

CISCO *Live!*



#CiscoLive



The bridge to possible

# Prepare for the Enterprise Automation (ENAUTO) Certification with Real-Life Applications

George Koukis, Lead Exam Program Manager, CCIEW#42079  
Jason Belk, Senior Technology Advocate  
BRKCRT-2014

# 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 17, 2022.



<https://cislive.ciscoevents.com/cislivebot/# BRKCRT-2014>



# Agenda

- Introduction to Cisco Certifications
- What to expect on the exam
- Exam vs Real life
- How to prepare for the exam

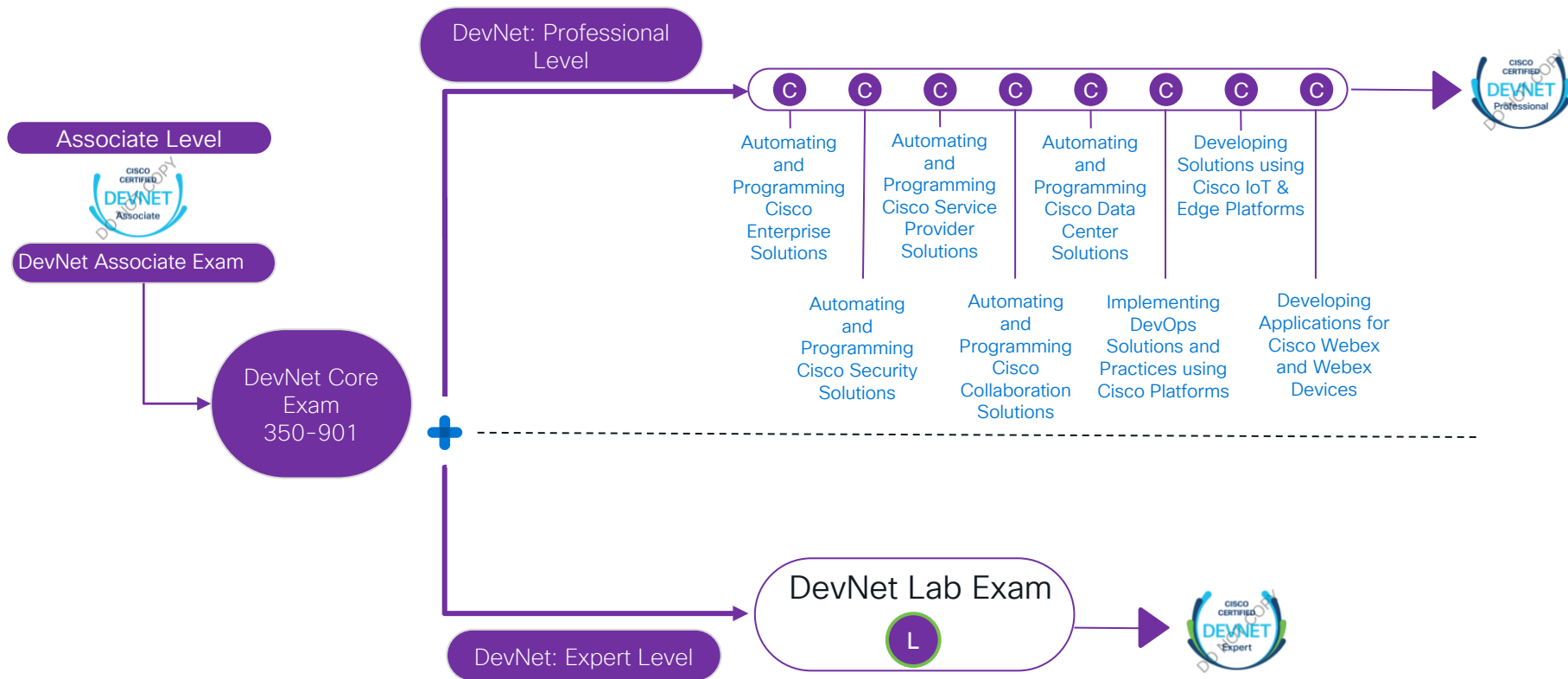
# Introduction to Cisco Certifications



# “DevNet professional, ENAUTO and Automation skills Required”



# Cisco DevNet-Automation certification track



# What to expect on the exam



# Exam Blueprint

<https://learningnetwork.cisco.com>

## 300-435 ENAUTO: Automating and Programming Cisco Enterprise Solutions

### Exam Description

The Automating and Programming Cisco Enterprise Solutions v1.0 (ENAUTO 300-435) exam is a 90-minute exam associated with the CCNP Enterprise, Cisco Certified DevNet Professional, and Cisco Certified DevNet Specialist – Enterprise Automation and Programmability certifications. This exam tests a candidate's knowledge of implementing Enterprise automated solutions, including programming concepts, Python programming, APIs, controllers and automation tools. The course, Implementing Cisco Enterprise Automation Solutions, helps candidates to prepare for this exam.

