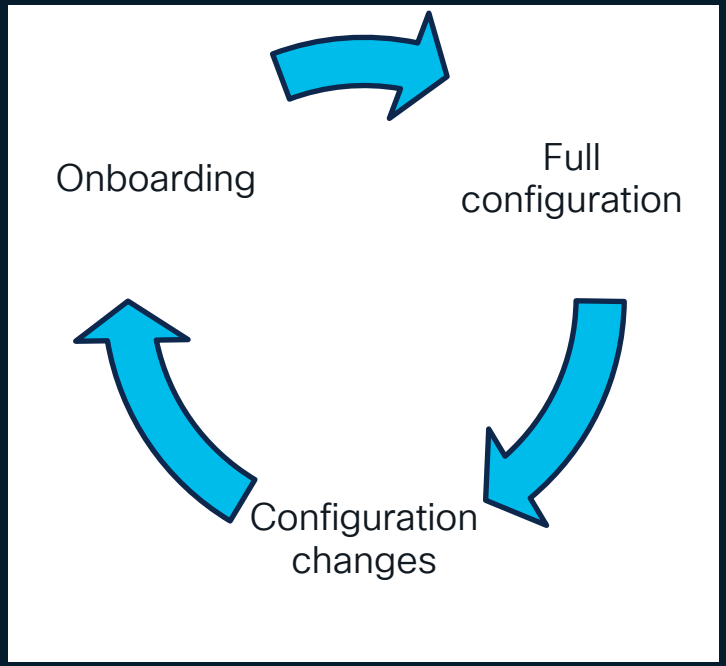
Complete Provisioning of Cisco Switches using Cisco Catalyst Center Templates

Jaromír Pilař - Technical Solutions Architect

BRKOPS-2566

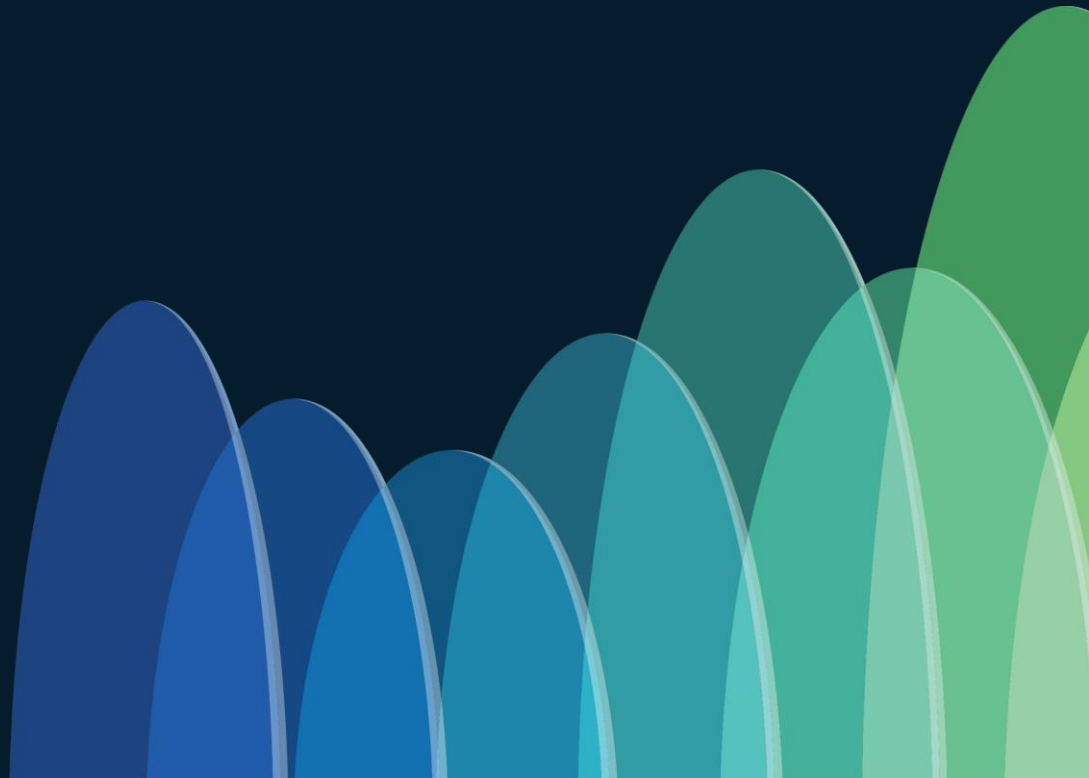
Use of Catalyst Center *CLI templates* can significantly simplify device configuration through its entire life cycle and helps to maintain configuration consistency.



Agenda

- Introduction
- How to create CLI templates
  - CLI templates basics
  - CLI templates advanced topics
- How to use CLI templates
  - Device onboarding
  - Device configuration
- Summary and references

Introduction



Few words about me

- Joined Cisco in 1996
- Based in Prague, Czech Republic
- Worked on different technical positions (SE, CSE, TSA) in country and regional teams
- CCIE #2910 (since 1997)
- Focus on enterprise technologies and multidomain (switching, SD Access, TrustSec, Common Policy etc.)
- Married with two children
- Enjoy trekking, skiing, biking and also astronomy and physics

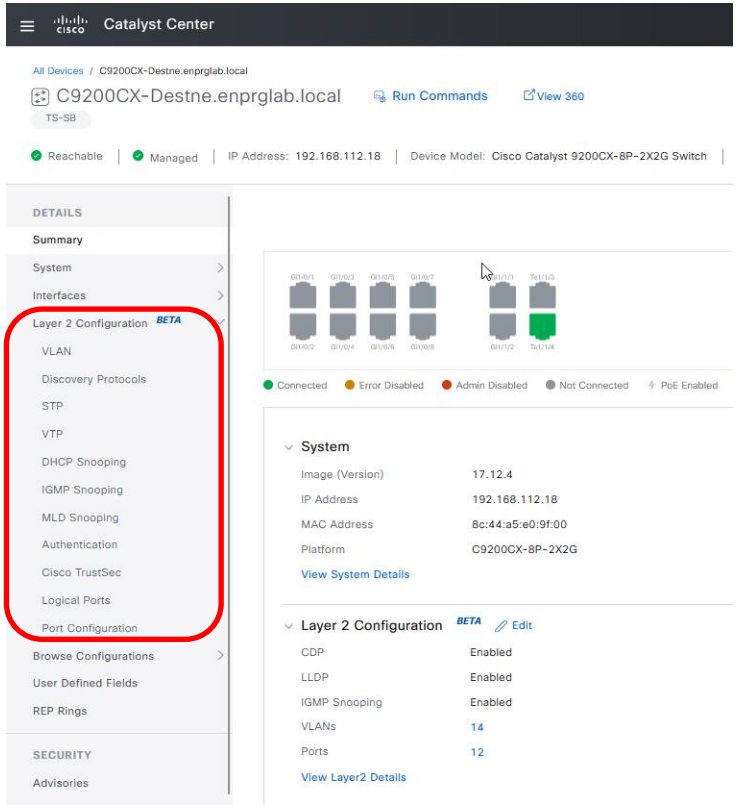
How the story began

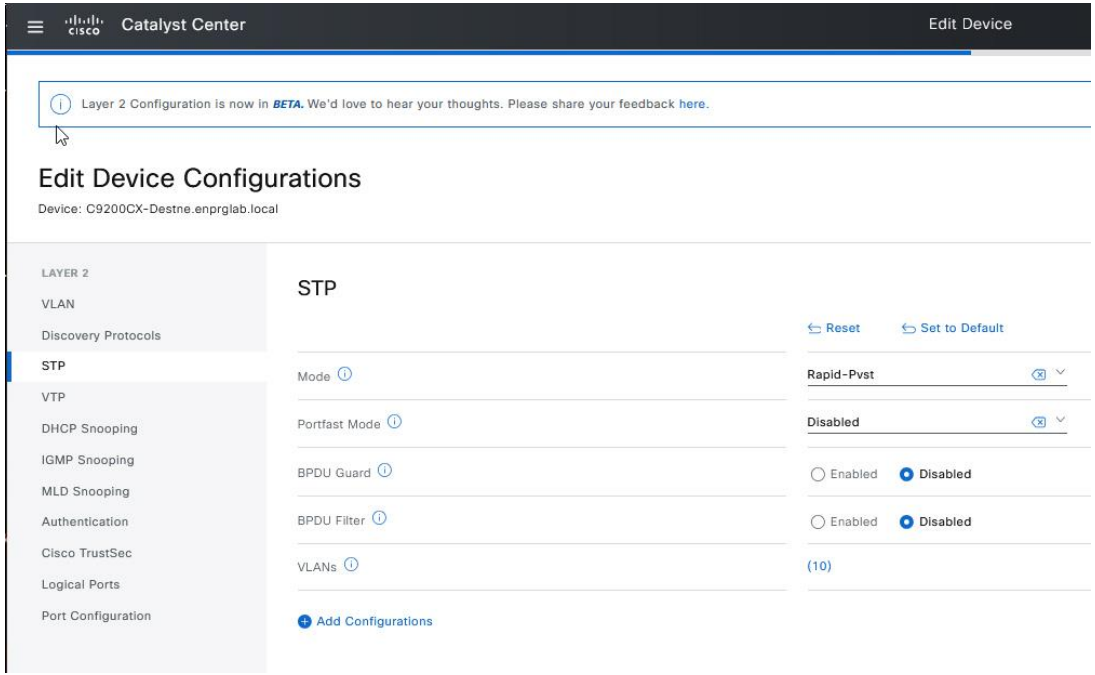
- Few years ago, during Christmas time with bad weather ☺
- First use case:
  - C3650/3850 imitating SD Access PEN configuration
- Limited “old fashioned” programming skills (Fortran77, Pascal, PDP11 and VAX11 assembler)
- Other use cases (examples):
  - Silent branch with VNs and TrustSec (C9200CX)
  - TrustSec branch with C9200/C9200L
  - Industrial switch (IE3400) with TrustSec, L2 NAT, CCV Sensor

Catalyst Center option to configure network

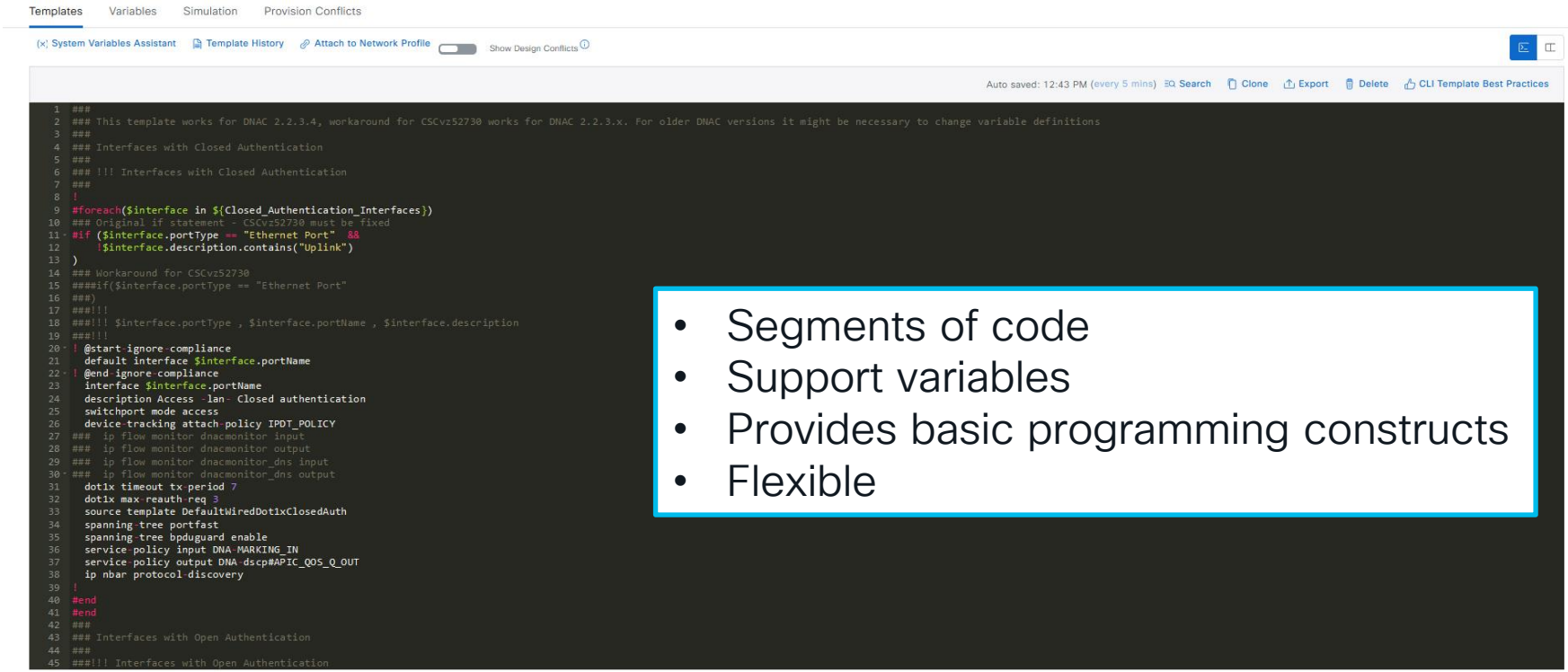
- L2 configuration from Catalyst Center GUI for switches (in public beta)
- CLI templates
- Workflows
- Use of API

L2 configuration from GUI for switches (in Beta)



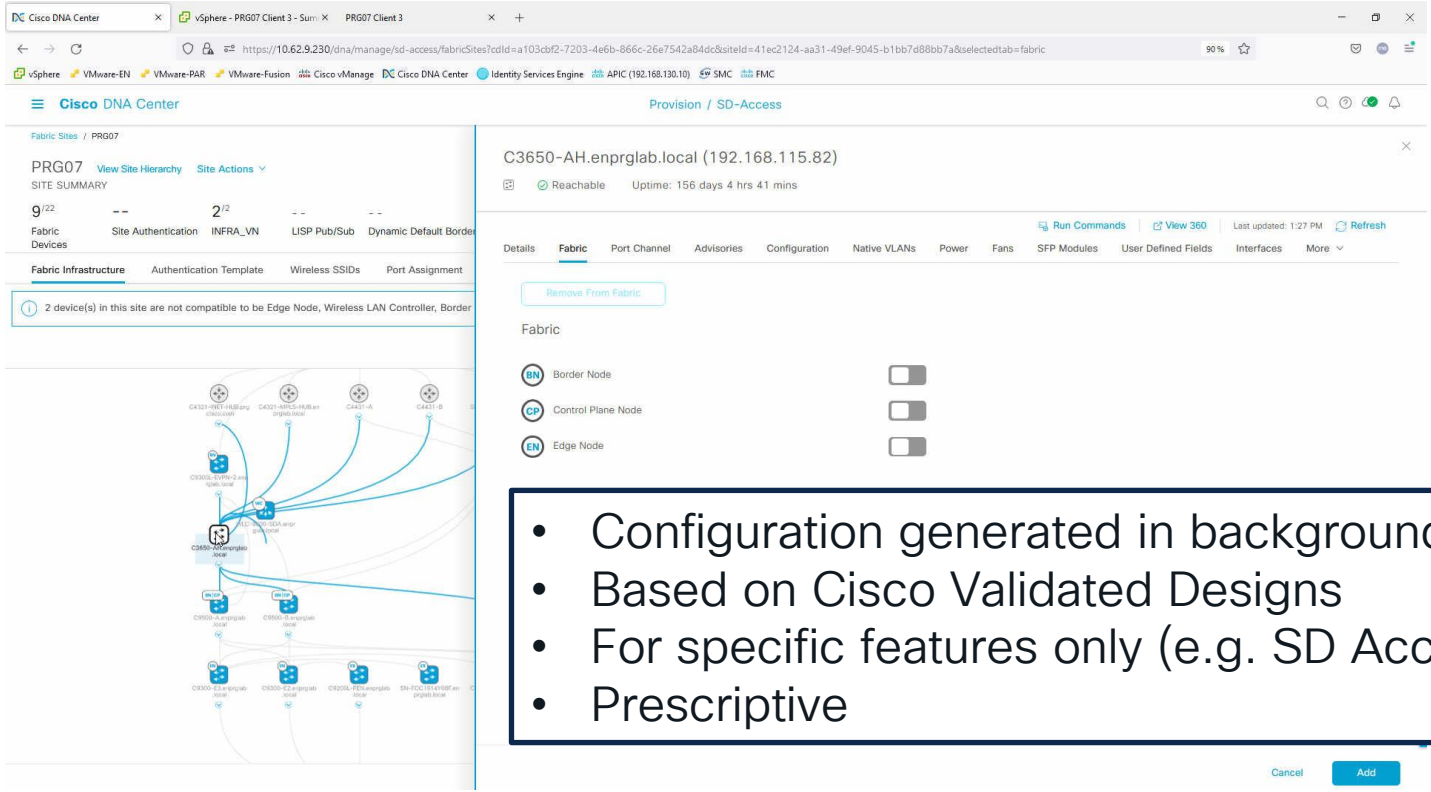


Configuration templates



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

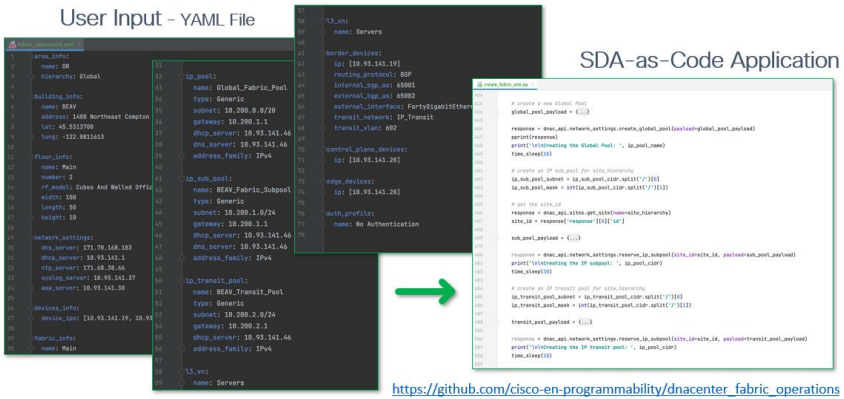
Workflow



- Configuration generated in background
- Based on Cisco Validated Designs
- For specific features only (e.g. SD Access)
- Prescriptive

