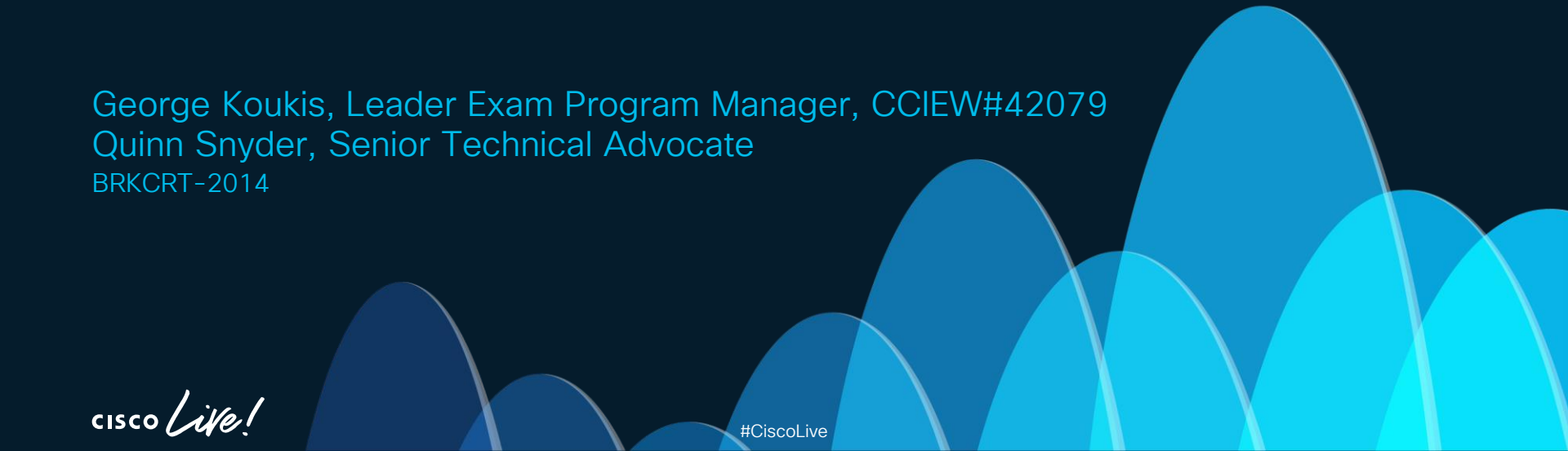
The bridge to possible

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

George Koukis, Leader Exam Program Manager, CCIEW#42079
Quinn Snyder, Senior Technical Advocate
BRKCRT-2014



cisco Live!

#CiscoLive

Cisco Webex App

https://ciscolive.ciscoevents.com/ciscolivebot/#BRKCRT-2014

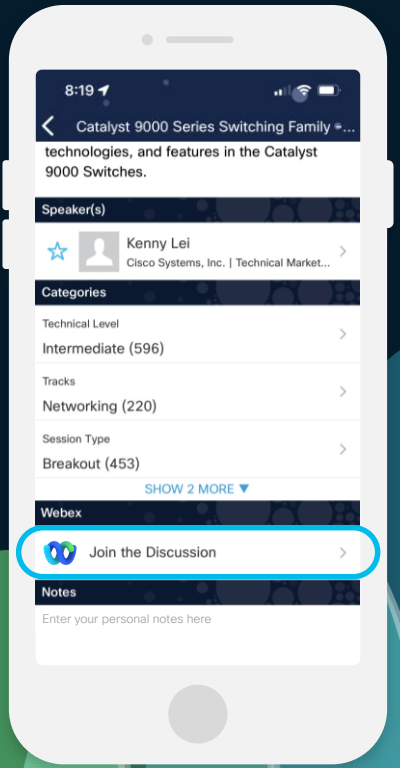
Questions?