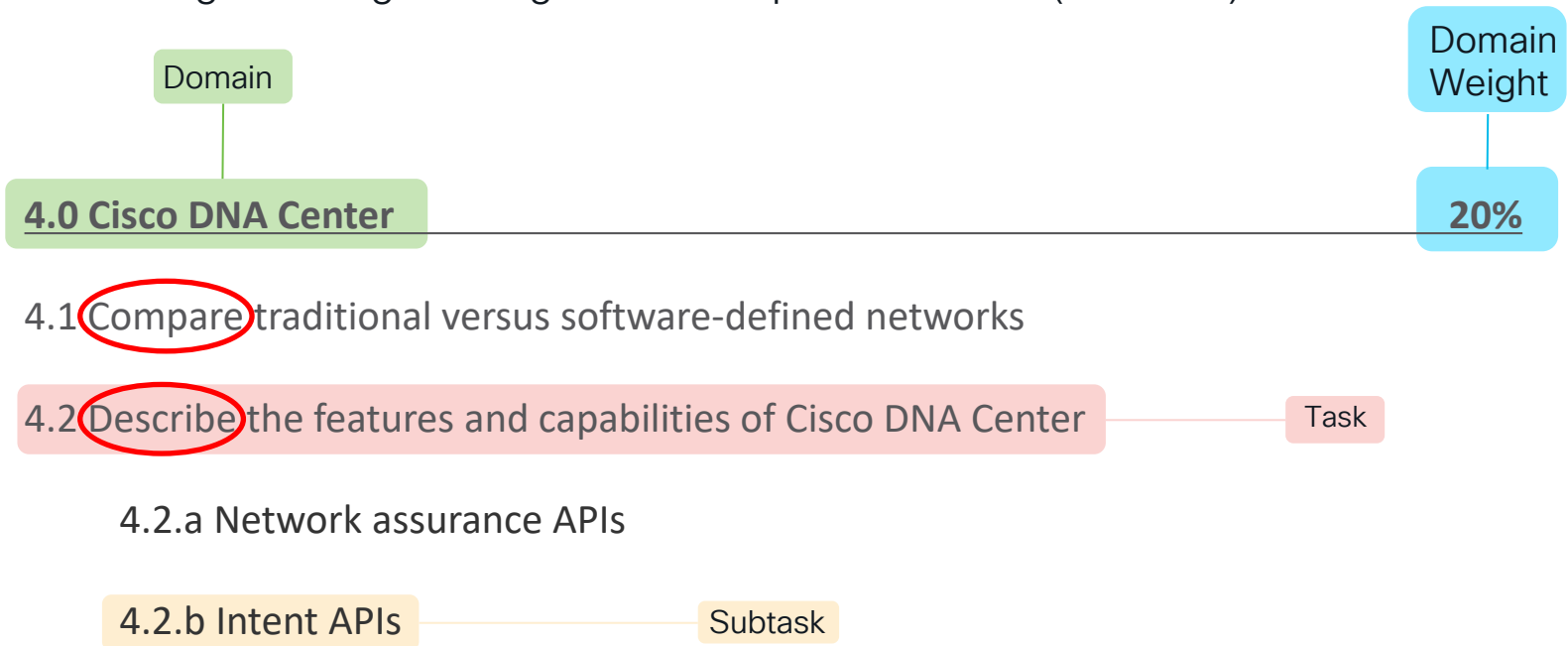
The following topics are general guidelines for the content likely to be included on the exam. However, other related topics may also appear on any specific delivery of the exam. To better reflect the contents of the exam and for clarity purposes, the guidelines below may change at any time without notice.

#### Download Complete List of Topics in PDF format

1.0 Network Programmability Foundation	10%	▾
2.0 Automate APIs and Protocols	10%	▾
3.0 Network Device Programmability	20%	▾
4.0 Cisco DNA Center	20%	▾
5.0 Cisco SD-WAN	20%	▾
6.0 Cisco Meraki	20%	▾

# Interpret the Blueprint:

Automating and Programming Cisco Enterprise Solutions (300-435)



# Blueprint Verbs

Describe/Explain

Compare

Configure/Implement/Construct/Utilize/Interpret

Troubleshoot/Identify

Depth of Knowledge

The diagram consists of four horizontal bars of increasing length and depth of knowledge, stacked vertically. The bars are colored in a gradient from dark blue to light blue. The text on the bars is white. Below the bars is a large yellow arrow pointing to the right, with the text 'Depth of Knowledge' centered inside it.