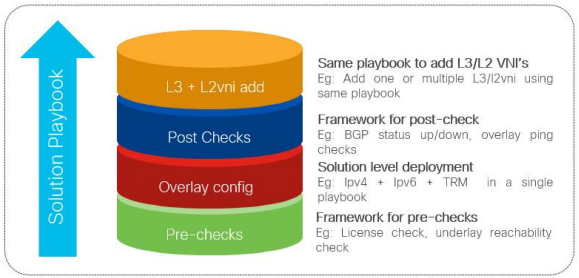
Using API

API - SDA-as-Code Application Files



https://github.com/cisco-en-programmability/dnacenter_fabric_operations

EVPN Ansible – Solution Playbook

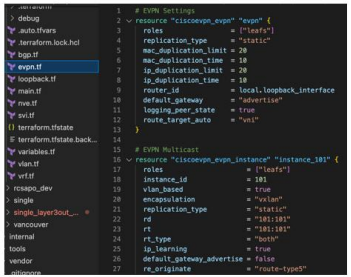


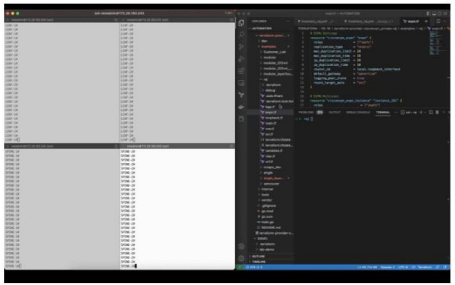
Simple to Use

- Single playbook for complete solution
- Single inventory file to add Leaf/Spine variables

https://github.com/CiscoDevNet/Cat9kEVPN-Ansible

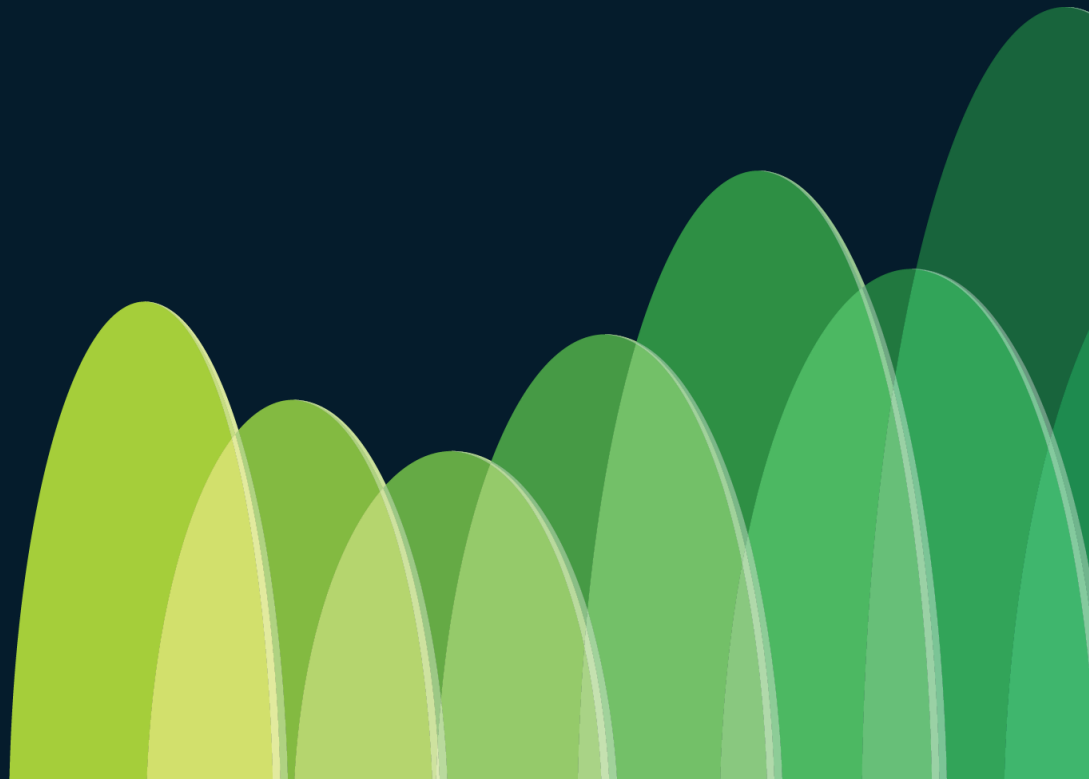
EVPN Automation with Terraform





Terraform Provider
(https://registry.terraform.io/providers/robertcsapo/ciscoevpn/1.0.1)
Terraform Examples
(https://github.com/netascode/terraform-iosxe-evpn-examples)

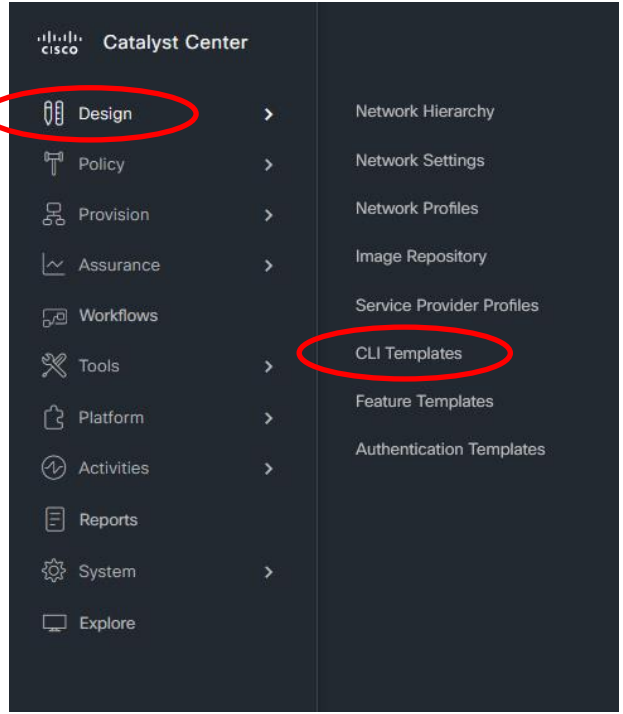
How to create CLI templates - basics



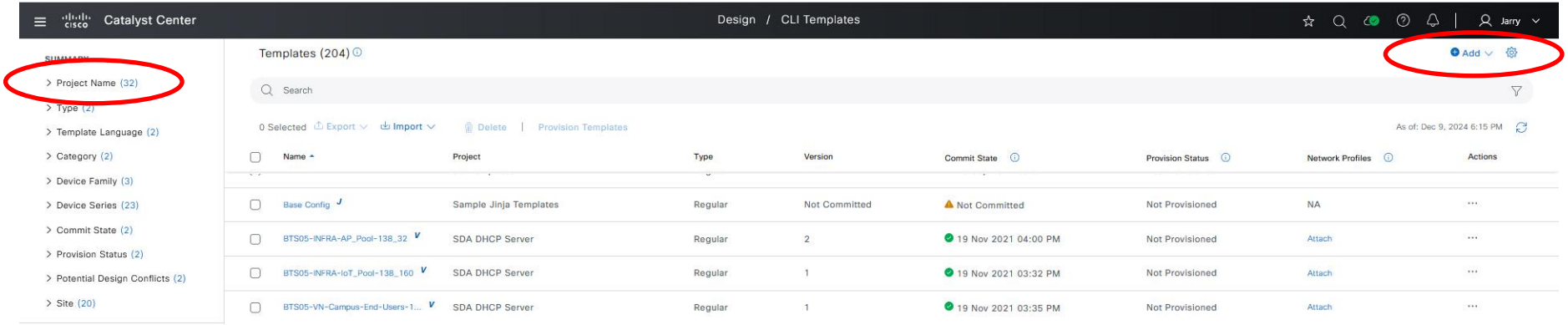
cisco Live!

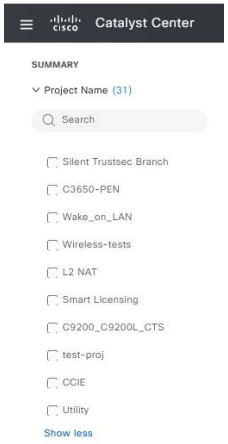
Where to start

- Currently under Design menu
  - Design -> CLI templates
- In previous versions
  - Tools -> Template Hub
  - Tools -> Template Editor



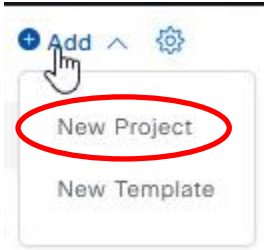
Projects and templates





Projects:

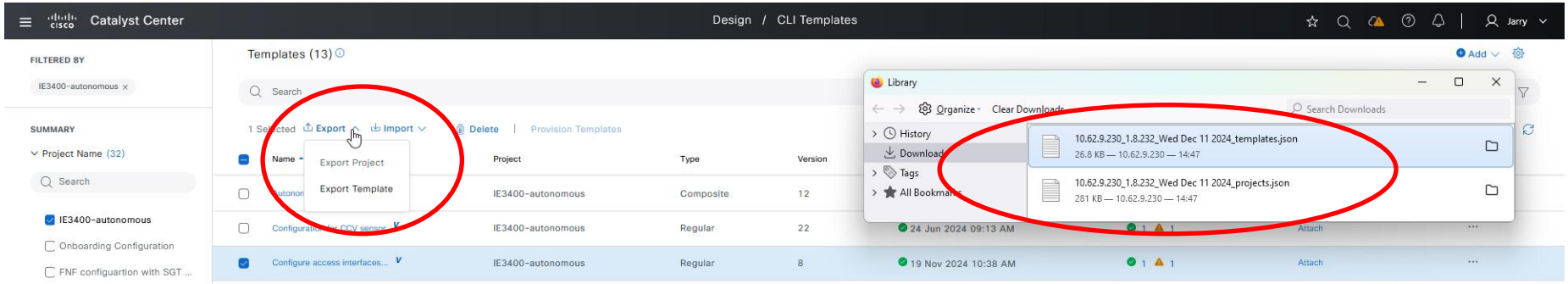
- "Folders" where templates are organized in structure
- Can be exported and imported



Adding Projects:

- Name
- Description (optional)

Projects and templates – export and import



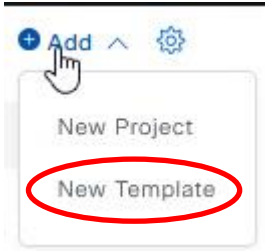

```
[
  {
    "name": "IE3400-autonomous",
    "description": "",
    "tags": [],
    "templates": [
      {
        "name": "Two VLAN L2 NAT and Profinet",
        "description": "",
        "tags": [],
        "author": "jpilar",
        "deviceTypes": [
          {
            "productFamily": "Switches and Hubs",
            "productSeries": "Cisco Catalyst IE3400 Rugged Series",
            "productType": "Cisco Catalyst IE-3400-8P2S Rugged Switch"
          }
        ],
        "softwareType": "IOS",
        "softwareVariant": "XE",
        "templateContent": "###\n###\n### Create Machine and External VLANs\n!\nvlan $vlani
\ninterface Vlan$vlanid_profinet\nip address $int_gw_addr $int_gw_mask\ninterface Vlan$vlan
\nl2nat instance $l2nat_instance_name\n!\nexit\n!\ndefault l2nat instance $l2nat_instance_n
deprecated\n### instance-id 1\n###\n###\n!\npermit unmatched\n###\n!\nfixup $fixup_protocol
\n#set($Outside_List_SRC = [])\n#set($Outside_List_DST = [])\n#set($Outside_List_NotEmpty =
false)\n###\n#set($Inside_List_SRC = [])\n#set($Inside_List_DST = [])\n#set($Inside_List_No
$Outside.replace(\" \",\"\"))\n#set($Outside = $temp)\n#set($temp = $Outside_GW.replace(\"
(!$Outside.isEmpty())\n #set($Outside_List = $Outside.split(\";\"))\n #set($n = $Outside_Li
#set($Outside_List_NotEmpty = $Outside_List_SRC.add($pair[0]))\n  #set($Outside_List_NotEmp
$Outside_GW.split(\";\"))\n #set($n = $Outside_GW_List.size() - 1)\n #foreach($i in [0..$n]
$Outside_GW_List_SRC.add($pair[0]))\n  #set($Outside_GW_List_NotEmpty = $Outside_GW_List_DS
#set($n = $Inside_List.size() - 1)\n #foreach($i in [0..$n])\n  #set($pair = $Inside_List[$
$Inside_List_DST.add($pair[1]))\n  #end\n#end\n###\n!\n#if (!$Outside.isEmpty())\n #set($n
```

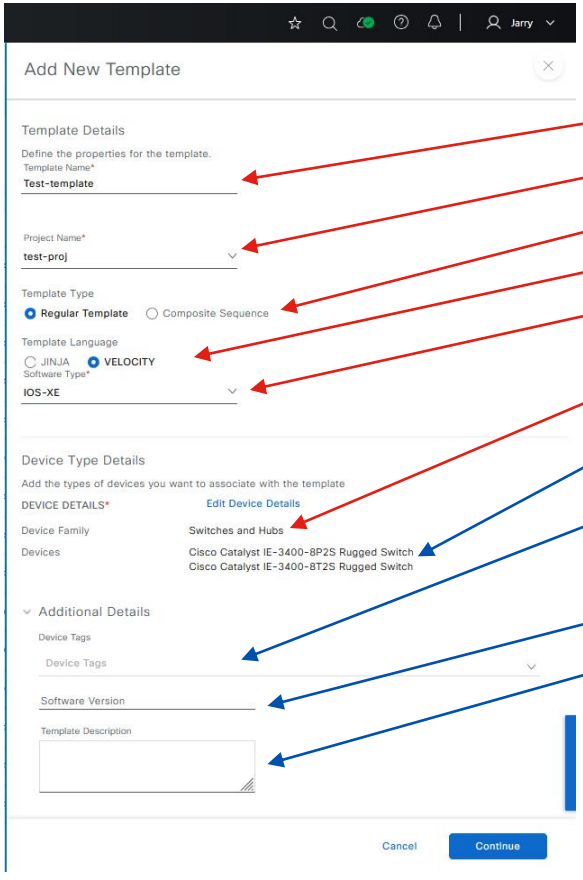

Project


```
[
  {
    "name": "Configure access interfaces and trunks",
    "description": "",
    "tags": [
      {
        "id": "8d046150-a014-4220-8309-b47d7516c9a1",
        "name": "C3k-PEN"
      }
    ],
    "author": "jpilar",
    "deviceTypes": [
      {
        "productFamily": "Switches and Hubs",
        "productSeries": "Cisco Catalyst IE3400 Rugged Series",
        "productType": "Cisco Catalyst IE-3400-8P2S Rugged Switch"
      }
    ],
    "softwareType": "IOS",
    "softwareVariant": "XE",
    "templateContent": "###\n### This template works for DNAC 2.2.3.4, workaround for CSCvz
\n### Interfaces with Closed Authentication\n###\n### !!! Interfaces with Closed Authentica
must be fixed\n#if ($interface.portType == \"Ethernet Port\"  &&\n    !$interface.descripti
\"\n###)\n###!!!\n###!!! $interface.portType , $interface.portName , $interface.description
Closed authentication\n  switchport mode access\n  device-tracking attach-policy IPDT_POLIC
dot1x max-reauth-req 3\n  source template DefaultWiredDot1xClosedAuth\n  spanning-tree port
dscp#APIC_QOS_Q_OUT\n###  ip nbar protocol-discovery\n!\n#end\n#end\n###\n### Interfaces wi
${Open_Authentication_Interfaces})\n### Original if statement - CSCvz52730 must be fixed\n#
Workaround for CSCvz52730\n####if($interface.portType == \"Ethernet Port\"\n###)\n###!!!\n#
$interface.portName\n  interface $interface.portName\n  description Access -lan- Open authe
dnacmonitor input\n###  ip flow monitor dnacmonitor output\n  dot1x timeout tx-period 7\n
bpduguard enable\n###  service-policy input DNA-MARKING_IN\n###  service-policy output DNA-
\n###\n### Initial settings\n###\n#set($Access VLAN List = [])\n#set($SGT List = [])\n#set(
```


Template

Adding CLI template





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

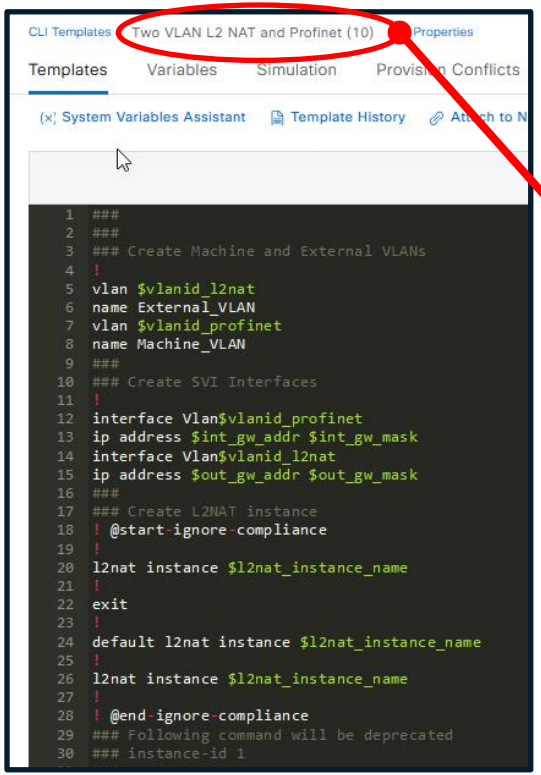
- Generates real configuration segments
- Can use variables
- Can use:
  - Source binding
  - Manipulation of variables
  - Conditional branches
  - Loops
  - Methods

