Catalyst Center checks for common template issues such as uninitialized variables, misused conditional statements, and CLI commands not supported by the target device. For example, an if statement that references an undefined variable will trigger a validation error. Likewise, using `ip ospf network broadcast` on a device that doesn't support OSPF can be flagged before deployment.

- **Save time by catching issues before pushing to production**

- Identifying and resolving issues during the validation phase reduces the risk of failed deployments, minimizes the need for post-deployment troubleshooting, and prevents unintended service disruptions. This proactive approach not only improves deployment accuracy but also streamlines operations by ensuring that templates are fully functional before they affect production environments.

Catalyst Center Template Deployment Error Handling

- **Review deployment logs in Tools > Template Editor > Deployment History**

- Use the built-in Deployment History section under the Template Editor to access logs for past deployment attempts. This centralized log viewer provides detailed insight into what happened during the deployment process, helping identify at what stage the error occurred and what components were affected.

- **Analyze logs for CLI rejection messages, variable mismatches, or timeout errors**

- The logs can reveal several types of errors including rejected commands (e.g., syntax not recognized by the target device), variables that were not properly assigned values at runtime, or timeouts if the device failed to respond. For example, if a command such as `ip ospf 1 area 0` is invalid for the device, the log will include the rejection reason, enabling administrators to adjust the template accordingly.

- **Verify device reachability using SSH, ping, or SNMP within Catalyst Center**

- Before concluding that the issue lies within the template, confirm that the device is reachable. Catalyst Center provides tools to test SSH connectivity, ICMP ping responses, and SNMP status. A failure in any of these checks may point to network path issues, access control restrictions, or device-level misconfigurations.

- **Cross-check results with actual CLI output using manual login when necessary**

- If automated diagnostics don't clearly explain the issue, manually logging into the device via SSH to inspect the running configuration and recent command history can offer critical insights. This method can confirm whether any part of the configuration was partially applied or help verify the current device state, which may differ from the Catalyst Center's expected outcome.

Catalyst Center Template Versioning and Rollback

- **Each edit to a template automatically creates a new version**
 - Catalyst Center tracks every change made to a template by generating a version history. This allows administrators to see a full audit trail of updates, ensuring transparency and traceability in configuration changes over time.
- **Easily rollback to a prior working version**
 - If a newly deployed version of a template results in failure or undesired behavior, administrators can quickly revert to a previously known good version with just a few clicks. This capability minimizes downtime and eliminates the need to manually reconstruct previous configurations.
- **Label versions clearly for better management (e.g., “Production v2.1”, “Lab v1.3”)**
 - Using meaningful version labels makes it easier to identify the purpose and stability of each version. For example, “Lab v1.3” could be used for testing, while “Production v2.1” indicates a stable, approved configuration used in live environments. This distinction helps avoid accidental deployment of experimental templates to production devices.
- **Use the rollback feature to mitigate failed production pushes**
 - When a deployment fails or introduces instability, the rollback function provides a safety net to restore previous configurations without delay. For instance, if a new template includes a faulty routing command that breaks connectivity, the rollback option ensures quick recovery to the last known-good state without needing a full template rewrite.

Catalyst Center Template CLI Verification Commands

- **Use CLI to validate that changes were applied correctly**
 - Logging into the device manually and reviewing the configuration through the command-line interface ensures that the changes match what was defined in the template. This helps confirm success or identify any partial or failed configurations.
- **show run | include [config] – Quickly locate specific configurations**
 - This command filters the running configuration to display only lines that match a keyword or pattern. For example, after deploying a VLAN configuration, use `show run | include vlan` to confirm that the VLAN-related lines were correctly inserted into the configuration.
- **show vlan brief or show ip route – Check operational status of key features**
 - These commands verify whether services configured by the template are now active. `show vlan brief` displays all VLANs and associated interfaces, confirming that VLAN creation was successful.
- **show archive – Review previous configuration changes**
 - This command provides a versioned view of configuration files over time, allowing administrators to see what was changed by a template deployment. It's helpful for auditing and for confirming that rollback procedures returned the system to a previous state.
- **show logging – Detect command rejections or syntax errors**
 - This command reveals system logs that may include error messages related to invalid commands or unsupported configurations.

Catalyst Center Template Best Practices

- **Develop and test templates in a lab environment before production deployment**
 - Always build and validate new templates in a controlled lab setting. This helps identify syntax issues, logical errors, and device compatibility problems without affecting live services. Once verified, the template can be confidently promoted to production.
- **Use meaningful and consistent naming conventions for templates and versions**
 - Standardizing template names, such as “Branch-Switch-Onboarding” or “Core-Router-v2.4”, improves organization and allows teams to quickly locate and identify the purpose of each template. Consistent versioning also helps with rollback and change tracking.
- **Document variable definitions and expected values for each template**
 - Provide clear internal documentation or annotations within templates explaining required variables, such as IP addresses, VLAN IDs, or interface names. This prevents deployment delays caused by missing or incorrect variable mappings.
- **Regularly review and clean up outdated or unused templates**
 - Periodically audit the template library to remove or archive templates no longer in use. This reduces clutter, prevents accidental use of deprecated configurations, and keeps the workspace manageable.

Catalyst Center Template Best Practices (cont.)