# Types of questions



Multiple choice



Drag and drop



Fill in the blanks

# Describe question

**Example:** Describe the functionality of these IP Services: DHCP, DNS, NAT, SNMP, NTP

Which protocol synchronizes the clock between computer systems?

- A. **NTP**
- B. NAT
- C. DNS
- D. SNMP

# Compare question

Example: Compare different API styles (REST and RPC)

What is the difference between REST and RPC APIs?

\*A. REST APIs are stateless, and RPC APIs are stateful.

B. REST APIs are vendor-specific, and RPC APIs are vendor-neutral.

C. REST APIs are...

D. REST APIs are...

**OR like this**

Drag and drop the characteristics from the left to the API style on the right.	
Options	Categories
stateful	<b>Rest APIs</b>
stateless	stateless
uses XML	uses XML
uses YAML	<b>RPC APIs</b>
	stateful
	uses YAML

# Construct question

Usually a code snippet with missing parts.

Example: Construct a Python unit test

Drag and the drop the code on the snippet to complete the Python unit test. Not all options are used.

```
import   
import logging  
  
log = logging.getLogger(__name__)  
log.setLevel(logging.DEBUG)  
  
 uniFest(unittest.TestCase):  
    def setUp(self):  
        log.info('I am doing the setUp')  
  
    def test_one(self):  
        log.info('one')  
  
    @classmethod  
    def tearDownClass(self):  
        log.info('tearDown class')  
  
 test_two(self):  
    log.info('two')  
  
    @classmethod  
    def setUpClass(self):  
        log.info('set up Class')
```

class

unittest

log

def

class

# Troubleshoot question

Something is broken and we need to fix it.

**Example:** Troubleshoot scripts based on HTTP codes

```
void
Sample_invoke_hello_world(void *link, json_t *rid, json_t *params) {
    if (!link || !rid) {
        mlib_log_err("NULL input arguments");
        mlib_rpc_send_invoke_response(link, rid,
            json_string("NULL input arguments"));
        return;
    }

    json_t *hello = json_object();
    json_object_set(hello, "Greeting",
        json_string("Hello World from IOx Micro Services"));

    mlib_rpc_send_invoke_response(link, rid, hello);
    json_decref(hello);
}
```

Refer to the exhibit. A network engineer has developed a script to automate the provision of newly added devices on the network. In the lab environment, where the script was tested, it was running with no issues. When it was applied to the customer side, it came back with a 404 error. What needs to be changed on the script to fix the error?

- \*A. The IP address on the script must be set to the customer's range.
- B. The device that needs to be provisioned must connect to the network.
- C. The script needs to be written in Yang instead of JSON for the devices to listen.
- D. The server running the script in a different range than the device to provision.



# Exam vs Real life



# What will be covered

2.1 Identify the JSON instance based on a YANG model

2.2 Identify the XML instance based on a YANG model

3.3 Configure device using RESTCONF API utilizing Python requests library

3.4 Utilize Ansible to configure an IOS XE device

# YANG & JSON in Cisco NSO

```
1  module router-model {
2      namespace "http://com/example/routermodel";
3      prefix router-model;
4      import ietf-inet-types {
5          prefix inet;
6      }
7      container router {
8          leaf name { type string;}
9          leaf address { type inet:ipv4-address;}
10         leaf operational-status {
11             type enumeration {
12                 enum up;
13                 enum down;}
14         }}}}
```

- YANG describes the constraints of an application's data, defining the structure of an application's data payload
- JSON is an encoding format to structure data to more easily parse data payloads
- YANG defines the structure, JSON is the structure, the application gives the values

# Let's Test It: YANG & JSON in Cisco NSO

```
admin@ncs# conf
Entering configuration mode terminal
admin@ncs(config)# router ?
Possible completions:
  address name operational-status
admin@ncs(config)# router name VEGAS-GW1 address 10.20.30.40 operational-status ?
Possible completions:
  [up] down up
admin@ncs(config)# router name VEGAS-GW1 address 10.20.30.40 operational-status down
admin@ncs(config)# end
```

Application

Input

Data Payload

- The YANG model lets us know what the application structure will be
- The YANG model lets us know what the payload will be, so if we need to get a value from a key, we can easily

```
admin@ncs# show running-config router
router name      VEGAS-GW1
router address   10.20.30.40
router operational-status down
admin@ncs# show running-config router | display json
{
  "data": {
    "router-model:router": {
      "name": "VEGAS-GW1",
      "address": "10.20.30.40",
      "operational-status": "down"
    }
  }
}
admin@ncs#
```

## 2.1 Identify the JSON instance based on a YANG model

```
module router-model {
  namespace "http://com/example/routermodel";
  prefix router-model;
  import ietf-inet-types {
    prefix inet;
  }
  container router {
    leaf name { type string;}
    leaf address { type inet:ipv4-address;}
    leaf operational-status {
      type enumeration {
        enum up;
        enum down;}
    }
  }
}
```

Refer to the exhibit. An engineer sees this YANG model in the docs for a new IOS-XE router. The engineer uses RESTCONF to get a JSON response to grab the router status and IP address. Which JSON instance represents the YANG model?