Composite

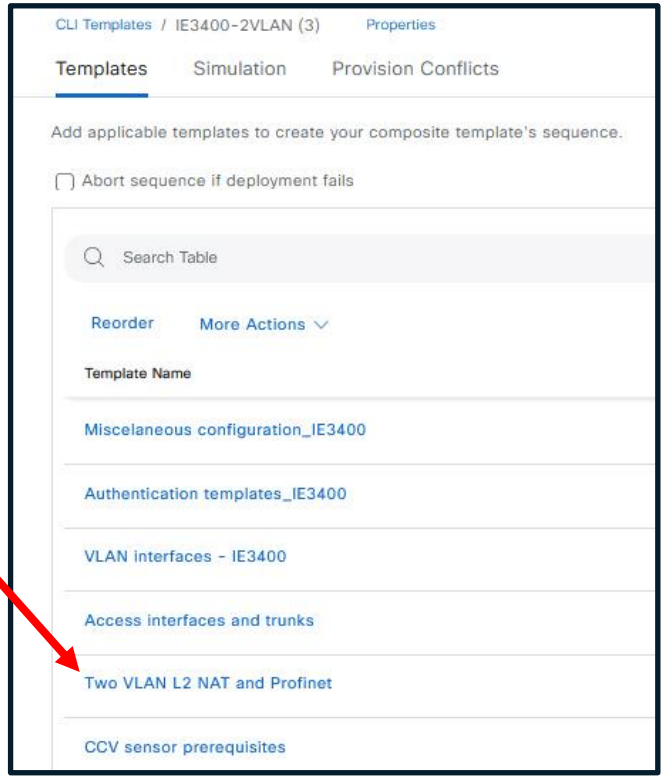
- Sequence of regular templates
- Sequence can be reordered
- Sequence can be aborted if deployment fails

CLI template types – regular vs. composite

Regular



Composite



CLI template languages

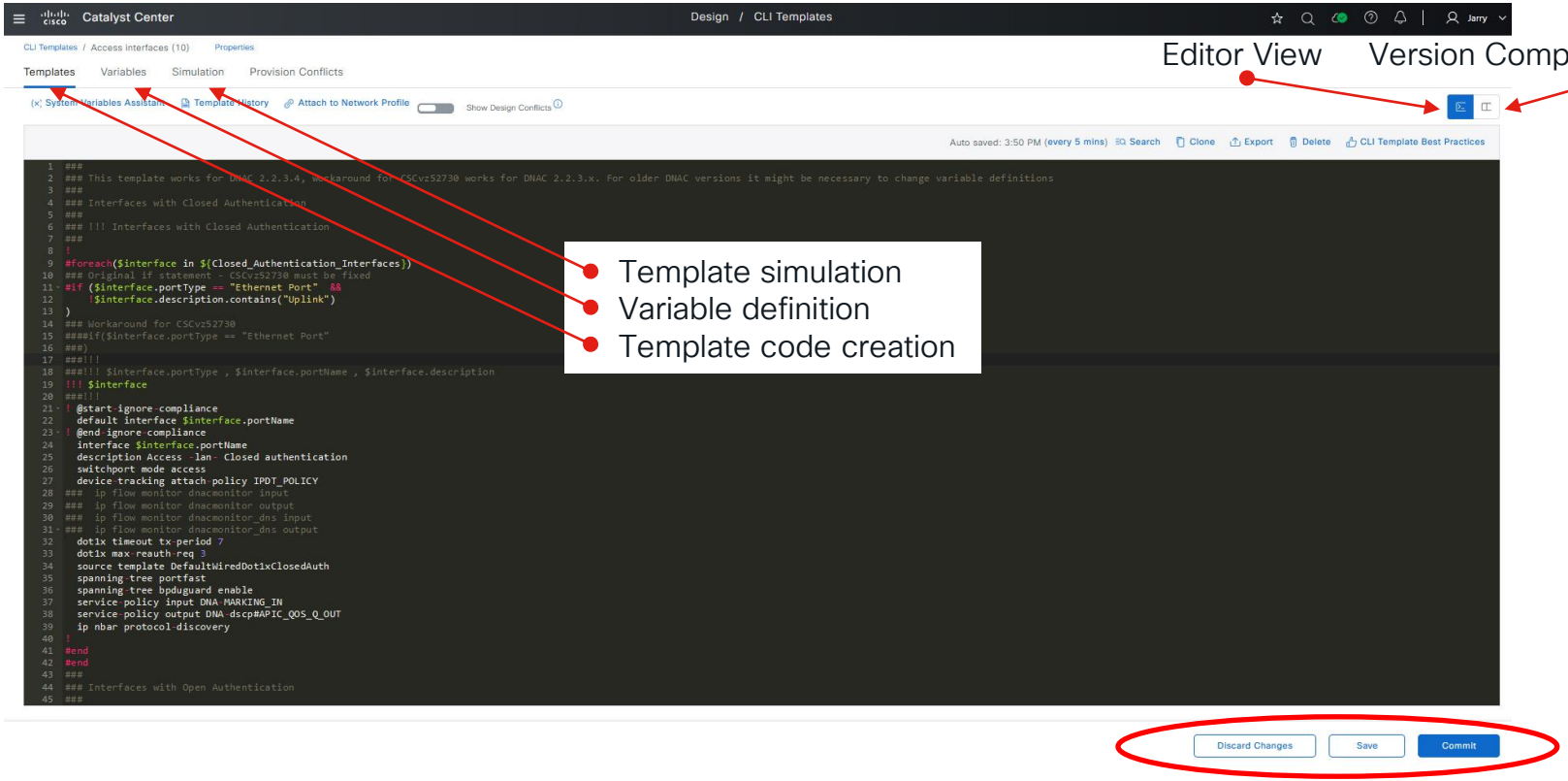
Apache Velocity Language

- 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

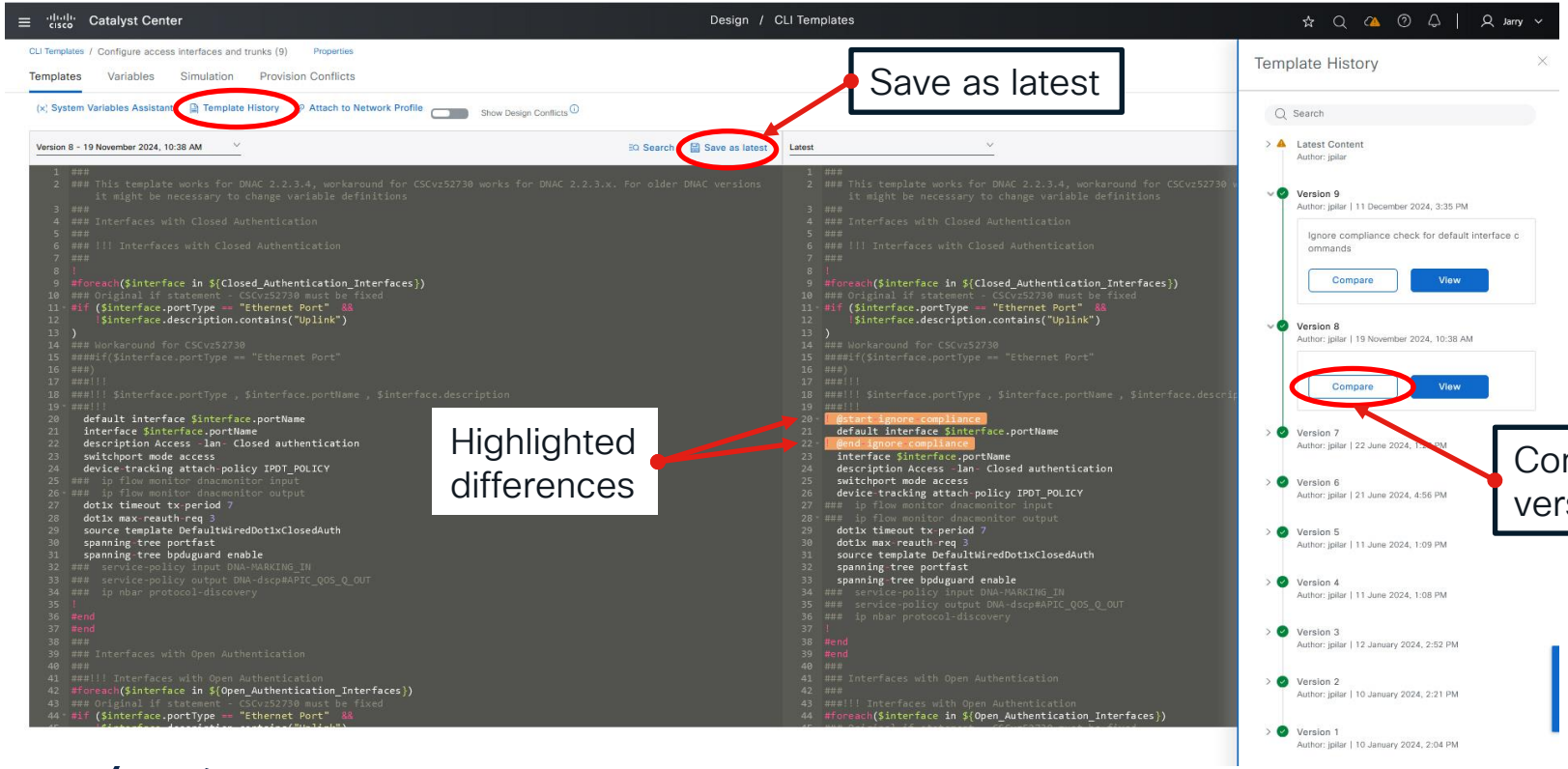
Jinja

- 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

CLI template editor



CLI template editor - history



CLI template variables

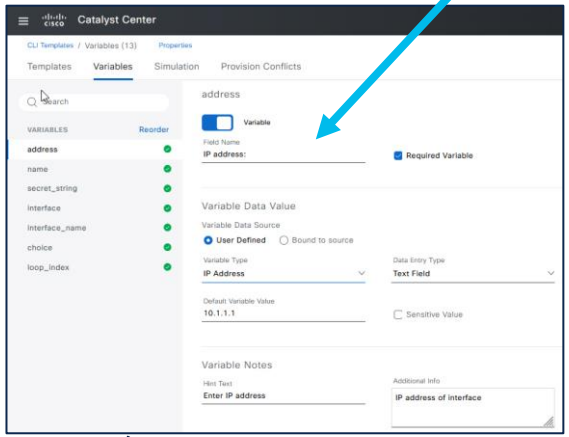
- Used to provide additional information to make use of it more flexible and usable for different devices in different sites
- Variables can be of different types:
  - User defined or bound to source (e.g. inventory) or system
  - Single value (string, integer, IP address, MAC address)
  - Entered by user or selected from predefined options (single select or multiselect)
  - Required or optional
  - Variable name starts with “$” or “$__” (system variable)
- Variables can be entered manually or imported for csv file

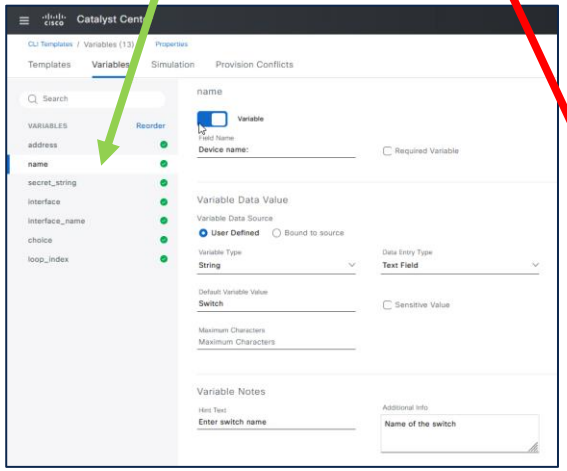
CLI template variables

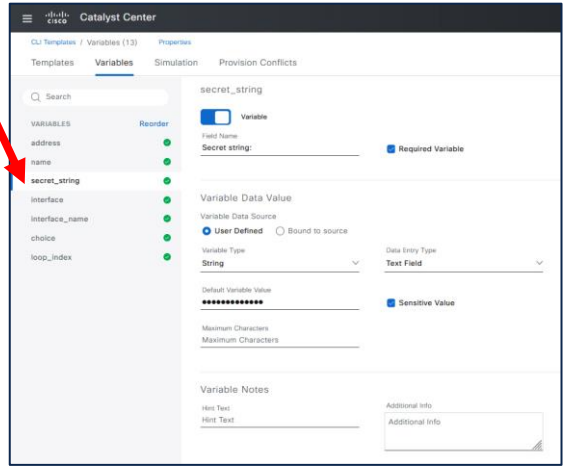
User defined – value defined and entered by user

```
!
! Template variables (user defined)
!
! User defined type IP address
$address
! User defined type string
$name
! User defined type string (private)
$secret_string
!
```

```
!
! Template variables (user defined)
!
! User defined type IP address
10.1.1.1
! User defined type string
Switch
! User defined type string (private)
####
!
```





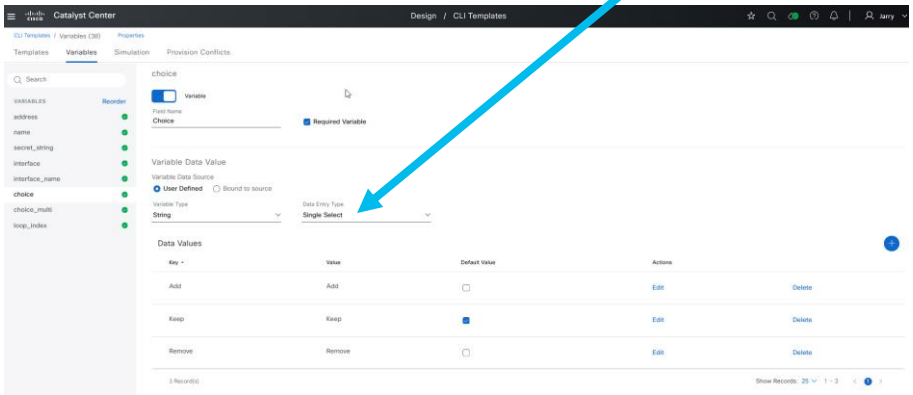


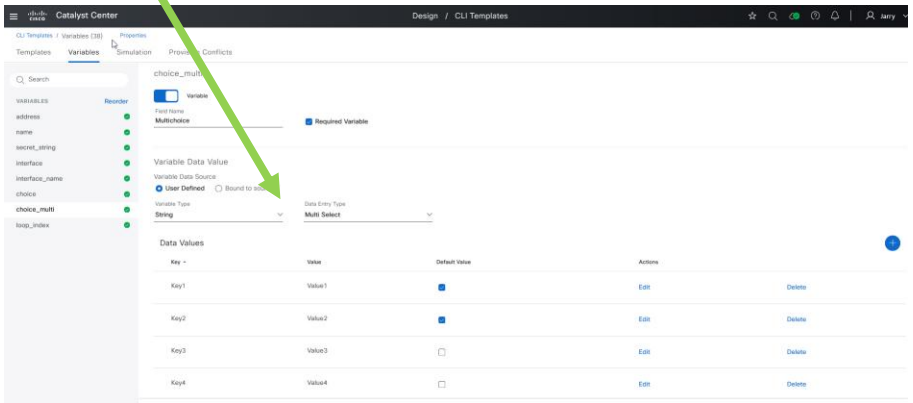
CLI template variables

User defined – value defined and selected by user (single or multiselect)

```
!
! Template variables (single and multiselect)
!
! User defined (single select)
$choice
!
! User defined (multiselect)
$choice_multi
```

```
! Template variables (single and multiselect)
!
! User defined (single select)
Add
!
! User defined (multiselect)
[Value1, Value3]
```



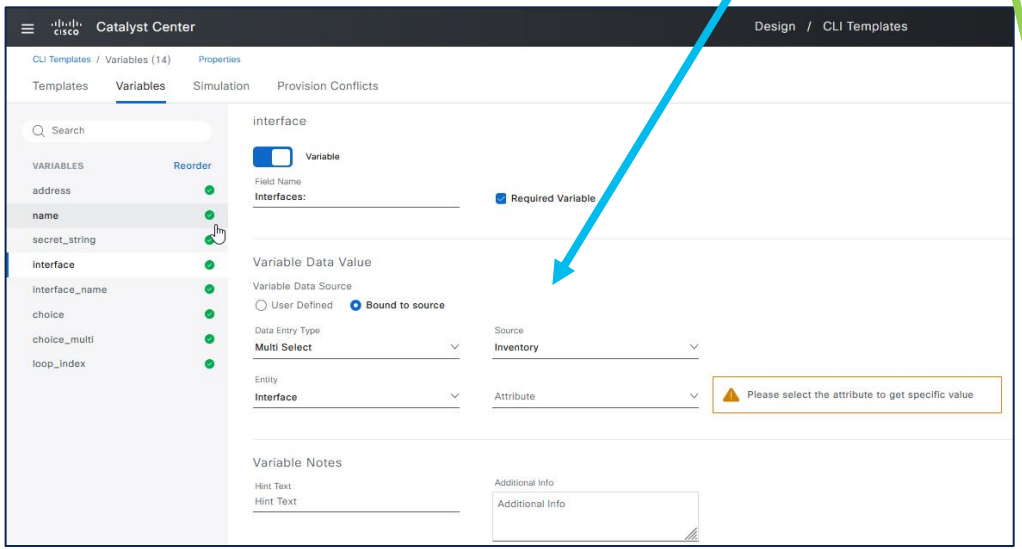


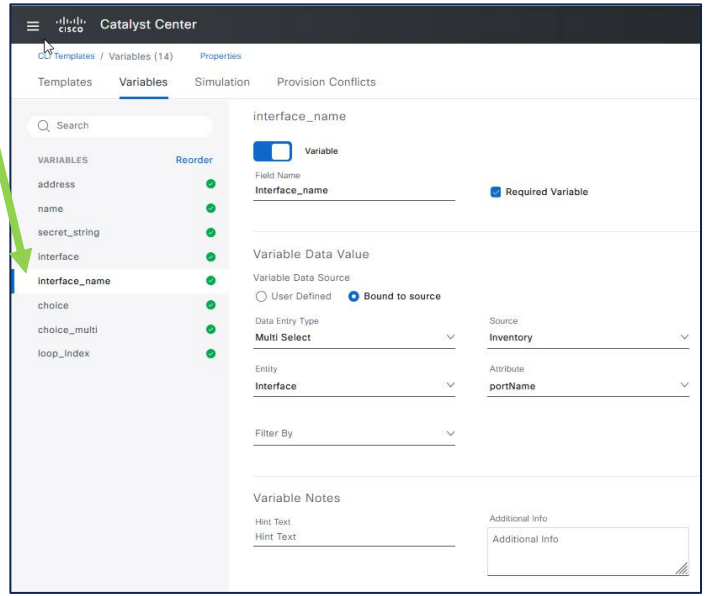
CLI template variables

Bound to source (Inventory, Interface) – structure and single attribute

```
! Template variables (bound to source)
!
! Bound to source - structure (multiselect)
#foreach ( $member in $interface )
 $member.portName , $member.portType , $member.description , $member.ifIndex
#end
!
! Bound to source - single attribute (multiselect)
$interface_name
```

```
! Template variables (bound to source)
!
! Bound to source - structure (multiselect)
GigabitEthernet1/2 , Ethernet Port , Uplink-to-Fabric-Edge, 2
GigabitEthernet1/1 , Ethernet Port , Access -lan- Trunk, 1
!
! Bound to source - single attribute (multiselect)
[GigabitEthernet1/3, GigabitEthernet1/4]
```