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

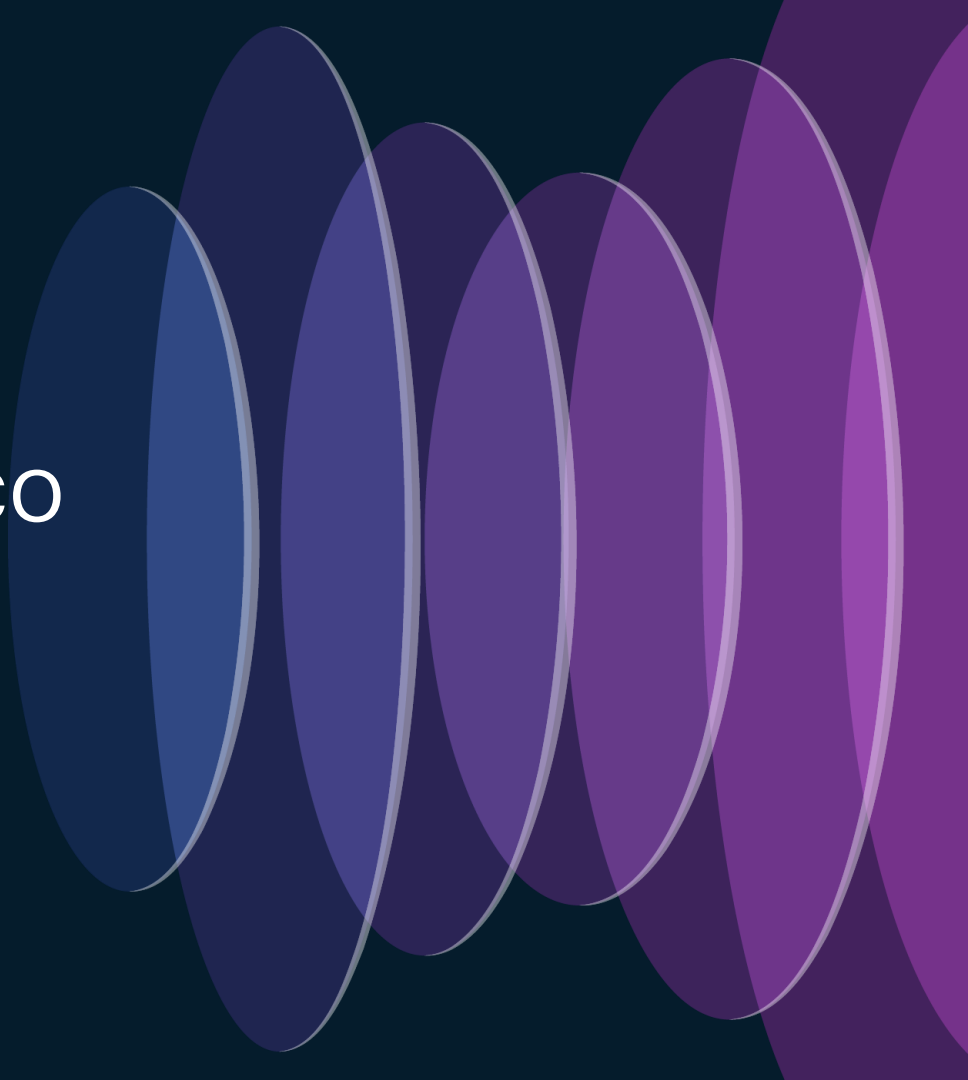
Webex spaces will be moderated by the speaker until June 7, 2024.