\*A.

```
{
  "data": {
    "router-model:router": {
      "name": "SJC-GW1",
      "address": "10.10.20.40",
      "operational-status": "up"
    }
  }
}
```

B.

```
{
  "data": {
    "router-model:router": {
      "name": "SJC-GW1",
      "address": "10.10.20.40",
      "operational-status": "good"
    }
  }
}
```

C.

```
{
  "data": {
    "router-model:router": {
      "name": "SJC-GW1",
      "address": "10.10.20.40/24",
      "operational-status": "alive"
    }
  }
}
```

D.

```
{
  "data": {
    "router-model:router": {
      "description": "SJC-GW1",
      "address": "10.10.20.40",
      "status": "operational"
    }
  }
}
```

# YANG & XML in Cisco NSO

```
belk-model.yang ×
rc > yang > belk-model.yang
1  module belk-model {
2      namespace "http://com/example/belkmodel";
3      prefix belk-model;
4      container person{
5          leaf name{ type string; }
6          leaf age { type uint32; }
7          leaf favorite-color { type string; }
8      }
9  }
```

- YANG describes the constraints of an application's data, defining the structure of an application's data payload
- XML is an encoding format to structure data to more easily parse data payloads
- YANG defines the structure, XML is the structure, the application gives the values

# Let's Test It: YANG & XML in Cisco NSO

```
admin@ncs# conf
Entering configuration mode terminal
admin@ncs(config)# person ?
Possible completions:
  age favorite-color name
admin@ncs(config)# person name ?
Possible completions:
  <string>
admin@ncs(config)# person name Jason age ?
Possible completions:
  <unsignedInt>
admin@ncs(config)# person name Jason age 33 favorite-color ?
Possible completions:
  <string>
admin@ncs(config)# person name Jason age 33 favorite-color green
admin@ncs(config)#
```

- The YANG model lets us know what the application structure will be
- The YANG model lets us know what the payload will be, so if we need to get a value from a key, we can easily

Application Input

Data Payload

```
admin@ncs# show running-config person
person name      Jason
person age       33
person favorite-color green
admin@ncs#
admin@ncs#
admin@ncs# show running-config person | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <person xmlns="http://com/example/belkmodel">
    <name>Jason</name>
    <age>33</age>
    <favorite-color>green</favorite-color>
  </person>
</config>
admin@ncs#
```

## 2.2 Identify the XML instance based on a YANG model

```
module example-model {  
  namespace "http://com/example/examplemodel";  
  prefix example-model;  
  container person {  
    leaf name { type string;}  
    leaf age {type uint32;}  
    leaf favorite-color { type string;}  
  }  
}
```

Refer to the exhibit. A new corporate inventory application uses this YANG data model to describe some basic attributes of a person object. An engineer needs to access the REST API structured by the model and identify the payload. Which XML instance represents the YANG model?

- \*A. 

```
<person xmlns="http://com/example/examplemodel">  
  <name>User1</name>  
  <age>100</age>  
  <favorite-color>green</favorite-color>  
</person>
```
- B. 

```
<person xmlns="http://com/example/examplemodel">  
  <name>User1<name>  
  <age>100<age>  
  <favorite-color>green<favorite-color>  
<person>
```
- C. 

```
<person xmlns="http://com/examplemodel">  
  <name>User1</name>  
  <age>100</age>  
  <favorite-color>green</favorite-color>  
</person>
```
- D. 

```
<person xmlns="https://com/example/examplemodel">  
  <name>User1</name>  
  <age>100</age>  
  <favorite-color>green</favorite-color>  
</person>
```