Example - Attributes for bind to interface

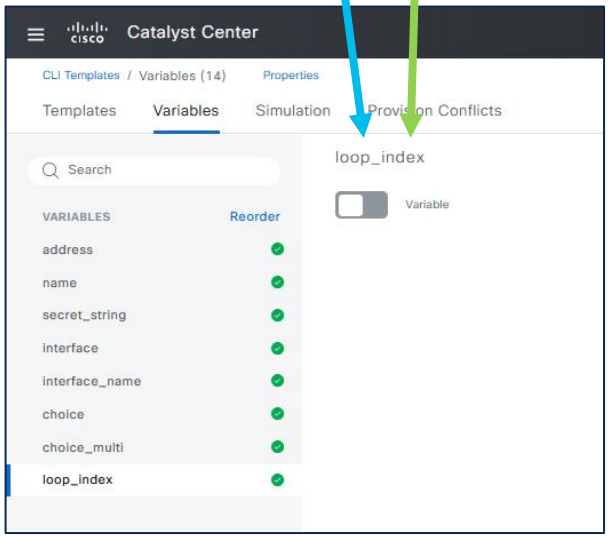
- These is the list of attributes supported for Entity Interface.
- This can be retrieved using API “/dna/intent/api/v1/interface”
- Platform->Developer Toolkit-> API-> Know Your Network
- Some useful parameters:
  - portName
  - portType
  - portMode
  - description (can be used to encode some information)

CLI template variables

Auxiliary variable (not exposed outside of template code)

```
29 ! Local variables (not exposed outside of template)
30 !
31 $loop_index
32 #set ($loop_index = 25)
33 ! $loop_index after value of 25 is assigned
34 $loop_index
```

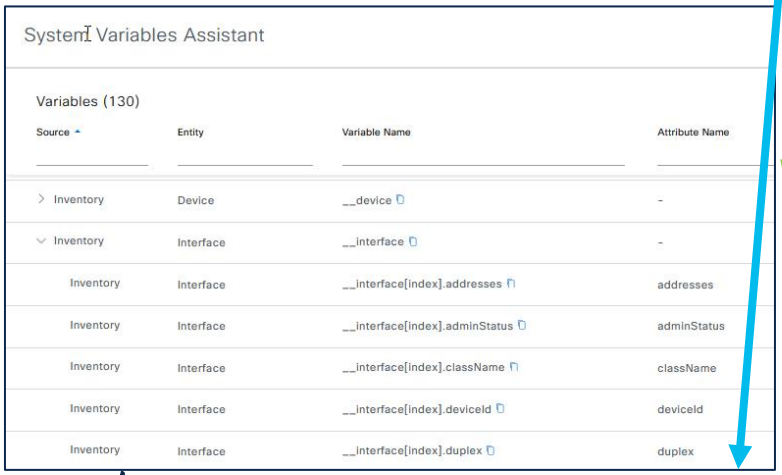
```
28 ! Local variables (not exposed outside of template)
29 !
30 $loop_index
31 ! 25 after value of 25 is assigned
32 25
```



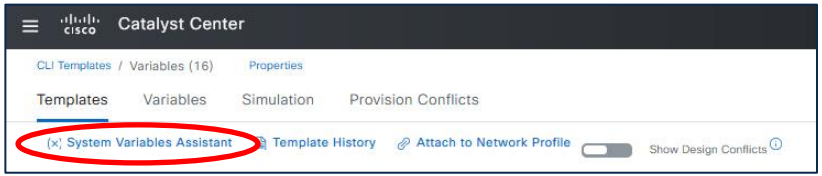
CLI template variables

System variables

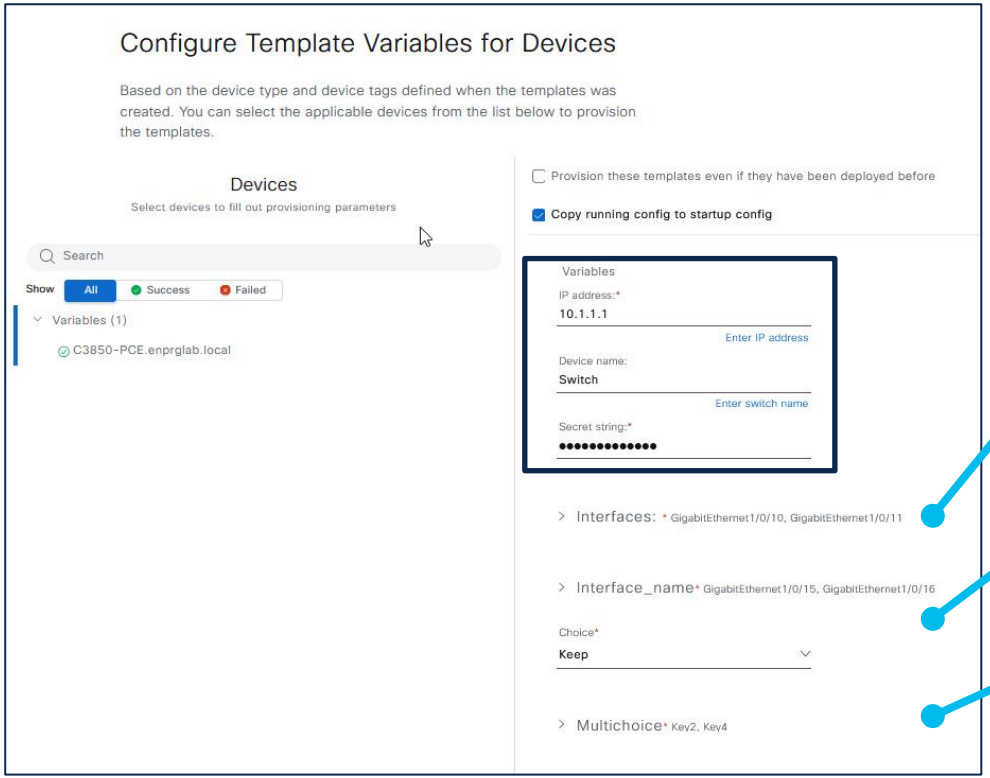
```
!
! System variables
!
! System variable - structure
#foreach ( $member in $__interface )
 $member.portName , $member.portType , $member.description
#end
!
! System variable - single attribute
$__device.hostname
!
```

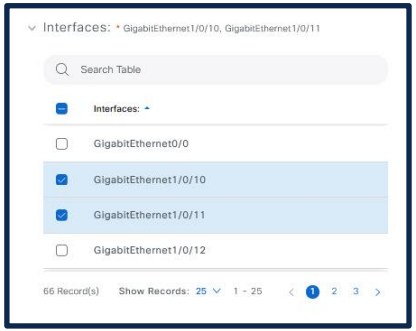


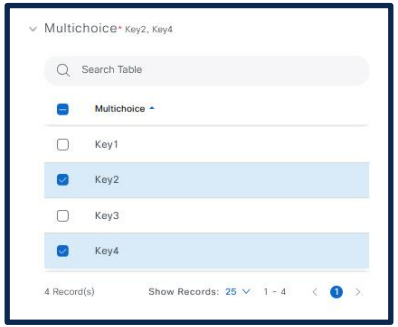

```
!
! System variables
!
! System variable - structure
GigabitEthernet1/10 , Ethernet Port ,
GigabitEthernet1/8 , Ethernet Port ,
GigabitEthernet1/2 , Ethernet Port , Uplink-to-Fabric-Edge
GigabitEthernet1/6 , Ethernet Port , Access -lan- Closed authentication
GigabitEthernet1/3 , Ethernet Port , Access -lan- No authentication
Vlan507 , Ethernet SVI ,
AppGigabitEthernet1/1 , Ethernet Port , Apphosting interface
GigabitEthernet1/9 , Ethernet Port ,
Vlan1023 , Ethernet SVI ,
GigabitEthernet1/5 , Ethernet Port , Access -lan- Closed authentication
GigabitEthernet1/7 , Ethernet Port ,
Vlan1 , Ethernet SVI ,
Vlan1027 , Ethernet SVI ,
GigabitEthernet1/1 , Ethernet Port , Access -lan- Trunk
GigabitEthernet1/4 , Ethernet Port , Access -lan- No authentication
!
! System variable - single attribute
IE3400-PLZ01.enprglab.local
!
```



CLI template variables – entering values







CLI template variables

Setting values and operators (integer)

```
49  !
50  ! Setting variable values
51  !
52  ! Integer
53  ! Setting value
54  #set ($loop_index = 25)
55  ! $loop_index
56  ! Addition (+ 1)
57  #set ($X = $loop_index + 1)
58  ! $X
59  ! Subtraction (- 1)
60  #set ($X = $loop_index - 1)
61  ! $X
62  ! Multiplication (* 2)
63  #set ($X = $loop_index * 2)
64  ! $X
65  ! Division (div 5)
66  #set ($X = $loop_index / 5)
67  ! $X
68  ! Remainder (mod 4)
69  #set ($X = $loop_index % 4)
70  ! $X
71  !
```

```
59  !
60  ! Setting variable values
61  !
62  ! Integer
63  ! Setting value
64  ! 25
65  ! Addition (+ 1)
66  ! 26
67  ! Subtraction (- 1)
68  ! 24
69  ! Multiplication (* 2)
70  ! 50
71  ! Division (div 5)
72  ! 5
73  ! Remainder (mod 4)
74  ! 1
75  !
```

CLI template variables

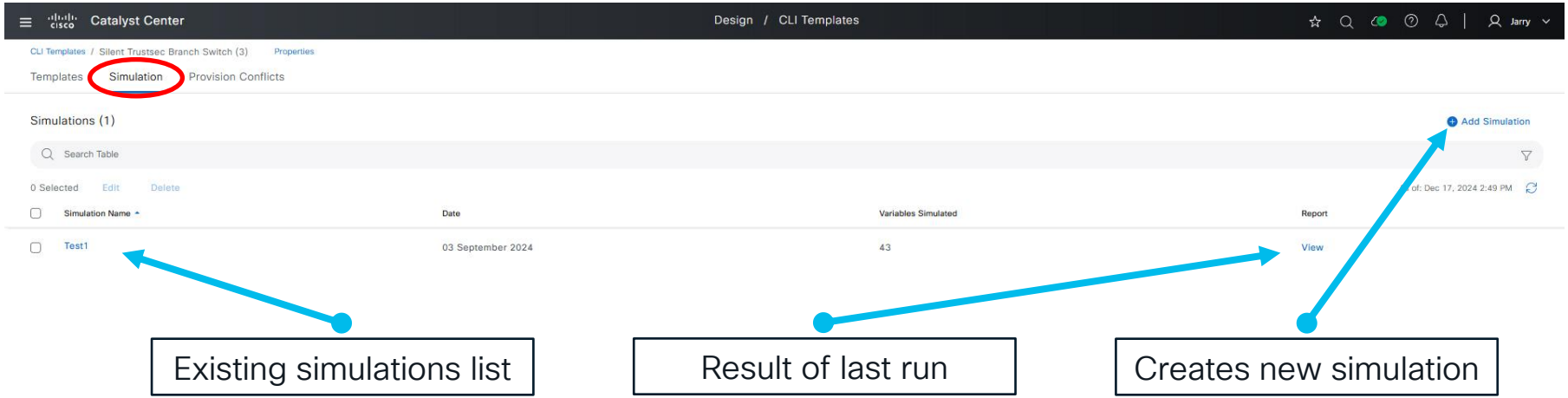
Setting values and operators (boolean and string)

```
!
! Boolean
! Setting value
#set ($T = true)
! $T
#set ($F = false)
! $F
! Logical NOT
#set ($Y = !$T)
! $Y
! Logical AND
#set ($Y = $T && $F)
! $Y
! Logical OR
#set ($Y = $T || $F)
! $Y
!
! String
#set ($Str = "Hello world")
! $Str
!
```

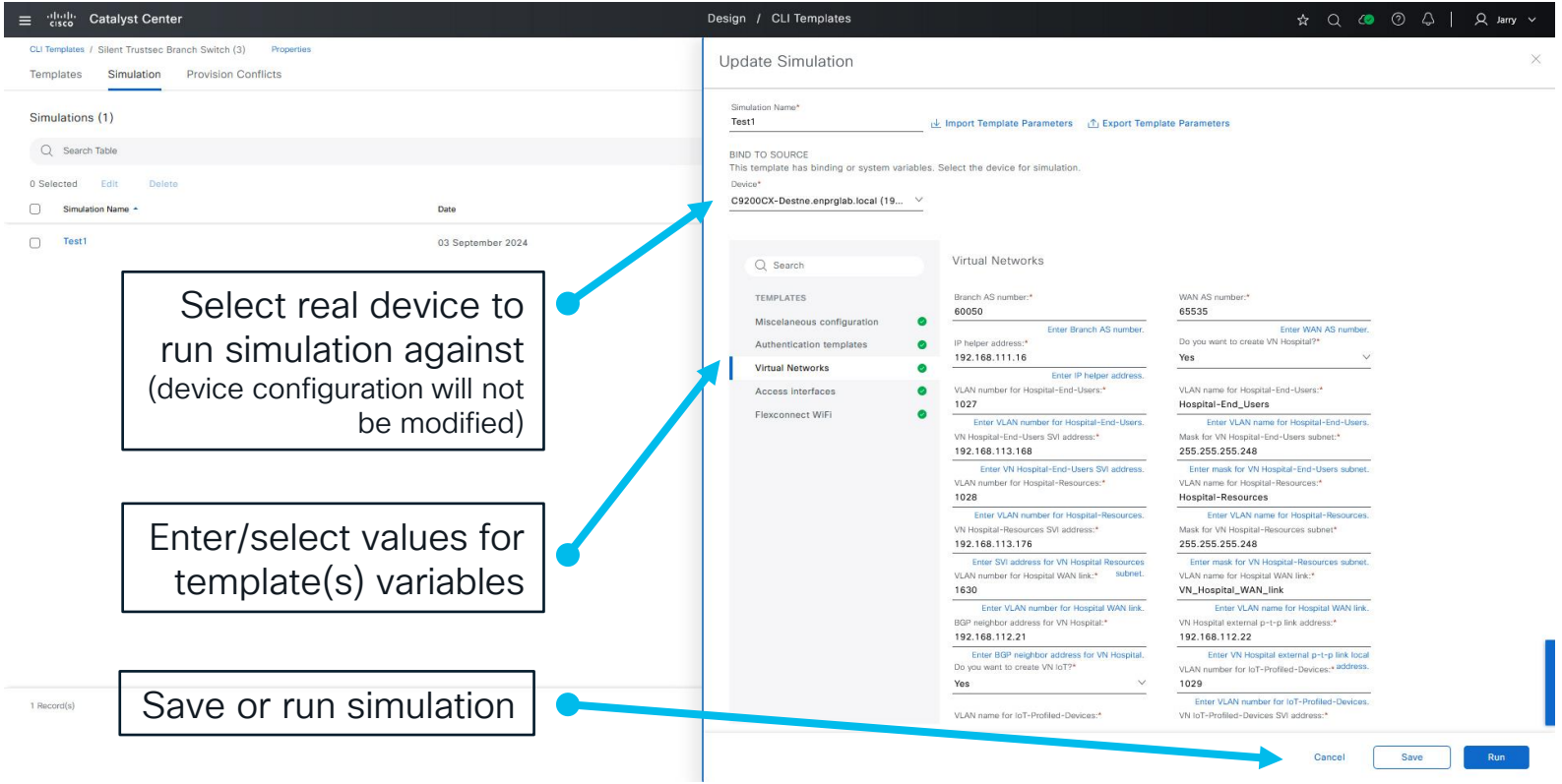
```
!
! Boolean
! Setting value
! true
! false
! Logical NOT
! false
! Logical AND
! false
! Logical OR
! true
!
! String
! Hello world
!
```