- **Use version control to track changes and maintain a clear audit trail**
 - Take advantage of Catalyst Center's built-in versioning to document the evolution of a template. Add comments describing each change (e.g., "added support for Catalyst 9500 interface naming") for future reference.
- **Limit template deployment to maintenance windows when possible**
 - To avoid network disruptions, schedule template deployments during planned maintenance periods or low-traffic hours, especially for core infrastructure devices. This ensures time for validation and rollback if needed.
- **Leverage role-based access control (RBAC) to limit who can edit or deploy templates**
 - Assign template editing and deployment permissions only to authorized administrators. This prevents accidental or unauthorized changes, ensuring consistency and compliance with change management policies.
- **Continuously monitor deployments and follow up with post-deployment verification**
 - After pushing a template, verify the actual configuration on the device using either Catalyst Center's monitoring tools or manual inspection. Check for misconfigurations or differences between expected and applied settings.

Template Compliance Checks

Avoiding Compliance Check

```
11 !
12 - #if ($interface.portType == "Ethernet Port" &&
13     !$interface.description.contains("Uplink"))
14 )
15 !
16 - ! @start-ignore-compliance
17     default interface $interface.portName
18 - ! @end-ignore-compliance
19     interface $interface.portName
20     description Access -lan- Closed authentication
21     switchport mode access
22     device-tracking attach-policy IPDT_POLICY
23     dot1x timeout tx-period 7
24     dot1x max-reauth-req 3
25     source template DefaultWiredDot1xClosedAuth
26     spanning-tree portfast
27     spanning-tree bpduguard enable
28 !
29 #end
30 !
```

Beginning of section where compliance is not checked

End of section where compliance is not checked

Compliance Checks - Uppercase and Lowercase Matters

- Simply put, uppercase and lowercase letters matter to the comparator engine
- If you look at the running configuration of a switch and you see which letters or words are capitalized in specific commands, that is exactly how the commands should look in your Day-N templates.
- If you would've written that command and similar ones (like on the image below) using an all **uppercase** "VLAN", these would've been flagged as **non-compliant** because in the running configuration, only the "V" is uppercase while the rest is lowercase.

```
44 ip radius source-interface VLAN${vlan_MGMT}  
45 snmp-server trap-source VLAN${vlan_MGMT}  
46 snmp-server source-interface informs VLAN${vlan_MGMT}  
47 ntp source VLAN${vlan_MGMT}
```

Bad (all uppercase)

```
44 ip radius source-interface Vlan${vlan_MGMT}  
45 snmp-server trap-source Vlan${vlan_MGMT}  
46 snmp-server source-interface informs Vlan${vlan_MGMT}  
47 ntp source Vlan${vlan_MGMT}
```

Good (correct upper-/lowercase)

Compliance Checks – Passwords and Secrets

- **Password** and **Secrets** are used in many places in the configuration of a network device and most of the time, hopefully, these cannot be easily read by humans due to something like “**service password-encryption**” being configured, which would encrypt these passwords/secrets using Type 7-encryption.
- If you have “**service password-encryption**” enabled and you still have cleartext passwords/secrets further down in this or other templates, the password/secret would be marked as **non-compliant**.

```
16 service password-encryption
17 !
18 radius server ISE-PSN1
19   address ipv4 10.10.10.101 auth-port 1812 acct-port 1813
20   automate-tester username SW-RAD-TEST probe-on
21   key MYSECRETKEY
22   timeout 2
23   retransmit 2
```

Bad (key will not match running-config post encryption)

```
16 service password-encryption
17 !
18 radius server ISE-PSN1
19   address ipv4 10.10.10.101 auth-port 1812 acct-port 1813
20   automate-tester username SW-RAD-TEST probe-on
21   key 7 140407651F113E7A767B
22   timeout 2
23   retransmit 2
```

Good (key will match running-config)

Compliance Checks - Shortened Commands

- Do NOT shorten Commands in Day-N Templates
- Only use full/complete commands in your Day-N Templates or they will be flagged as non-compliant. Shortening commands like you probably would've done when entering them in a switch manually will not pass the compliance check.

```
1 int g1/0/1
2 swi mo acc
3 swi acc vlan 10
4 swi voi vl 666
5 swi non
```

Bad

```
1 interface GigabitEthernet1/0/1
2 switchport mode access
3 switchport access vlan 16
4 switchport voice vlan 666
5 switchport nonegotiate
```

Good

Catalyst Center Training

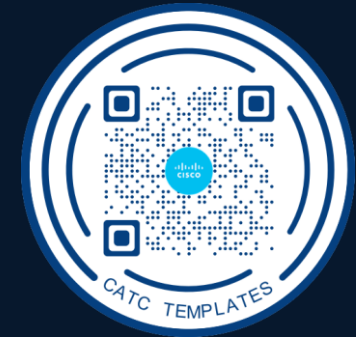
Current Technologies Computer Learning Center

Administering and Troubleshooting Cisco Catalyst Center – Assurance
(CATALYST-CENTER)

- <https://www.ctclc.com/courses/cisco/catalyst-center>

40 CE Credits / 45 CLCs

[Outline](#)



Dcloud Catalyst Center Labs

cs.co/DNAC-TEMPLATES

CiscoU Training

<https://u.cisco.com/search?query=catalyst%20center>

Complete your session evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to win 1 of 5 full conference passes to Cisco Live 2026.



Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.



Level up and earn exclusive prizes!



Complete your surveys in the Cisco Live mobile app.

Continue your education



Visit the Cisco Showcase for related demos



Book your one-on-one Meet the Engineer meeting



Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs



Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact me at: agardner@ctclc.com

Thank you

CISCO Live !