# Demo Overview: RESTCONF & Python Requests

```
1 import requests, json
2 from urllib3.exceptions import InsecureRequestWarning
3 requests.packages.urllib3.disable_warnings(category=InsecureRequestWarning)
4
5 URL = "https://10.10.20.48:443"
6 USER = 'developer'
7 PASS = 'C1sco12345'
8
9 url = URL + "/restconf/data/Cisco-IOS-XE-native:native/interface/GigabitEthernet=2"
10 headers = {'content-type': 'application/yang-data+json',
11           'accept': 'application/yang-data+json'}
12 result = requests.get(url, auth=(USER, PASS), headers=headers, verify=False)
13 print(result.text)
14
15 payload = json.dumps({
16     "Cisco-IOS-XE-native:GigabitEthernet": {
17         "name": "2",
18         "description": "Description updated by RESTCONF"
19     }
20 })
21 url = URL + "/restconf/data/Cisco-IOS-XE-native:native/interface/GigabitEthernet"
22 result = requests.patch(url, auth=(USER, PASS),
23                         headers=headers, verify=False, data=payload)
24 print(result.text)
25
26 url = URL + "/restconf/data/Cisco-IOS-XE-native:native/interface/GigabitEthernet=2"
27 result = requests.get(url, auth=(USER, PASS), headers=headers, verify=False)
28 print(result.text)
```

- RESTCONF uses HTTPS to interact with an IOS-XE network device API
- Requests is a Python library to Create/Read/Update/Delete config using RESTCONF
- This script gets the current state, prints it, then configures an interface description and gets the new interface state and prints it to verify the change

# Let's Test It: RESTCONF & Python Requests

```
(brkcrt) restconf-requests$ python interface-restconf-configure-full.py
{
  "Cisco-IOS-XE-native:GigabitEthernet": {
    "name": "2",
    "description": "Network Interface",
    "shutdown": [null],
    "mop": {
      "enabled": false,
      "sysid": false
    },
    "Cisco-IOS-XE-ethernet:negotiation": {
      "auto": true
    }
  }
}
```

RESTCONF state of interface description pre-change

```
{
  "Cisco-IOS-XE-native:GigabitEthernet": {
    "name": "2",
    "description": "Description updated by RESTCONF",
    "shutdown": [null],
    "mop": {
      "enabled": false,
      "sysid": false
    },
    "Cisco-IOS-XE-ethernet:negotiation": {
      "auto": true
    }
  }
}

(brkcrt) restconf-requests$
```

RESTCONF state of interface description post-change

### 3.3 Configure device using RESTCONF API utilizing Python requests library

Drag and drop the code onto the blanks where the code is missing to use the request package to configure the description on GigabitEthernet 2 using RESTCONF. Not all options are used.

```
import requests, json
base_path = "[ ]://10.10.20.48:443"
url = base_path + "/restconf/data/Cisco-IOS-XE-native:native/
interface/GigabitEthernet"
headers = {'[ ]': 'application/yang-data+json',
'accept': 'application/yang-data+json'}
payload = json.dumps({"Cisco-IOS-XE-native:GigabitEthernet":
{"name": "2","description": "Description updated by RESTCONF"}})
result = requests.[ ](url,
auth=("developer", "Cisco12345"),
headers=headers, verify=False, [ ]=payload)
```

http

https

content-type

patch

data

get

status

# Demo Overview: Using Ansible to Configure IOS-XE

```
ansible > ≡ inventory.ini
1  [iosxe]
2  csr1 ansible_host=10.10.20.48 ansible_network_os=ios
3
4  [all:vars]
5  ansible_user=developer
6  ansible_ssh_pass=Cisco12345
7  ansible_connection=network_cli
```

```
ansible > ! pb-configure-snmp.yaml > ...
1  ---
2  - name: PLAY 1 - DEPLOYING SNMP CONFIGURATIONS ON IOS
3    hosts: "csr1"
4    connection: network_cli
5    gather_facts: no
6    tasks:
7      - name: "TASK 1 in PLAY 1 - CONFIGURE SNMP LINES"
8        cisco.ios.ios_config:
9          lines:
10            - snmp-server community belk-demo R0
11              snmp-server location VEGAS
12              snmp-server contact JASON_BELK
13      - name: "TASK 2 in PLAY 1 - VERIFY SNMP LINES PRESENT"
14        cisco.ios.ios_command:
15          commands:
16            - "show run | include snmp-server"
```

- The Ansible inventory file defines the device credentials, IP address and device type behind the scenes
- The Ansible playbook lists a series of tasks to be executed upon the device “csr1”
- The playbook has two tasks:
  - First, it sends three SNMP configuration lines
  - Second, it sends a verification show command to see if the change was successful