Using template simulation capability

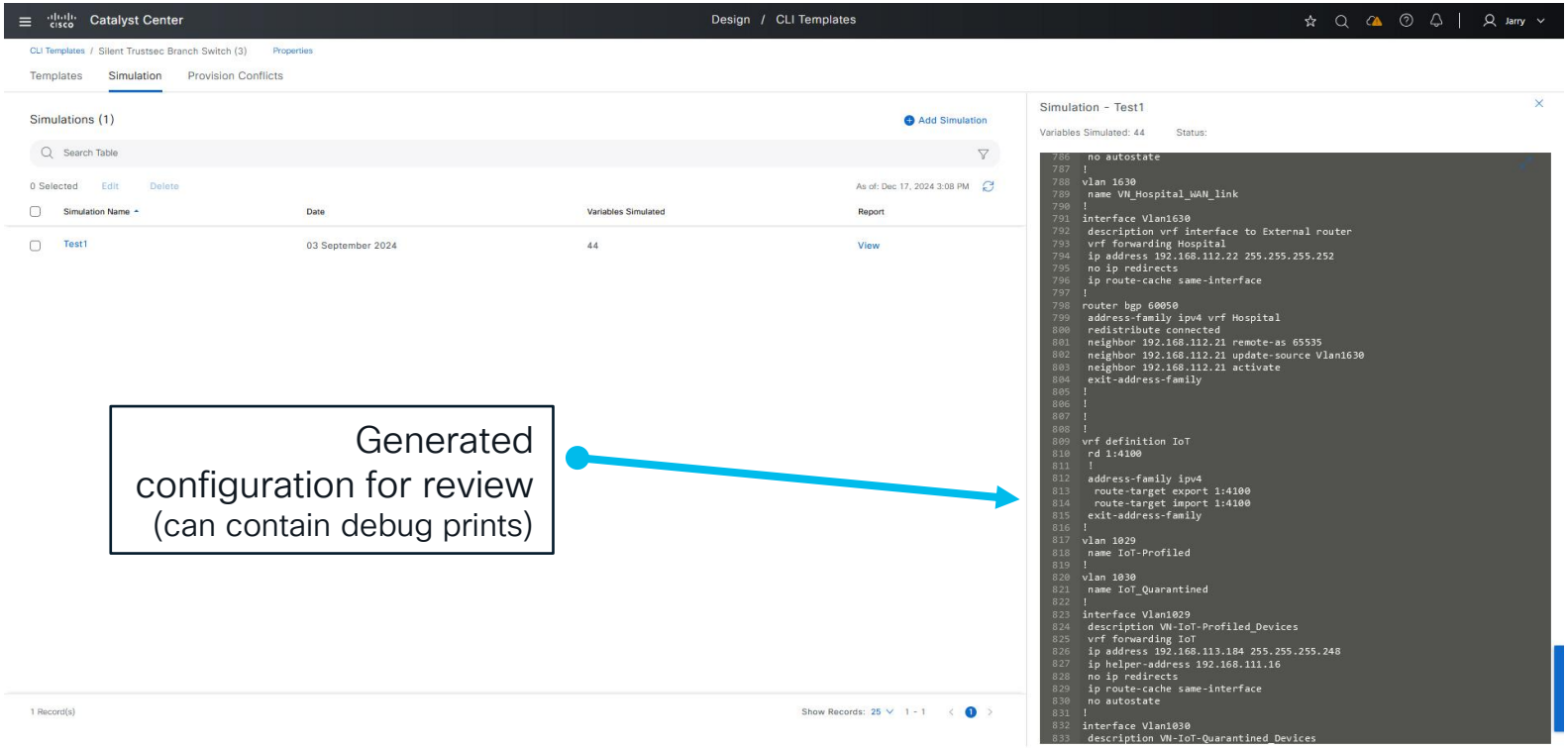
Available for both regular as well as composite templates



Using template simulation capability



Using template simulation capability



Generated configuration for review (can contain debug prints)

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**

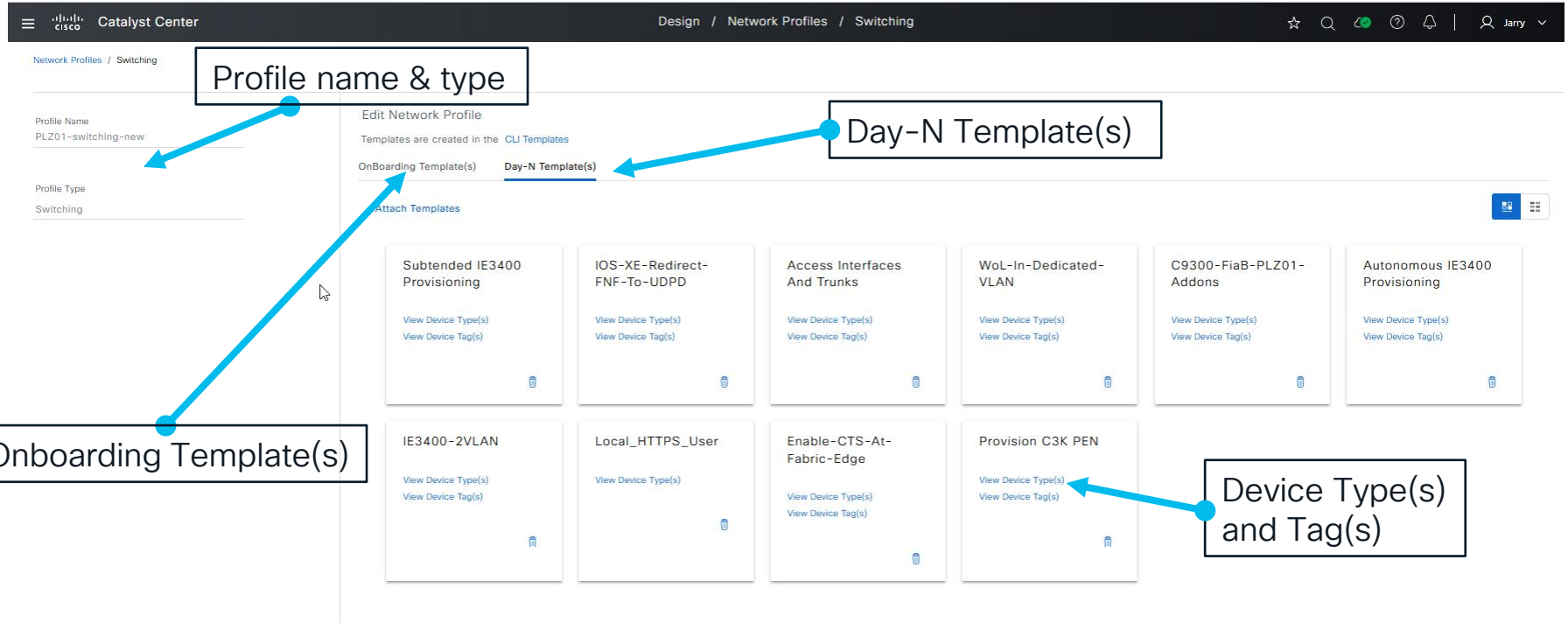
Day N (Provision) templates

- 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**

Network profiles

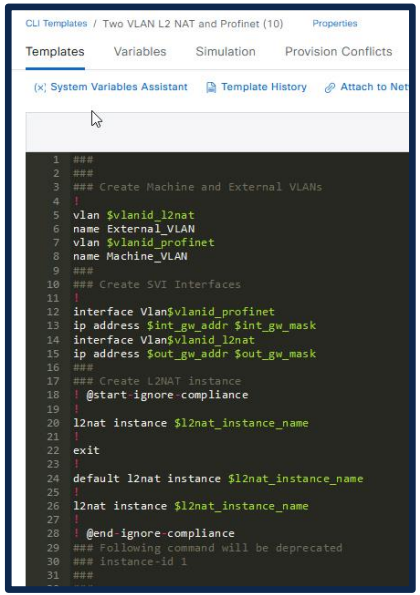
- Template must be attached to network profile before it is used to provision device
- Different network profile types- switching, routing, wireless, …
- Each profile contains section for onboarding templates and Day-N templates
- Device to template mapping is managed by device series/model selection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

Network profile (example)

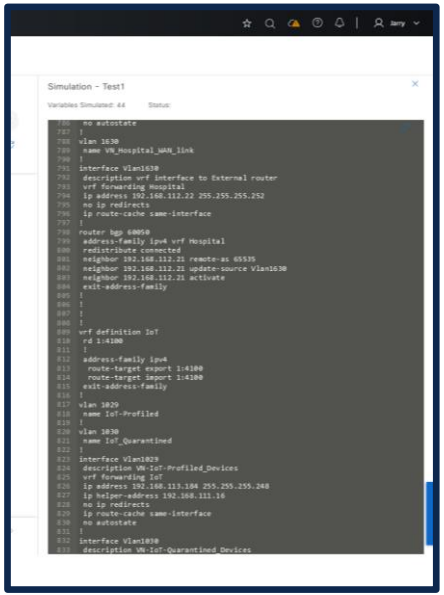


Steps to make CLI template ready to be used

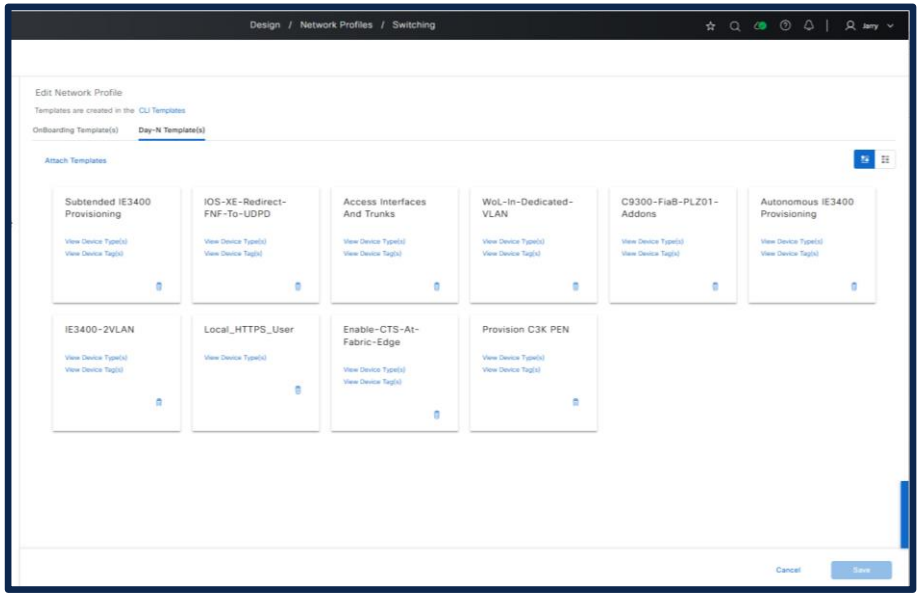
Create project and templates



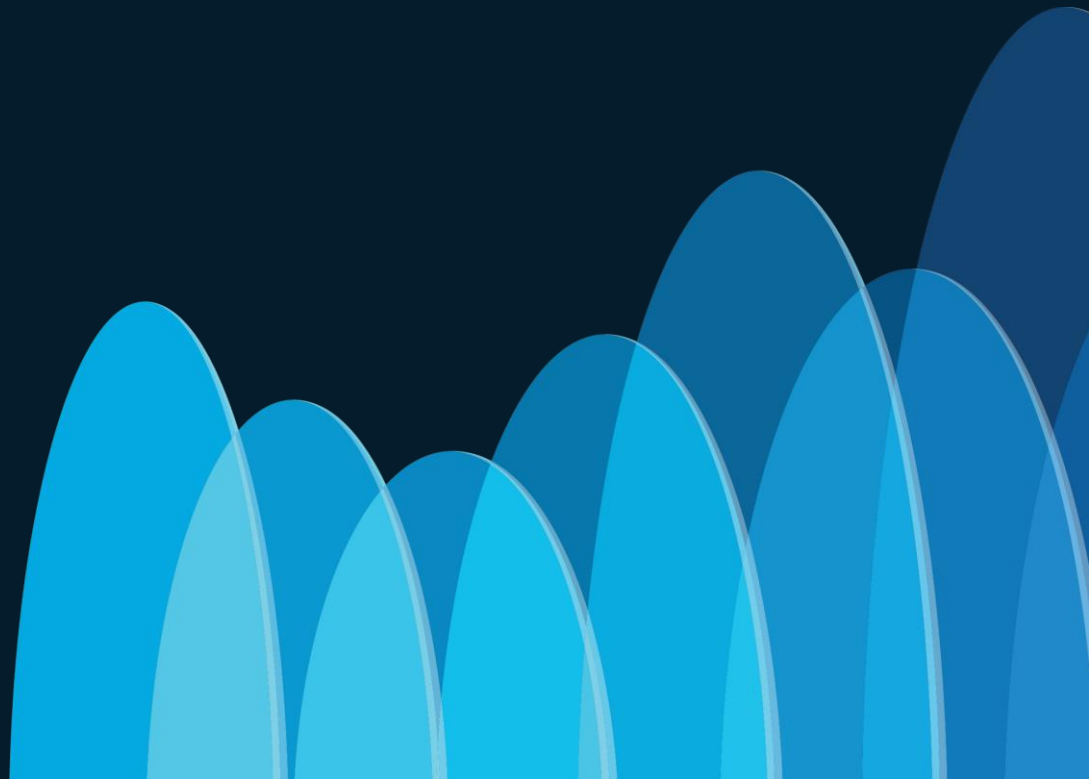
Simulate and test functionality



Assign template to Network profile



How to create CLI templates – advanced topics



cisco Live!

Conditionals (if – elseif – else)

```
1  !
2  ! x = $x , y = $y
3  #if( $x < 10 )
4  !
5  ! Choice 1 (if)
6  !
7  #elseif( $x == 10 )
8  !
9  ! Choice 2 (first elseif)
10 !
11 #elseif( $y == 6 )
12 !
13 ! Choice 3 (second elseif)
14 !
15 #else
16 !
17 ! Choice 4 (else - no previous condition met)
18 !
19 #end
```

```
1  !
2  ! x = 8 , y = 6
3  !
4  ! Choice 1 (if)
5  !
```

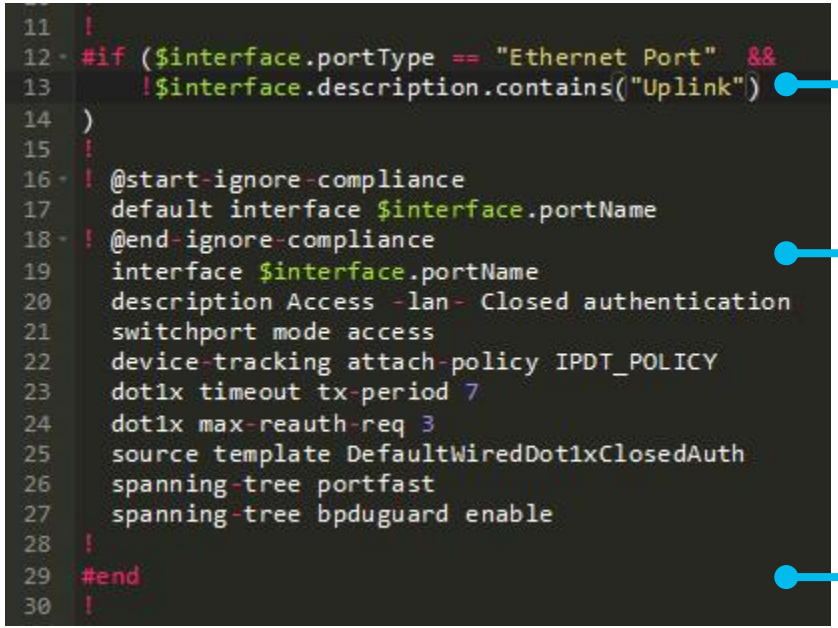
```
1  !
2  ! x = 10 , y = 6
3  !
4  ! Choice 2 (first elseif)
5  !
```

```
1  !
2  ! x = 11 , y = 6
3  !
4  ! Choice 3 (second elseif)
5  !
```

```
1  !
2  ! x = 11 , y = 7
3  !
4  ! Choice 4 (else - no previous condition met)
5  !
```

First match wins

Conditionals (example)



```
!
#if ($interface.portType == "Ethernet Port"  &&
    !$interface.description.contains("Uplink")
)
!
! @start-ignore-compliance
  default interface $interface.portName
! @end-ignore-compliance
  interface $interface.portName
  description Access -lan- Closed authentication
  switchport mode access
  device-tracking attach-policy IPDT_POLICY
  dot1x timeout tx-period 7
  dot1x max-reauth-req 3
  source template DefaultWiredDot1xClosedAuth
  spanning-tree portfast
  spanning-tree bpduguard enable
!
#end
!
```

Start of conditional block. Condition is met if port is ethernet port, **and** description does **not** contain string “Uplink”

Block of configuration generated if condition is met, if condition is not met nothing is generated

End of conditional block

Foreach loops

```
!
! 1 to 5 loop
!
#foreach( $i in [1..5] )
! Run $i
#end
!
! Loop for each element
!
#foreach( $interface in $interfaces )
!    $interface.portName
#end
!
! Loop with break
!
#foreach( $interface in $interfaces )
 #if( $foreach.count > 3 )
   #break
 #end
!    $interface.portName
#end
```

```
!
! 1 to 5 loop
!
! Run 1
! Run 2
! Run 3
! Run 4
! Run 5
!
! Loop for each element
!
!    GigabitEthernet1/1
!    GigabitEthernet1/2
!    GigabitEthernet1/3
!    GigabitEthernet1/4
!    GigabitEthernet1/5
!    GigabitEthernet1/6
!
! Loop with break
!
!    GigabitEthernet1/1
!    GigabitEthernet1/2
!    GigabitEthernet1/3
```

Foreach loop (example)

```
!
#foreach($interface in ${Closed_Authentication_Interfaces})
!
#if ($interface.portType == "Ethernet Port"  &&
    !$interface.description.contains("Uplink")
)
!
! @start-ignore-compliance
  default interface $interface.portName
! @end-ignore-compliance
  interface $interface.portName
  description Access -lan- Closed authentication
  switchport mode access
  device-tracking attach-policy IPDT_POLICY
  dot1x timeout tx-period 7
  dot1x max-reauth-req 3
  source template DefaultWiredDot1xClosedAuth
  spanning-tree portfast
  spanning-tree bpduguard enable
!
#end
#end
!
```

Loop for all interfaces selected into multiselect variable bound to inventory

Conditional block from previous example (starts with #if, ends with first #end)