Agenda

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

Introduction to Cisco Certifications

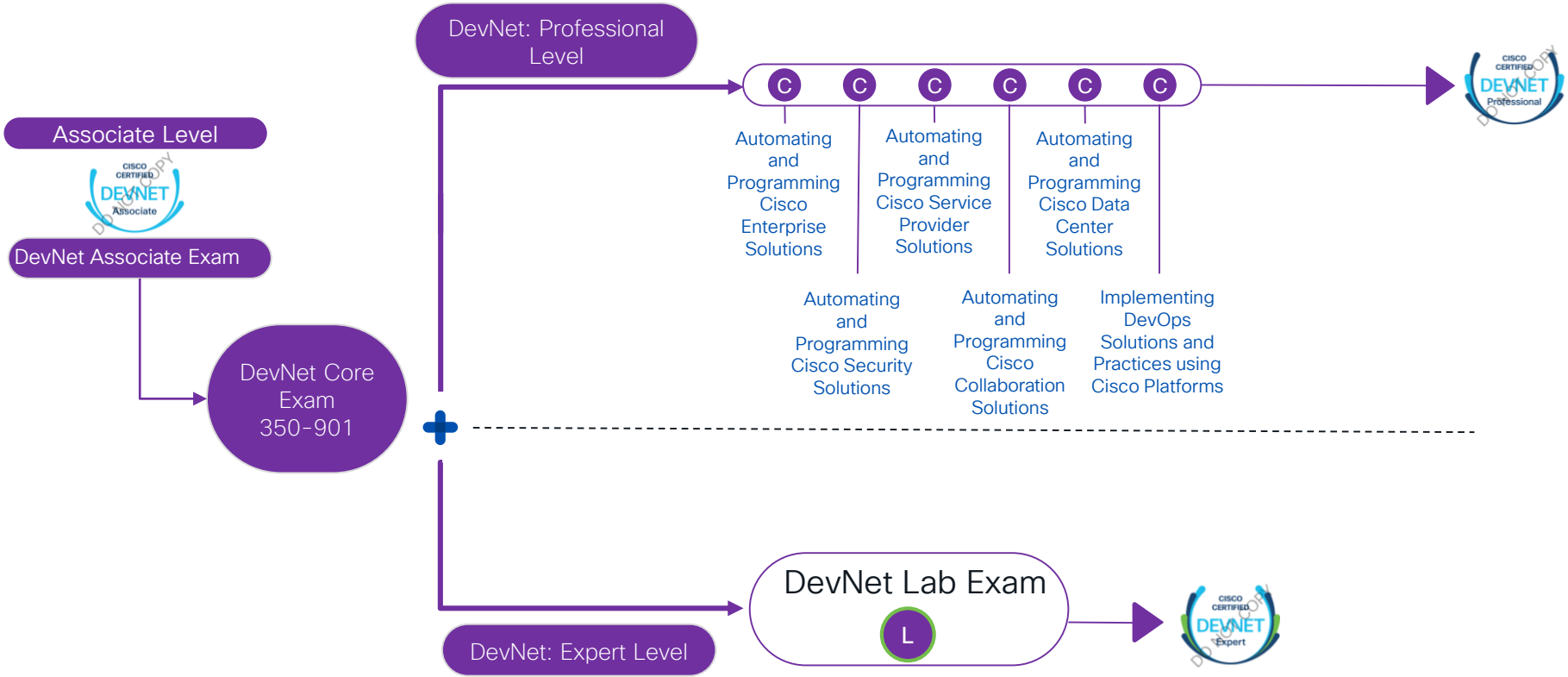


cisco Live!