# Let's Test It: Using Ansible to Configure IOS-XE

- Ansible is executed with the playbook file and the inventory file as inputs, as well as an optional flag “-v” to show additional verbosity, to see the show command output in the terminal
- Both tasks are successful, the first task is highlighted yellow to show a change took place and the second task is green because it did not change anything, just a show command was used

```
(brkcrt) ansible$ ansible-playbook -i inventory.ini pb-configure-snmp.yaml -v
Using /etc/ansible/ansible.cfg as config file

PLAY [PLAY 1 - DEPLOYING SNMP CONFIGURATIONS ON IOS] *****

TASK [TASK 1 in PLAY 1 - CONFIGURE SNMP LINES] *****
[WARNING]: To ensure idempotency and correct diff the input configuration lines
should be similar to how they appear if present in the running configuration on
device
changed: [csr1] => {"banners": {}, "changed": true, "commands": ["snmp-server commu
nity belk-demo R0", "snmp-server location VEGAS", "snmp-server contact JASON_BELK"]
, "updates": ["snmp-server community belk-demo R0", "snmp-server location VEGAS", "
snmp-server contact JASON_BELK"]}

TASK [TASK 2 in PLAY 1 - VERIFY SNMP LINES PRESENT] *****
ok: [csr1] => {"changed": false, "stdout": ["snmp-server community belk-demo R0\sn
mp-server location VEGAS\nsnmp-server contact JASON_BELK"], "stdout_lines": [["snmp
-server community belk-demo R0", "snmp-server location VEGAS", "snmp-server contact
JASON_BELK"]]}

PLAY RECAP *****
csr1                : ok=2    changed=1    unreachable=0    failed=0    skip
ped=0    rescued=0    ignored=0

(brkcrt) ansible$
```

## 3.4 Utilize Ansible to configure an IOS XE device

```
---
- name: PLAY 1 - DEPLOYING SNMP CONFIGURATIONS ON IOS
  [REDACTED]
  - name: "TASK 1 in PLAY 1 - CONFIGURE SNMP LINES"
    cisco.ios.ios_config:
      lines:
        - snmp-server community belk-demo RO
        - snmp-server location VEGAS
        - snmp-server contact ADMIN
  - name: "TASK 2 in PLAY 1 - VERIFY SNMP LINES PRESENT"
    cisco.ios.ios_command:
      commands:
        - "show run | include snmp-server"
```

Refer to the exhibit. An engineer needs to configure SNMP on a CSR router with hostname csr1. Which code snippet will complete the Ansible playbook?

\*A.

```
hosts: "csr1"
connection: network_cli
gather_facts: no
tasks:
```

B.

```
hosts: "Router1"
connection: cli
gather_facts: no
facts:
```

C.

```
hosts: "CSR"
connection: HTTP
gather_facts: yes
tasks:
```

D.

```
hosts: "CS-Router"
connection: HTTPS
gather_facts: yes
facts:
```

# How to prepare for the exam





# Developer.cisco.com study resources

The screenshot shows the Cisco DevNet website. The top navigation bar includes links for Discover, Technologies, Community, Support, Events, and New Announcement, along with a search icon, a 'SIGN UP FREE' button, and a 'LOG IN' button. Below the navigation bar, the breadcrumb 'Certification > 300-435 ENAUTO' is visible. The main heading is 'Automating and Programming Cisco Enterprise Solutions v1.0 (300-435)'. A paragraph below the heading describes the exam: 'Automating and Programming Cisco Enterprise Solutions v1.0 (ENAUTO 300-435) is a 90-minute exam associated with the CCNP Enterprise Certification and DevNet Professional Certification. This exam tests a candidate's knowledge of implementing Enterprise automated solutions, including programming concepts, Python programming, APIs, controllers and automation tools. The course, Implementing Cisco Enterprise Automation Solutions, helps candidates to prepare for this exam.' A blue button labeled 'Sign up for updates' is positioned at the bottom left of the main content area.

DEVNET Discover Technologies Community Support Events New Announcement SIGN UP FREE LOG IN

Certification > 300-435 ENAUTO

## Automating and Programming Cisco Enterprise Solutions v1.0 (300-435)

Automating and Programming Cisco Enterprise Solutions v1.0 (ENAUTO 300-435) is a 90-minute exam associated with the CCNP Enterprise Certification and DevNet Professional Certification. This exam tests a candidate's knowledge of implementing Enterprise automated solutions, including programming concepts, Python programming, APIs, controllers and automation tools. The course, Implementing Cisco Enterprise Automation Solutions, helps candidates to prepare for this exam.

Sign up for updates

## Exam overview

The screenshot shows the 'Exam overview' section of the Cisco DevNet website. The top bar indicates '10%' completion and '1.0 Network Programmability Foundation' as the current section. The 'Exam Topics' section lists three topics: 1.1 Utilize common version control operations with git (add, clone, push, commit, diff, branching, merging conflict), 1.2 Describe characteristics of API styles (REST and RPC), and 1.3 Describe the challenges encountered and patterns used when consuming APIs synchronously and. The 'Study Material' section states: 'These resources are meant to supplement your learning experience and exam preparation. They are NOT designed to serve as a complete self-study program, but intended only as a suggested starting point. Login to access these materials.' Below this, there are three bullet points with links: 'Setting up your Linux (Ubuntu) workstation as a development environment', 'Setting up your Windows workstation as a development environment', and 'Setting up your macOS workstation as a development environment'.