End of loop

Working with strings

- Many ways how to manipulate with strings like:
  - Test if string contains substring – *$interface.description.**contains**("Uplink")*
  - Replace characters by others in string – *$SGTs.**replace**(" ","")*
  - Split string into array by separator – *$SGT_List = $SGTs.**split**(",")*
  - Compare two strings – *(($NoAuthInt_List[$i].**compareTo**($NoAuthInt_List[$j]) > 0)*
  - More options at:
    - https://docs.oracle.com/javase/7/docs/api/java/lang/String.html or https://docs.oracle.com/javase/8/docs/api/java/lang/String.html

Encoding values into string

- Multiple ways how to encode values into string
  - Single value in string (e.g. ‘startup_vlan’)
  - Multiple values of single type in single strings (e.g. ‘vlan1, vlan2, …, vlanx’)
  - Multiple values of different types in single strings (e.g. ‘addr1,mask1; addr2,mask2; …; addrx,maskx’)
  - Multiple values of different types in multiple strings and then match by position (e.g. ‘addr1, addr2, …, addrx’ and ‘mask1, mask2, …, maskx’)

Working with strings – example

Generating static routes

```
### Initial settings
#set($Subnets_Subnet = [])
#set($Subnets_Mask = [])
#set($Subnets_List_NotEmpty = false)
### Convert Subnets and Masks input into arrays
#set($temp = $Subnets.replace(" ",""))
#set($Subnets = $temp)
###
#if (!$Subnets.isEmpty())
 #set($Subnets_List = $Subnets.split(";"))
 #set($n = $Subnets_List.size() - 1)
 #foreach($i in [0..$n])
  #set($pair = $Subnets_List[$i].split(","))
  #set($Subnets_List_NotEmpty = $Subnets_Subnet.add($pair[0]))
  #set($Subnets_List_NotEmpty = $Subnets_Mask.add($pair[1]))
 #end
#end
### Generate static route commands
#if (!$Subnets.isEmpty())
 #set($n = $Subnets_Subnet.size() - 1)
 #foreach($i in [0..$n])
ip route $Subnets_Subnet[$i] $Subnets_Mask[$i] $next_hop
#end
#end
```

Variable values:
$Subnets (string)
1.0.0.0, 255.0.0.0; 2.1.0.0, 255.255.0.0; 3.1.1.0, 255.255.255.0
$next_hop (string)
5.1.1.1

```
C3560CX-PRG07#sho running-config | include route
 remark netvmg-traceroute
C3560CX-PRG07#
010663: Jan  2 11:59:22.462: %SYS-5-CONFIG_I: Configured from console by dnac on vty2 (192.168.111.230)
010664: Jan  2 11:59:24.147: %SYS-5-CONFIG_I: Configured from console by dnac on vty2 (192.168.111.230)
C3560CX-PRG07#
C3560CX-PRG07#sho running-config | include route
ip route 1.0.0.0 255.0.0.0 5.1.1.1
ip route 2.1.0.0 255.255.0.0 5.1.1.1
ip route 3.1.1.0 255.255.255.0 5.1.1.1
 remark netvmg-traceroute
C3560CX-PRG07#
```

Debugging prints

Generating static routes

```
### Initial settings
#set($Subnets_Subnet = [])
#set($Subnets_Mask = [])
#set($Subnets_List_NotEmpty = false)
### Convert Subnets and Masks input into arrays
#set($temp = $Subnets.replace(" ",""))
#set($Subnets = $temp)
###
#if (!$Subnets.isEmpty())
 #set($Subnets_List = $Subnets.split(";"))
 #set($n = $Subnets_List.size() - 1)
 #foreach($i in [0..$n])
  #set($pair = $Subnets_List[$i].split(","))
  #set($Subnets_List_NotEmpty = $Subnets_Subnet.add($pair[0]))
  #set($Subnets_List_NotEmpty = $Subnets_Mask.add($pair[1]))
### Debugging prints
  ! Run/index $i
  ! Subnets_List_Not_Empty = $Subnets_List_NotEmpty
  ! Subnet = $Subnets_Subnet
  ! Mask = $Subnets_Mask
  #end
#end
### Configure static routes
!
#if (!$Subnets.isEmpty())
 #set($n = $Subnets_Subnet.size() - 1)
 #foreach($i in [0..$n])
ip route $Subnets_Subnet[$i] $Subnets_Mask[$i] $next_hop
#end
#end
```

```
! Run/index 0
! Subnets_List_Not_Empty = true
! Subnet = [1.0.0.0]
! Mask = [255.0.0.0]
  ! Run/index 1
! Subnets_List_Not_Empty = true
! Subnet = [1.0.0.0, 2.1.0.0]
! Mask = [255.0.0.0, 255.255.0.0]
  ! Run/index 2
! Subnets_List_Not_Empty = true
! Subnet = [1.0.0.0, 2.1.0.0, 3.1.1.0]
! Mask = [255.0.0.0, 255.255.0.0, 255.255.255.0]

ip route 1.0.0.0 255.0.0.0 5.1.1.1
ip route 2.1.0.0 255.255.0.0 5.1.1.1
ip route 3.1.1.0 255.255.255.0 5.1.1.1
```

Debugging prints inserted in template code

Sorting strings into array

Problem origin

```
!
! Bound to source - single attribute (multiselect)
$interface_name
!
```

Multiselect variable in template

How values are presented when entering values

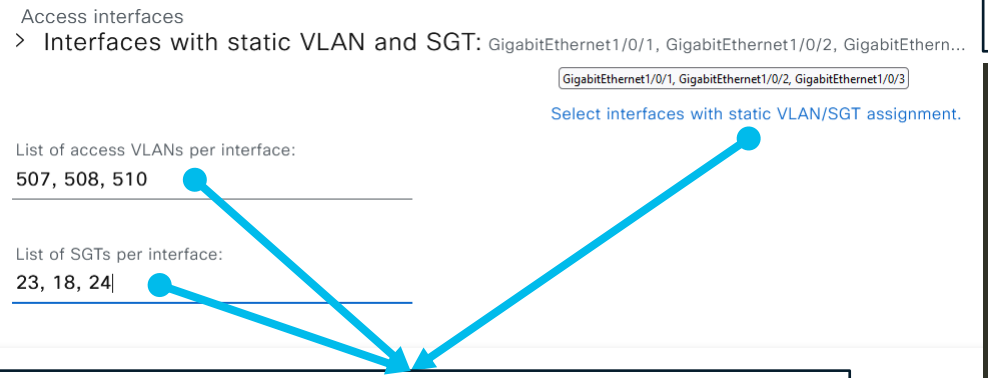
How values are presented to template when generating configuration

```
!
! Bound to source - single attribute (multiselect)
[GigabitEthernet1/0/10, GigabitEthernet1/0/12, GigabitEthernet1/0/13, GigabitEthernet1/0/11]
!
```

When configuration for all interfaces is the same there is no issue, but if not ....

Sorting strings into array

Problem definition and solution



Intent is to assign VLANs and SGTs to interfaces in the same order as they appear on screen (GigabitEthernet1/0/1 to VLAN 507 with SGT 23, GigabitEthernet1/0/2 to VLAN 508 with SGT 18, GigabitEthernet1/0/3 to VLAN 510 with SGT 24)

*) Must be tested and possibly modified for platforms used

Possible solution (or workaround) is to sort them into array*

```
###
### Sort Interface Names from shortest to longest with Bubblesort
###
#if($NoAuthInt_List.size() > 1)
 #set($m = $NoAuthInt_List.size() - 1)
 #set($n = $NoAuthInt_List.size() - 2)
 #foreach($i in [0..$n])
  #set($k=$i + 1)
  #foreach($j in [$k..$m])
   #if (($NoAuthInt_List[$i].length()) < ($NoAuthInt_List[$j].length()))
    #set($temp = ($NoAuthInt_List[$i]))
    #set($NoAuthInt_List[$i] = $NoAuthInt_List[$j])
    #set($NoAuthInt_List[$j] = $temp)
   #end
   #set($j=$j+1)
  #end
  #set($i=$i+1)
 #end
#end
###
### Sort Interface Names Lexicographically with Bubblesort
###
#if($NoAuthInt_List.size() > 1)
 #set($m = $NoAuthInt_List.size() - 1)
 #set($n = $NoAuthInt_List.size() - 2)
 #foreach($i in [0..$n])
  #set($k=$i + 1)
  #foreach($j in [$k..$m])
   #if (($NoAuthInt_List[$i].compareTo($NoAuthInt_List[$j]) > 0) ||
   (($NoAuthInt_List[$i].length()) > ($NoAuthInt_List[$j].length())))
    #set($temp = ($NoAuthInt_List[$i]))
    #set($NoAuthInt_List[$i] = $NoAuthInt_List[$j])
    #set($NoAuthInt_List[$j] = $temp)
   #end
   #set($j=$j+1)
  #end
  #set($i=$i+1)
 #end
#end
###
```

Dealing with existing configuration overlap

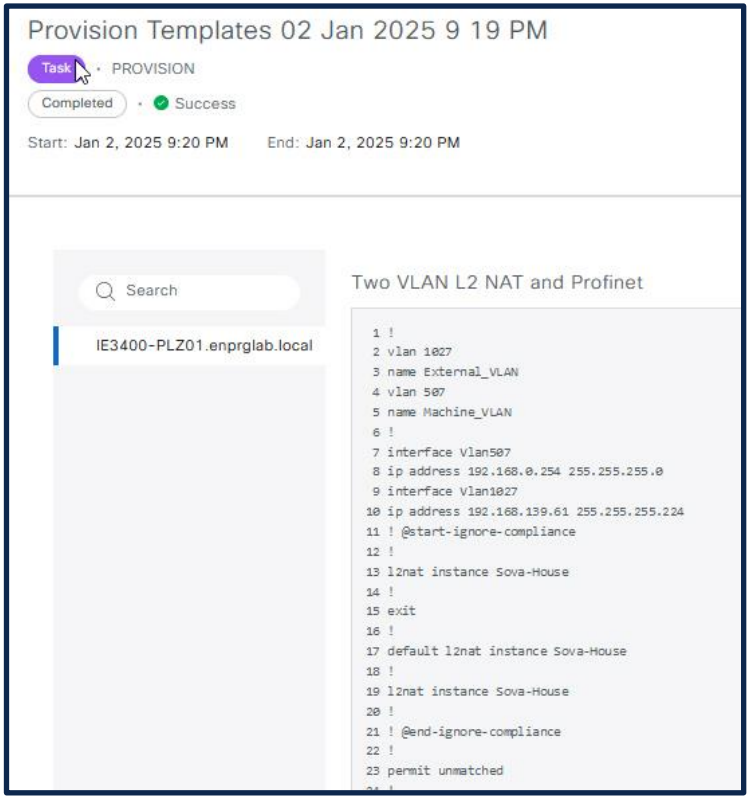
In some cases, pushing already existing commands leads to failure

```
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  ### Configure NAT entries
28  ###
```

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  ###
```



Removing part of existing configuration

- Creation of new version of template without given command is not enough
- Several possible options how to solve it:
  - Use the default (if possible) and recreate configuration from scratch
  - Use “no” command to remove desired line of configuration
  - Create section of 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
```

Using macros to optimize template code

Macro definition with variable

```
#macro(Closed_Auth_Int_cfg $interface)
! @start-ignore-compliance
  default interface $interface.portName
! @end-ignore-compliance
  interface $interface.portName
  description Access -lan- Closed authentication
  switchport mode access
  device-tracking attach-policy IPDT_POLICY
  dot1x timeout tx-period 7
  dot1x max-reauth-req 3
  source template DefaultWiredDot1xClosedAuth
  spanning-tree portfast
  spanning-tree bpduguard enable
  service-policy input DNA-MARKING_IN
  service-policy output DNA-dscp#APIC_QOS_Q_OUT
#end
```

Macro use in template

```
#foreach($interface in ${Closed_Authentication_Interfaces})
#if ($interface.portType == "Ethernet Port"  &&
    !$interface.description.contains("Uplink"))
#Closed_Auth_Int_cfg
#end
#end
```

Generated code for two interfaces

```
! @start-ignore-compliance
  default interface GigabitEthernet1/0/2
! @end-ignore-compliance
  interface GigabitEthernet1/0/2
  description Access -lan- Closed authentication
  switchport mode access
  device-tracking attach-policy IPDT_POLICY
  dot1x timeout tx-period 7
  dot1x max-reauth-req 3
  source template DefaultWiredDot1xClosedAuth
  spanning-tree portfast
  spanning-tree bpduguard enable
  service-policy input DNA-MARKING_IN
  service-policy output DNA-dscp#APIC_QOS_Q_OUT

! @start-ignore-compliance
  default interface GigabitEthernet1/0/1
! @end-ignore-compliance
  interface GigabitEthernet1/0/1
  description Access -lan- Closed authentication
  switchport mode access
  device-tracking attach-policy IPDT_POLICY
  dot1x timeout tx-period 7
  dot1x max-reauth-req 3
  source template DefaultWiredDot1xClosedAuth
  spanning-tree portfast
  spanning-tree bpduguard enable
  service-policy input DNA-MARKING_IN
  service-policy output DNA-dscp#APIC_QOS_Q_OUT
```

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

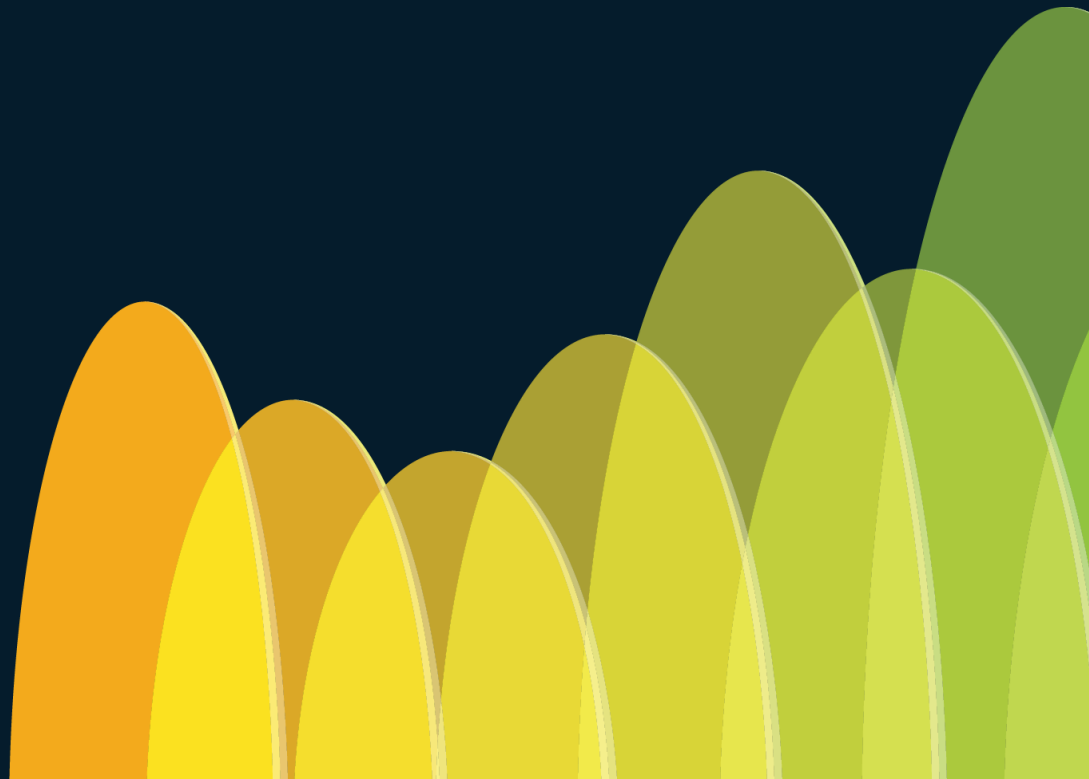
```
1  #MODE_ENABLE
2  write memory
3  #INTERACTIVE
4  reload in 1<IQ>confirm<R>y
5  #ENDS_INTERACTIVE
6  #MODE_END_ENABLE
```

Beginning of interactive command

End of interactive command

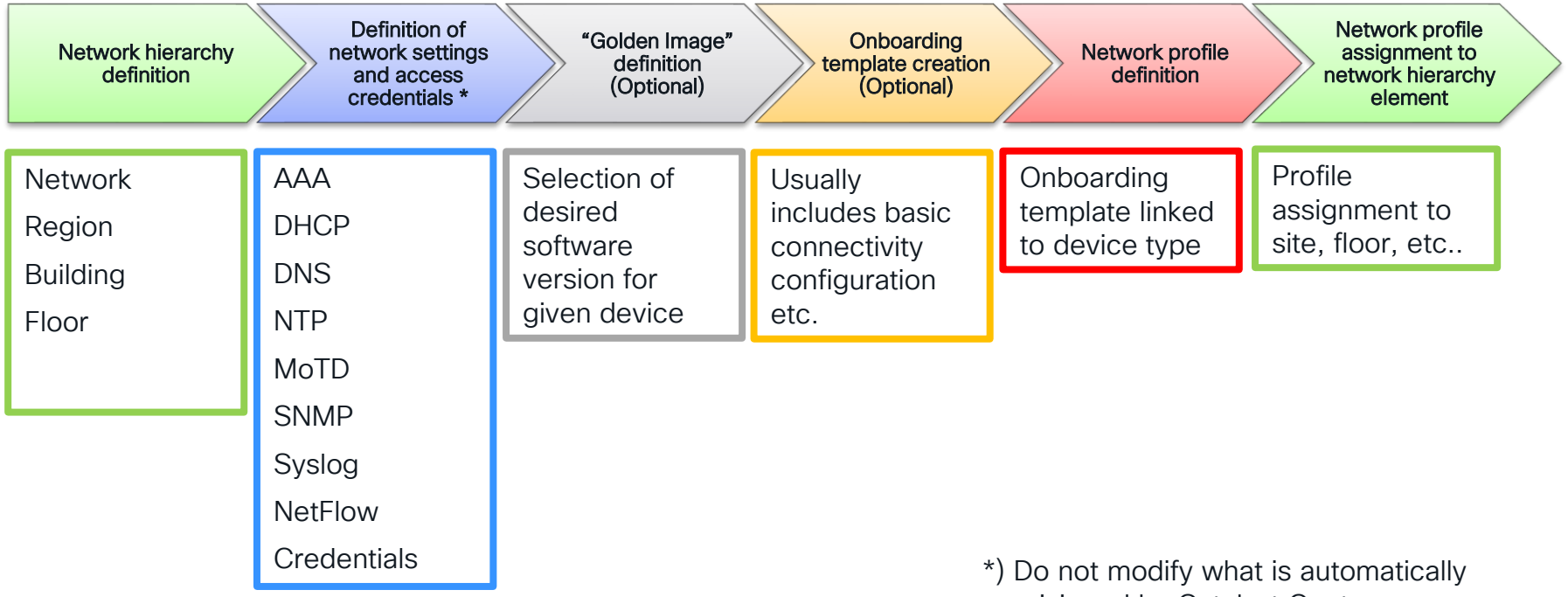
- An interactive command contains the input that must be entered following the execution of a command
- To enter an interactive command in the CLI content area, use the following syntax
  `CLI_command<IQ>interactive_question_1<R>response_1<IQ>interactive_question_2<R>response_2`
- <IQ> and <R> tags are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

Use of CLI templates – device onboarding



cisco Live!