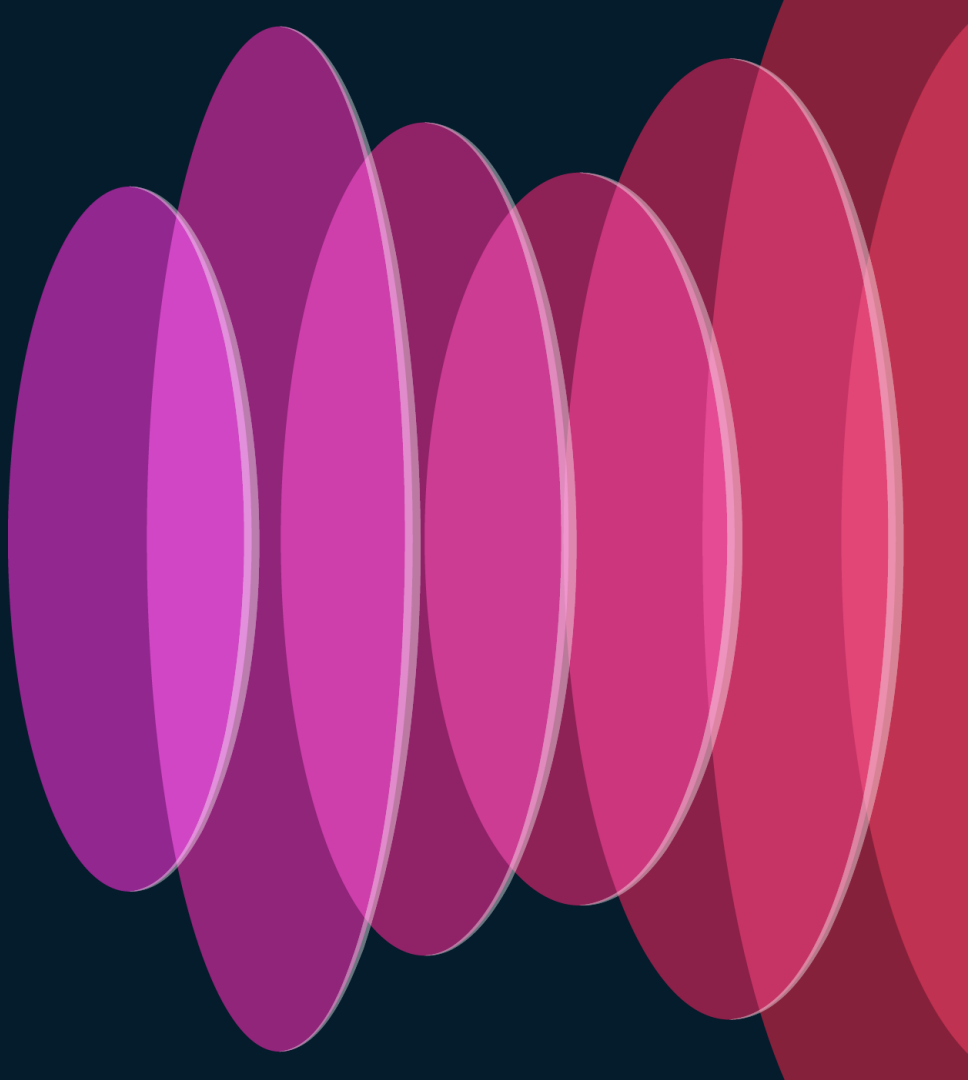
“DevNet professional, ENAUTO and Automation skills Required”

Cisco Career Certifications

Cisco DevNet-Automation certification track



Roadmap changes



Certification roadmap publishing process

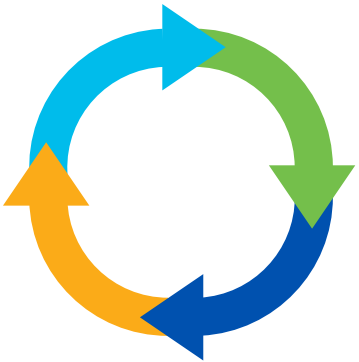
Stay in sync with the latest updates on a regular, rotating schedule

How it works:

1. Cisco **reviews** each technology on the same quarterly schedule each year to make sure our exams align with the latest Cisco technologies.
2. We **announce** blueprint changes 3–6 months in advance along with revised exam topics and release notes, if applicable.
3. We **publish** the updated exam 3–6 months after the exam blueprint publication, if applicable.

Data Center & Collaboration
Q1: Review/Job Task analysis
Q2: New blueprints published
Q3: Updated exam go live

Enterprise & DevNet
Q2: Review/Job Task analysis
Q3: New blueprints published
Q4: Updated exam go live



Security & CyberOps
Q4: Review/Job Task analysis
Q1: New blueprints published
Q2: Updated exam go live