10% 1.0 Network Programmability Foundation

### Exam Topics

- 1.1 Utilize common version control operations with git (add, clone, push, commit, diff, branching, merging conflict)
- 1.2 Describe characteristics of API styles (REST and RPC)
- 1.3 Describe the challenges encountered and patterns used when consuming APIs synchronously and

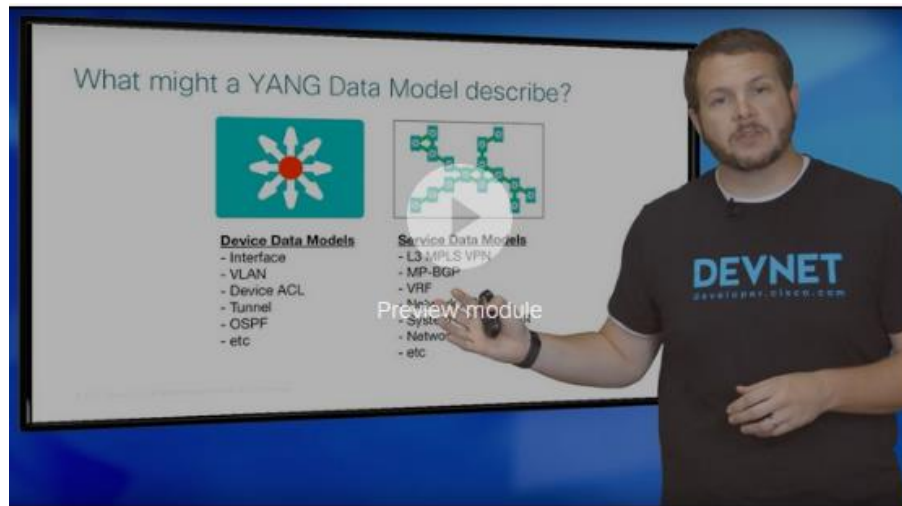
### Study Material

These resources are meant to supplement your learning experience and exam preparation. They are NOT designed to serve as a complete self-study program, but intended only as a suggested starting point. Login to access these materials.

- [Setting up your Linux \(Ubuntu\) workstation as a development environment](#)
- [Setting up your Windows workstation as a development environment](#)
- [Setting up your macOS workstation as a development environment](#)

# DevNet Video Courses

<https://developer.cisco.com>



## Network Device APIs

Network programmability is more than sending CLI with Python. Learn about the latest in programmatic device interfaces in this module.



Play module

- |  |       |
|--|-------|
| Getting the "YANG" of it with Standard Data Models     | 18:41 |
| Goodbye SNMP <hello> NETCONF!                          | 27:00 |
| Learn to CRUD with GET, POST and DELETE using RESTCONF | 22:41 |

# Community – Study groups



The Cisco Learning Network

[Log in](#) | [Register](#) [Help](#)

[Home](#) ▾

[Certifications](#) ▾

[Community](#) ▾

[Learning Center](#) ▾

[IT Careers](#) ▾

[Store](#)

[Browse](#) ▾

[Search](#)

[Home](#) > [Enterprise Networking Certifications Study Group](#)

## Enterprise Networking Certifications Study Group

[Overview](#)

[Content](#)

[People](#)

[Log in](#) to follow, share, and participate in this socialgroup. Not a member? [Join Now!](#)

Welcome to the Enterprise Networking Technology track Study Group, the place on the Cisco Learning Network where you can ask questions, share ideas and connect with other members as you prepare for your Certification exams.



**Join the Discussion**  
[Participate Now](#)



**Find People**  
[Connect Now](#)



**Member Created Documents**  
[View Now](#)

### Meet Your Community Managers

This team can help you locate the resources you need to meet your study goals and act as your advocate on the Cisco Learning Network.



**Karlo Bobiles**  
[Connect with me!](#)

# Cisco courses from CLN

The Cisco Learning Network Store



Certification Training ▾

Technology Training ▾

All Training ▾

[Home](#) / [On Demand E-Learning](#)

CCNP Enterprise, DevNet Professional - Concentration



## Implementing Automation for Cisco Enterprise Solutions (ENAI) v1.2

Continuing Education Credits: 24 ⓘ



Labs



Self-Paced  
Training



Video  
Training



Access Duration: 180 days



Share

### Overview

Implementing Automation for Cisco Enterprise Solutions (ENAI) v1.2 teaches you how to implement Cisco Enterprise automated solutions, including programming concepts, orchestration, telemetry, and automation tools.