Device onboarding- design workflow



*) Do not modify what is automatically provisioned by Catalyst Center

Device onboarding- provision workflow

Do not touch console until explicitly stated it is allowed

DHCP

DNS

Cloud redirection

List of devices detected by PnP server or manually added *

Option to change device name

Optional assignment of device to Site

Option to select software version

Option to select onboarding template

Option to configure stack

Enter or import onboarding template parameters

Configuration of parameters defined in network settings and by onboarding template

SUDI validation

Device is enrolled into inventory and assigned to the Site

*) If device, which can work as SDA PEN is connected to SDA FE node, shall not be configured as PEN, workaround shown in demo must be used otherwise SDA code will automatically claim it and make it PEN

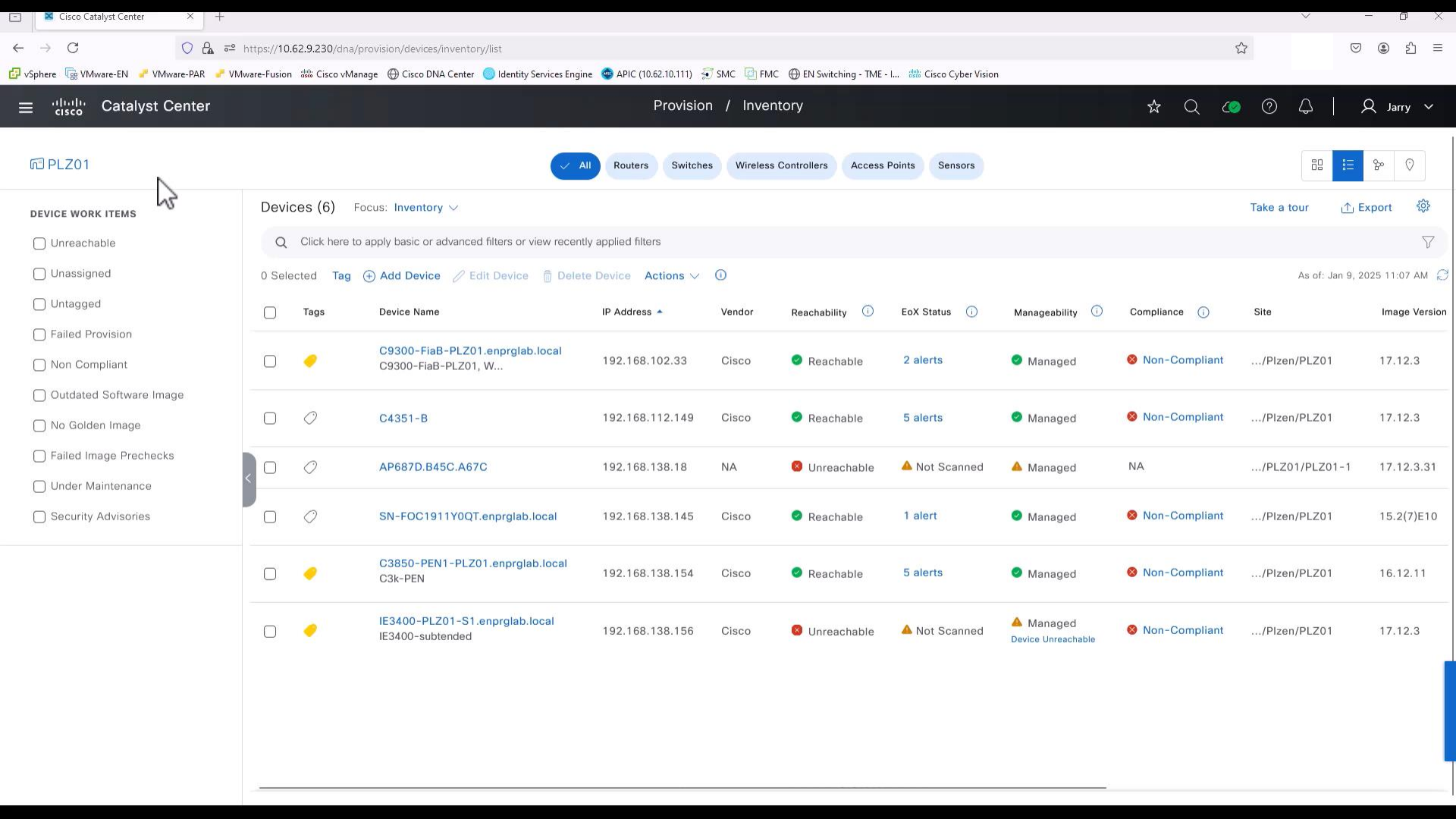
SCAN ME

Demo 1

Device onboarding with use of Day-0 template

Scan QR code or visit
https://youtu.be/HTNK1DeWhcg

cisco Live!


Cisco Catalyst Center
https://10.62.9.230/dna/provision/devices/inventory/list
vSphere
VMware-EN
VMware-PAR
VMware-Fusion
Cisco vManage
Cisco DNA Center
Identity Services Engine
APIC (10.62.10.111)
SMC
FMC
EN Switching - TME - I...
Cisco Cyber Vision
Catalyst Center
Provision / Inventory
Jarry
PLZ01
All
Routers
Switches
Wireless Controllers
Access Points
Sensors
DEVICE WORK ITEMS
Unreachable
Unassigned
Untagged
Failed Provision
Non Compliant
Outdated Software Image
No Golden Image
Failed Image Prechecks
Under Maintenance
Security Advisories
Devices (6)
Focus: Inventory
Take a tour
Export
Click here to apply basic or advanced filters or view recently applied filters
0 Selected
Tag
Add Device
Edit Device
Delete Device
Actions
As of: Jan 9, 2025 11:07 AM
Tags
Device Name
IP Address
Vendor
Reachability
EoX Status
Manageability
Compliance
Site
Image Version
C9300-FiaB-PLZ01.enprglab.local
C9300-FiaB-PLZ01, W...
192.168.102.33
Cisco
Reachable
2 alerts
Managed
Non-Compliant
.../Plzen/PLZ01
17.12.3
C4351-B
192.168.112.149
Cisco
Reachable
5 alerts
Managed
Non-Compliant
.../Plzen/PLZ01
17.12.3
AP687D.B45C.A67C
192.168.138.18
NA
Unreachable
Not Scanned
Managed
NA
.../PLZ01/PLZ01-1
17.12.3.31
SN-FOC1911Y0QT.enprglab.local
192.168.138.145
Cisco
Reachable
1 alert
Managed
Non-Compliant
.../Plzen/PLZ01
15.2(7)E10
C3850-PEN1-PLZ01.enprglab.local
C3k-PEN
192.168.138.154
Cisco
Reachable
5 alerts
Managed
Non-Compliant
.../Plzen/PLZ01
16.12.11
IE3400-PLZ01-S1.enprglab.local
IE3400-subtended
192.168.138.156
Cisco
Unreachable
Not Scanned
Managed
Device Unreachable
Non-Compliant
.../Plzen/PLZ01
17.12.3

Use of CLI templates – device configuration

cisco Live!

How to use CLI template for device configuration

- Option 1 – Provisioning device from 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 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 configuration generated by template is pushed
  - In Tasks it is shown as “Provision Templates” task (if not modified)

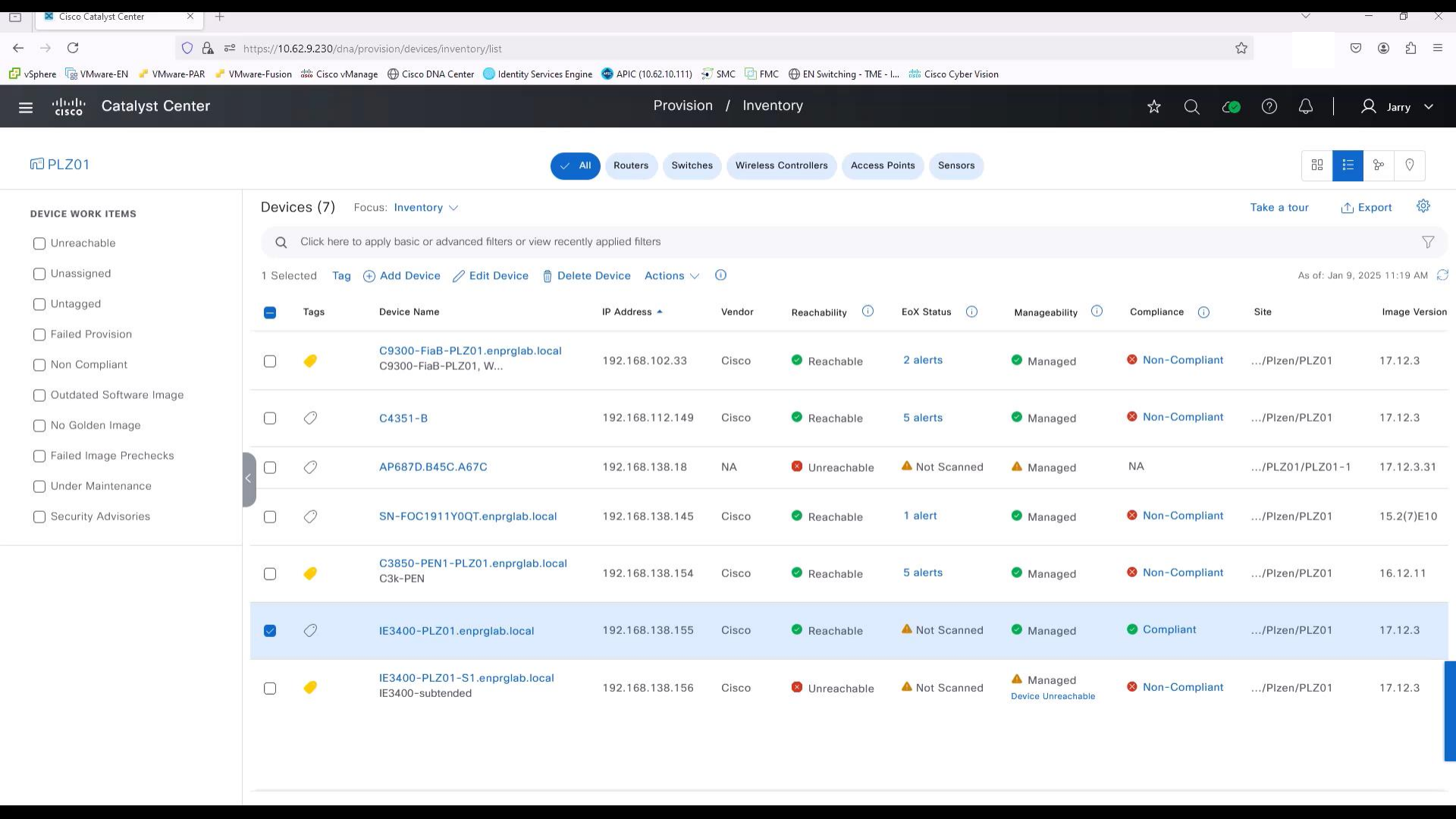
SCAN ME

Demo 2

Device provisioning with use of composite Day-N template

Scan QR code or visit
https://youtu.be/ZjTuAFHEy9U

cisco Live!


Cisco Catalyst Center
https://10.62.9.230/dna/provision/devices/inventory/list
vSphere
VMware-EN
VMware-PAR
VMware-Fusion
Cisco vManage
Cisco DNA Center
Identity Services Engine
APIC (10.62.10.111)
SMC
FMC
EN Switching - TME - I...
Cisco Cyber Vision
Catalyst Center
Provision / Inventory
Jarry
PLZ01
All
Routers
Switches
Wireless Controllers
Access Points
Sensors
DEVICE WORK ITEMS
Unreachable
Unassigned
Untagged
Failed Provision
Non Compliant
Outdated Software Image
No Golden Image
Failed Image Prechecks
Under Maintenance
Security Advisories
Devices (7)
Focus: Inventory
Take a tour
Export
Click here to apply basic or advanced filters or view recently applied filters
1 Selected
Tag
Add Device
Edit Device
Delete Device
Actions
As of: Jan 9, 2025 11:19 AM
Tags
Device Name
IP Address
Vendor
Reachability
EoX Status
Manageability
Compliance
Site
Image Version
C9300-FiaB-PLZ01.enprglab.local
C9300-FiaB-PLZ01, W...
192.168.102.33
Cisco
Reachable
2 alerts
Managed
Non-Compliant
.../Plzen/PLZ01
17.12.3
C4351-B
192.168.112.149
Cisco
Reachable
5 alerts
Managed
Non-Compliant
.../Plzen/PLZ01
17.12.3
AP687D.B45C.A67C
192.168.138.18
NA
Unreachable
Not Scanned
Managed
NA
.../PLZ01/PLZ01-1
17.12.3.31
SN-FOC1911Y0QT.enprglab.local
192.168.138.145
Cisco
Reachable
1 alert
Managed
Non-Compliant
.../Plzen/PLZ01
15.2(7)E10
C3850-PEN1-PLZ01.enprglab.local
C3k-PEN
192.168.138.154
Cisco
Reachable
5 alerts
Managed
Non-Compliant
.../Plzen/PLZ01
16.12.11
IE3400-PLZ01.enprglab.local
192.168.138.155
Cisco
Reachable
Not Scanned
Managed
Compliant
.../Plzen/PLZ01
17.12.3
IE3400-PLZ01-S1.enprglab.local
IE3400-subtended
192.168.138.156
Cisco
Unreachable
Not Scanned
Managed
Device Unreachable
Non-Compliant
.../Plzen/PLZ01
17.12.3