# Cisco Live On-Demand Library

(www.ciscolive.com)

## On-Demand Library

Filters

Clear

SD-Access

My Favorites

+ EVENT

+ FEATURED

+ TECHNOLOGY

+ PROGRAM

+ FOCUS

+ SOLUTIONS

+ TECHNICAL LEVEL

"SD-Access" x

152 sessions

**Cisco SD-Access - A Look Under the Hood - BRKCRS-2810**

Event:  
2019 Barcelona

Shawn Wargo, PRINCIPAL ENGINEER.TECHNICAL MARKETING - **Distinguished Speaker**

**Cisco SD-Access - A Look Under the Hood - BRKCRS-2810**

Event:  
2018 Barcelona

Shawn Wargo, PRINCIPAL ENGINEER.TECHNICAL MARKETING - **Distinguished Speaker**

**Cisco DNA SD-Access (SDA) - Introduction to SDA Fabric - BRKARC-2009**

Event:  
2019 San Diego

Prabhjit Bagga, SR ENGINEER.IT ENGINEERING

# Technical Session Surveys

- Attendees who fill out a minimum of four session surveys and the overall event survey will get Cisco Live branded socks!
- Attendees will also earn 100 points in the Cisco Live Game for every survey completed.
- These points help you get on the leaderboard and increase your chances of winning daily and grand prizes.



# Cisco Learning and Certifications

From technology training and team development to Cisco certifications and learning plans, let us help you empower your business and career. [www.cisco.com/go/certs](http://www.cisco.com/go/certs)

## Pay for Learning with Cisco Learning Credits

(CLCs) are prepaid training vouchers redeemed directly with Cisco.



## Learn

### Cisco U.

IT learning hub that guides teams and learners toward their goals

### Cisco Digital Learning

Subscription-based product, technology, and certification training

### Cisco Modeling Labs

Network simulation platform for design, testing, and troubleshooting

### Cisco Learning Network

Resource community portal for certifications and learning



## Train

### Cisco Training Bootcamps

Intensive team & individual automation and technology training programs

### Cisco Learning Partner Program

Authorized training partners supporting Cisco technology and career certifications

### Cisco Instructor-led and Virtual Instructor-led training

Accelerated curriculum of product, technology, and certification courses



## Certify

### Cisco Certifications and Specialist Certifications

Award-winning certification program empowers students and IT Professionals to advance their technical careers

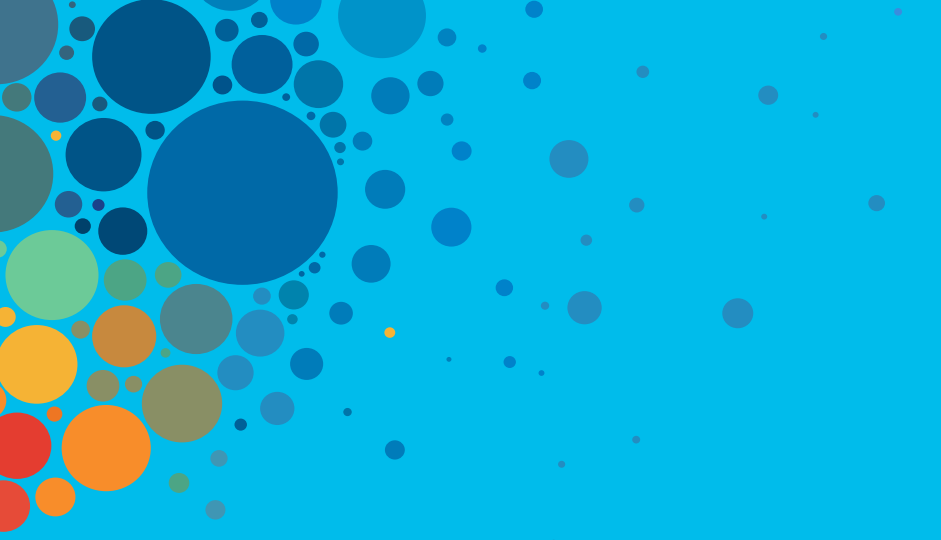
### Cisco Guided Study Groups

180-day certification prep program with learning and support

### Cisco Continuing Education Program

Recertification training options for Cisco certified individuals

Here at the event? Visit us at **The Learning and Certifications lounge at the World of Solutions**



# 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](http://www.CiscoLive.com/on-demand)





The bridge to possible

# Thank you

CISCO *Live!*



#CiscoLive