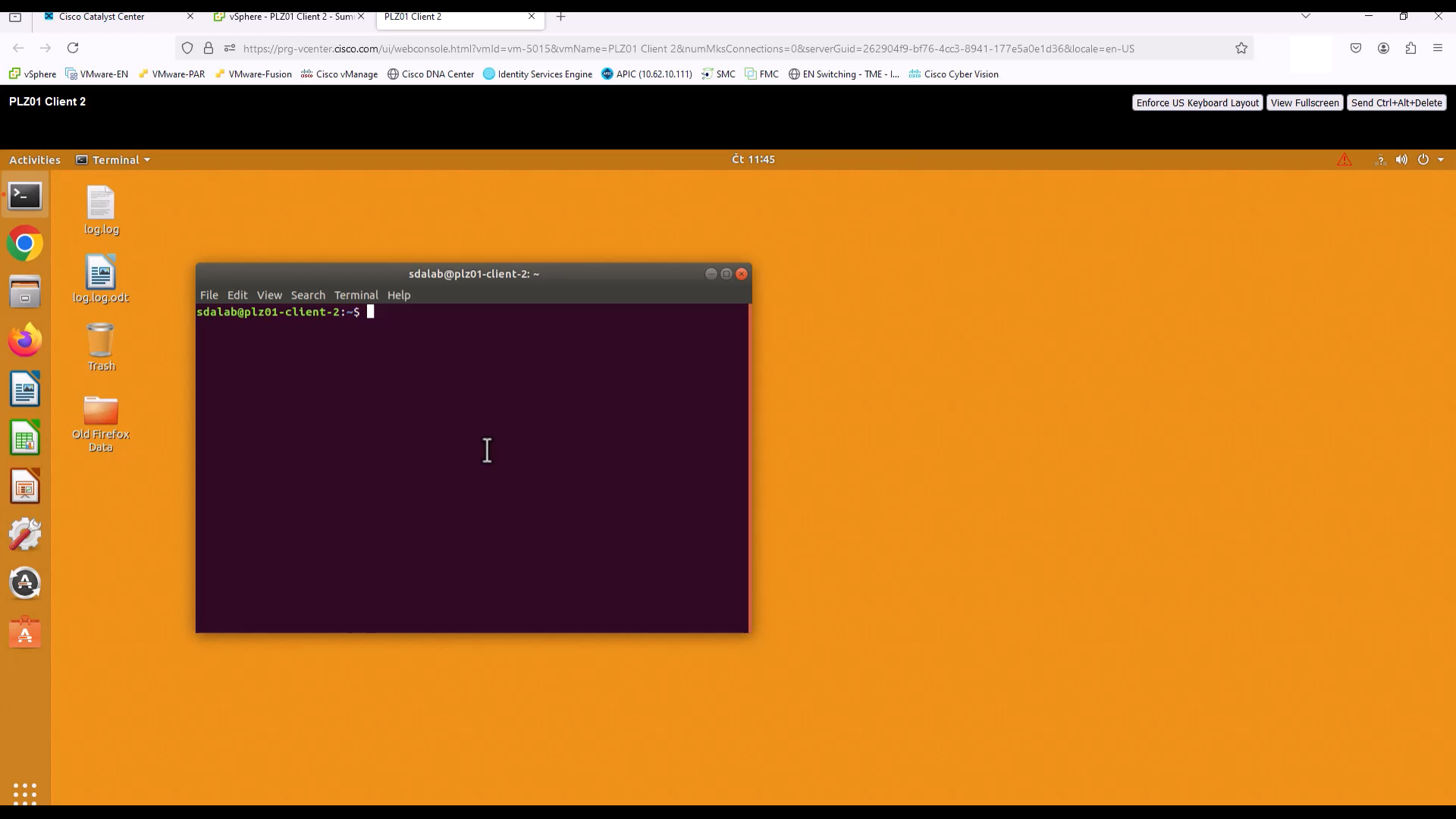
SCAN ME

Demo 3

Device configuration change with use of single Day-N template

Scan QR code or visit
https://youtu.be/hJEqrhwevlo

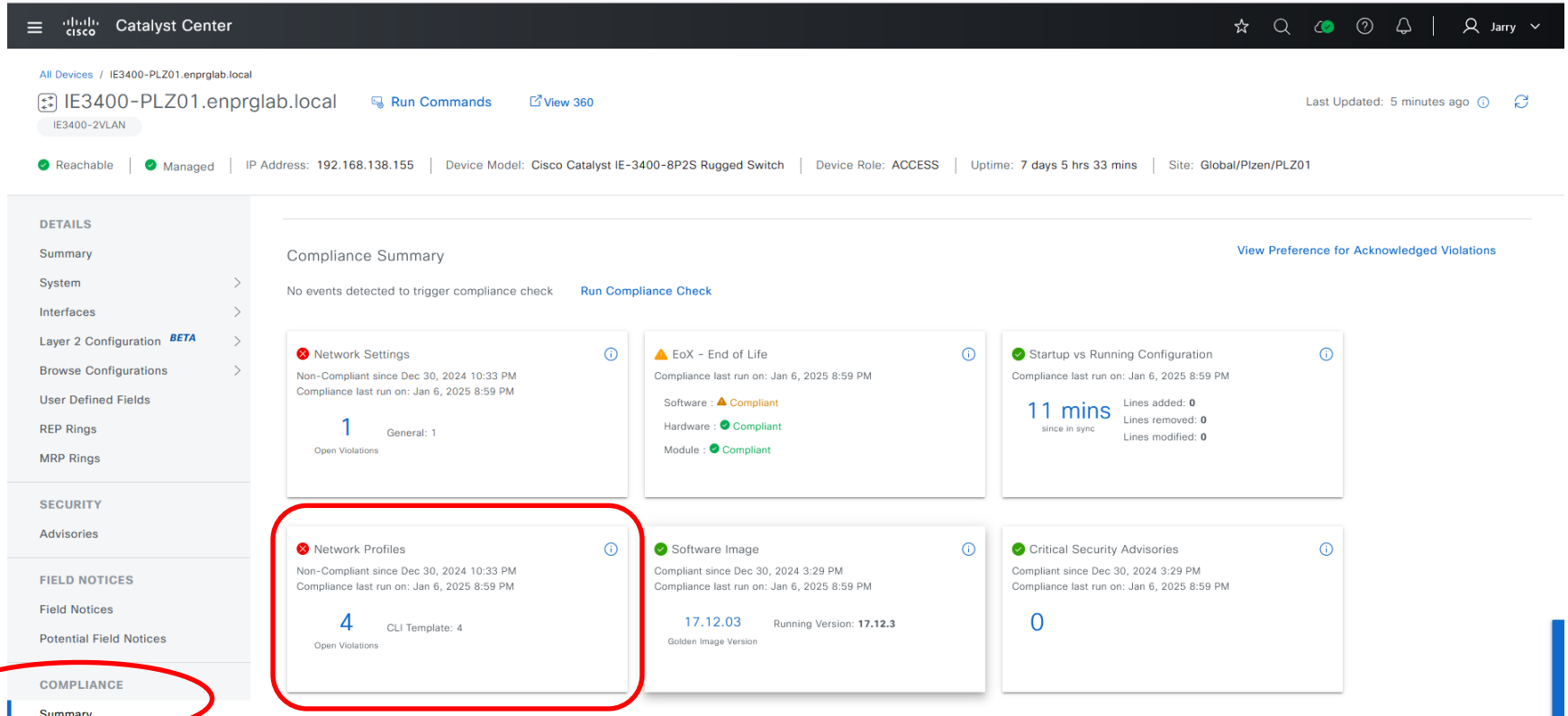
cisco Live!


Cisco Catalyst Center
vSphere - PLZ01 Client 2 - Sum
PLZ01 Client 2
https://prg-vcenter.cisco.com/ui/webconsole.html?vmId=vm-5015&vmName=PLZ01 Client 2&numMksConnections=0&serverGuid=262904f9-bf76-4cc3-8941-177e5a0e1d36&locale=en-US
vSphere
VMware-EN
VMware-PAR
VMware-Fusion
Cisco vManage
Cisco DNA Center
Identity Services Engine
APIC (10.62.10.111)
SMC
FMC
EN Switching - TME - I...
Cisco Cyber Vision
PLZ01 Client 2
Enforce US Keyboard Layout
View Fullscreen
Send Ctrl+Alt+Delete
Activities
Terminal
Čt 11:45
log.log
log.log.odt
Trash
Old Firefox Data
sdalab@plz01-client-2: ~
File Edit View Search Terminal Help
sdalab@plz01-client-2:~$

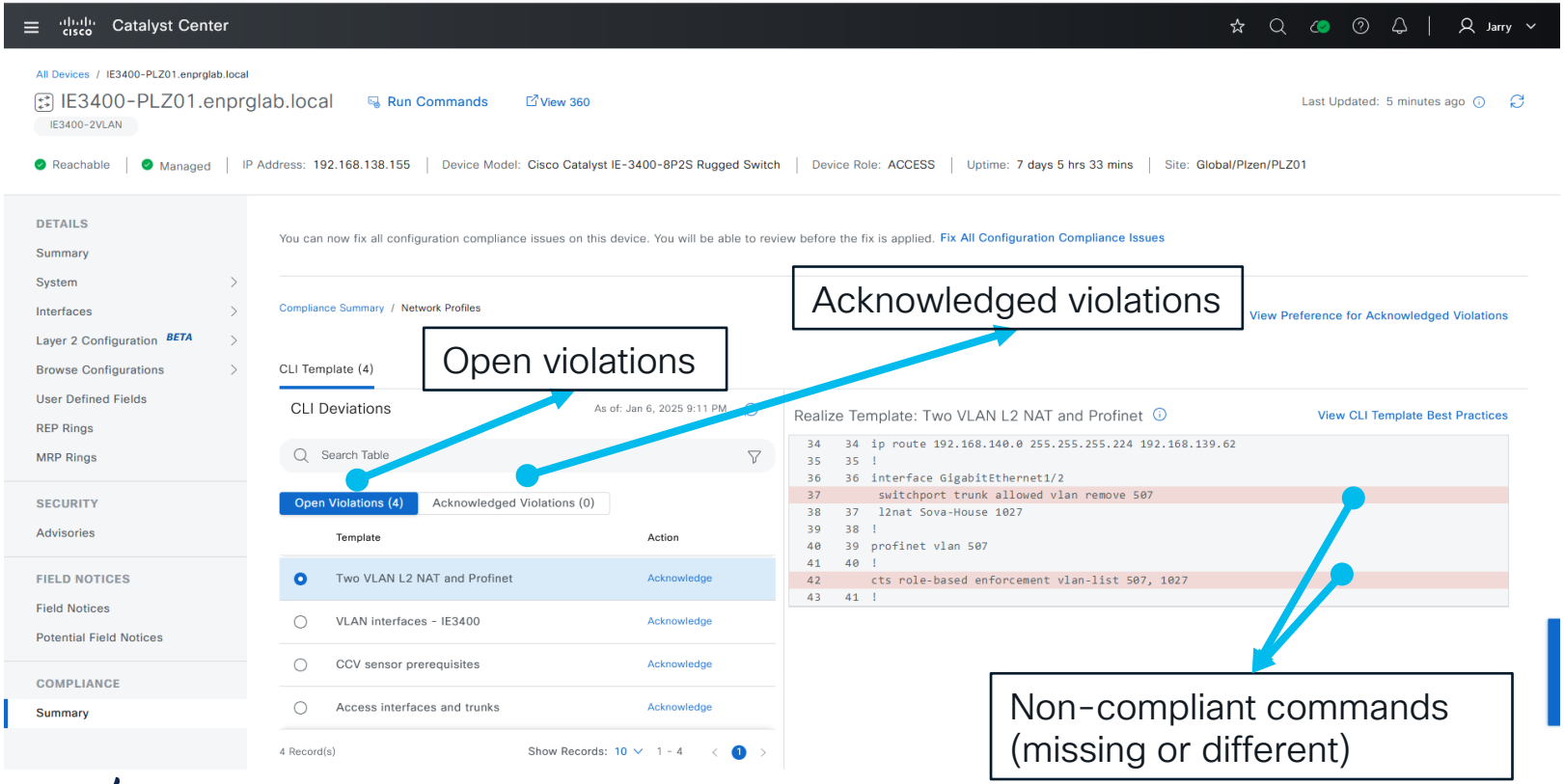
Device configuration compliance with templates

cisco Live!

Compliance – Network Profiles



Compliance - Violations



Avoiding compliance check

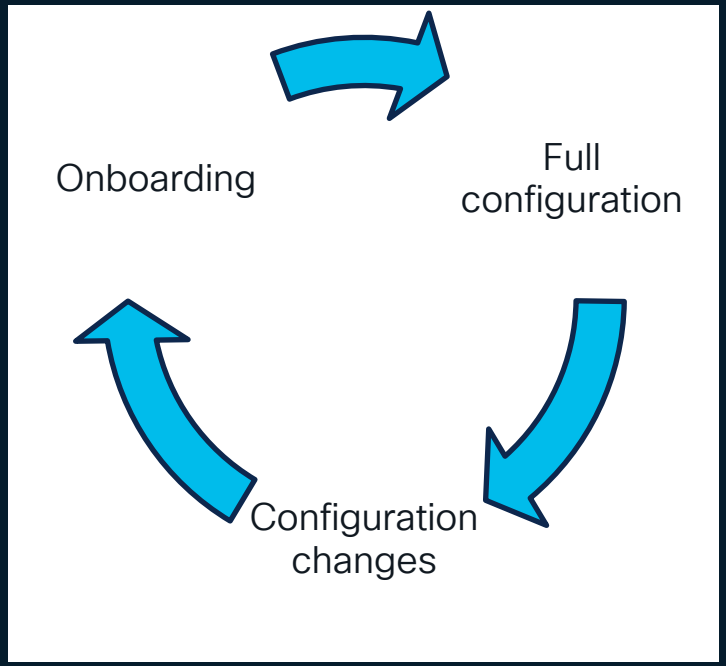
```
!
#if ($interface.portType == "Ethernet Port"  &&
    !$interface.description.contains("Uplink")
)
!
! @start-ignore-compliance
  default interface $interface.portName
! @end-ignore-compliance
  interface $interface.portName
  description Access -lan- Closed authentication
  switchport mode access
  device-tracking attach-policy IPDT_POLICY
  dot1x timeout tx-period 7
  dot1x max-reauth-req 3
  source template DefaultWiredDot1xClosedAuth
  spanning-tree portfast
  spanning-tree bpduguard enable
!
#end
!
```

Beginning of section where compliance is not checked

End of section where compliance is not checked

Summary and References

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



Reference information

Apache Velocity Language

- Apache Velocity Project
  - ✓ https://velocity.apache.org/
- AVL User Guide
  - ✓ https://velocity.apache.org/engine/2.4/user-guide.html
- String manipulation
  - ✓ https://docs.oracle.com/javase/7/docs/api/java/lang/String.html or https://docs.oracle.com/javase/8/docs/api/java/lang/String.html
- Example: comparing of strings
  - ✓ https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#compareTo(java.lang.String) or https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#compareTo-java.lang.String-

Reference information

Jinja

- Cisco Catalyst Center Jinja Templating – Part 1
  - ✓ https://ciscolearning.github.io/cisco-learning-codelabs/posts/cat-center-j2-part-1/#0
- Cisco Catalyst Center Jinja Templating – Part 2
  - ✓ https://ciscolearning.github.io/cisco-learning -codelabs/posts/cat-center-j2-part-2/#0

Reference information

Additional links

- Adam Radford's blog
  - https://blogs.cisco.com/developer/velocity-templates-dnac-1
  - https://blogs.cisco.com/developer/velocity-templates-dnac-2
  - https://blogs.cisco.com/developer/velocity-templates-in-dna-center-3
  - https://blogs.cisco.com/developer/velocity-templates-in-dnac-4
- Devnet/Github collections
  - https://developer.cisco.com/codeexchange/github/repo/kebaldwi/DNAC-TEMPLATES/
  - https://github.com/ramkchel/Cisco-DNA-Center-Templates
  - https://developer.cisco.com/docs/dna-center/#!device-provisioning/configuration-template-api
- YouTube videos (search for DNAC or Catalyst Center templates)
  - https://www.youtube.com/watch?v=_m-6rhLHJ1I
  - https://www.youtube.com/watch?v=JrauG5iGkeM
  - https://www.youtube.com/watch?v=79f4Sjo6UNQ&t=802s

Webex App

Questions?

Use the Webex app to chat with the speaker after the session

How

1. Find this session in the Cisco Events 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 February 28, 2025.

Fill Out Your Session Surveys

Participants who fill out a minimum of 4 session surveys and the overall event survey will get a unique Cisco Live t-shirt.

(from 11:30 on Thursday, while supplies last)

All surveys can be taken in the Cisco Events mobile app or by logging in to the Session Catalog and clicking the ‘Participant Dashboard’

Content Catalog

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 ciscolive.com/on-demand. Sessions from this event will be available from March 3.

Contact me at: Webex or jpilar@cisco.com

You can start to practice now !

Walk-in Lab:

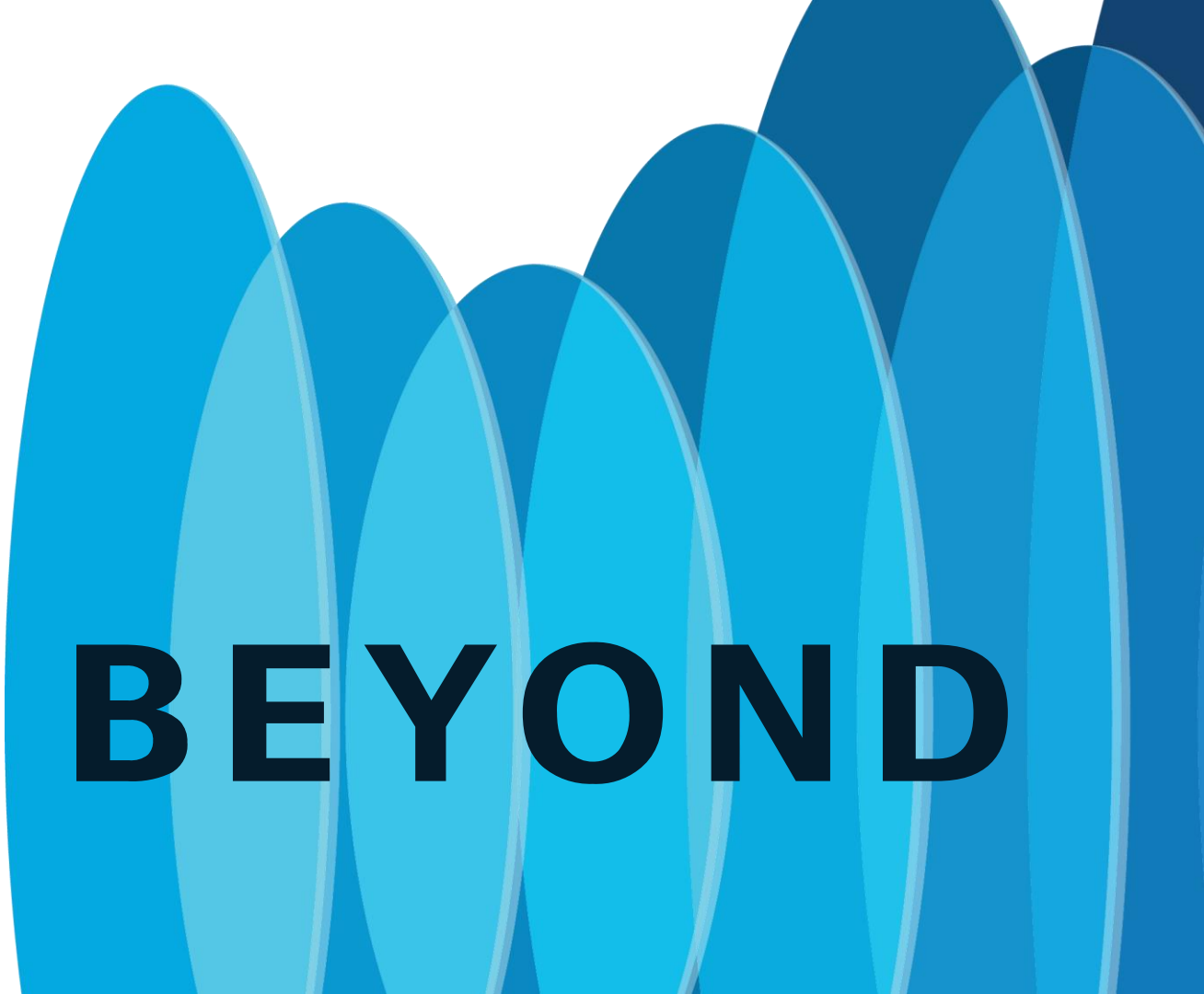
LABOPS-1470: Click Once, Configure Everything with Cisco Catalyst Center using Configuration Templates

Osvaldo Jose De la Hoz Hadechiny, Technical Consulting Engineer
Carlos Moreno Almazan, Technical Leader

Thank you

cisco Live!

cisco Live!



GO BEYOND