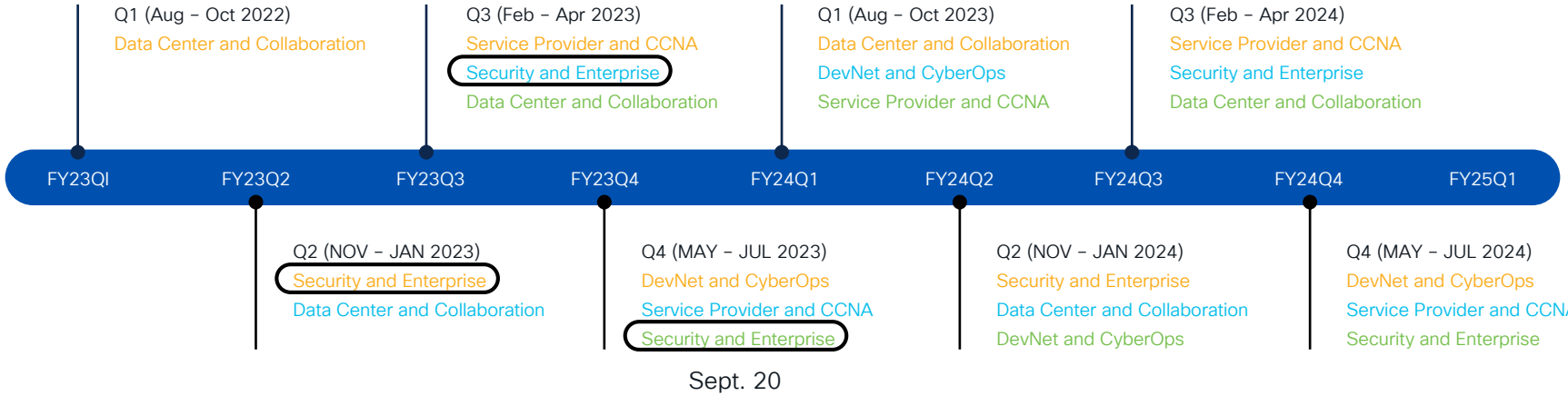
CCNA & Service Provider
Q3: Review/Job Task analysis
Q4: New blueprints published
Q1: Updated exam go live

Dates shown reflect Cisco's fiscal year calendar.
Q1: August–October, Q2: November–January, Q3: February–April, Q4: May–July

Certification roadmap

How it works:

1. Cisco reviews each technology on the same quarterly schedule each year to make sure our exams align with the latest Cisco technologies.
2. We announce blueprint changes 3-6 months in advance along with revised exam topics and release notes.
3. We publish the updated exam 3-6 months after the exam blueprint publication.

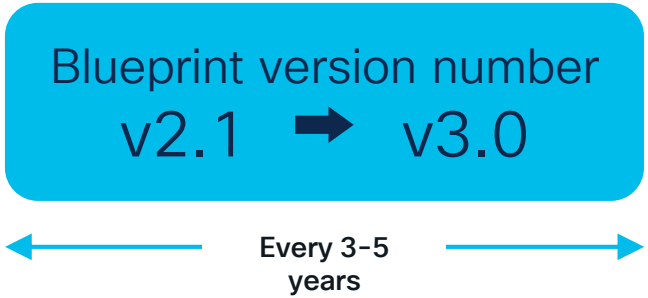


Sept. 20

Revision framework

Major revision

(Traditional revision model)



Large revisions

Major changes

Steep learning curve

Wider alignment
(Product & Technology)

Minor revision

(Agile revision model)

Smaller modular revisions

Incremental changes

Easy bite-size learning model

Frequent alignment
(Product & Technology)

ENAUTO v1.0 vs ENAUTO v1.1

1.6 Explain the benefits of using network configuration tools such as Ansible and Terraform ~~Puppet~~ for automating IOS XE platforms

2.1 Identify the JSON instance based on a YANG model (including YANG Suite)

2.2 Identify the XML instance based on a YANG model (including YANG Suite)

4.2 Describe the features and capabilities of Cisco DNA Center

- 4.2.c ~~Multivendor support (3rd party SDKs)~~
- 4.2 eSDA

4.4 Implement API requests for Cisco DNA Center to accomplish network management tasks

- 4.4.d SDA APIs

5.1 Describe features and capabilities of Cisco SD-WAN vManage ~~Certificate Management~~ APIs

https://learningnetwork.cisco.com/s/cisco-certification-roadmaps?tabset-94f5d=1

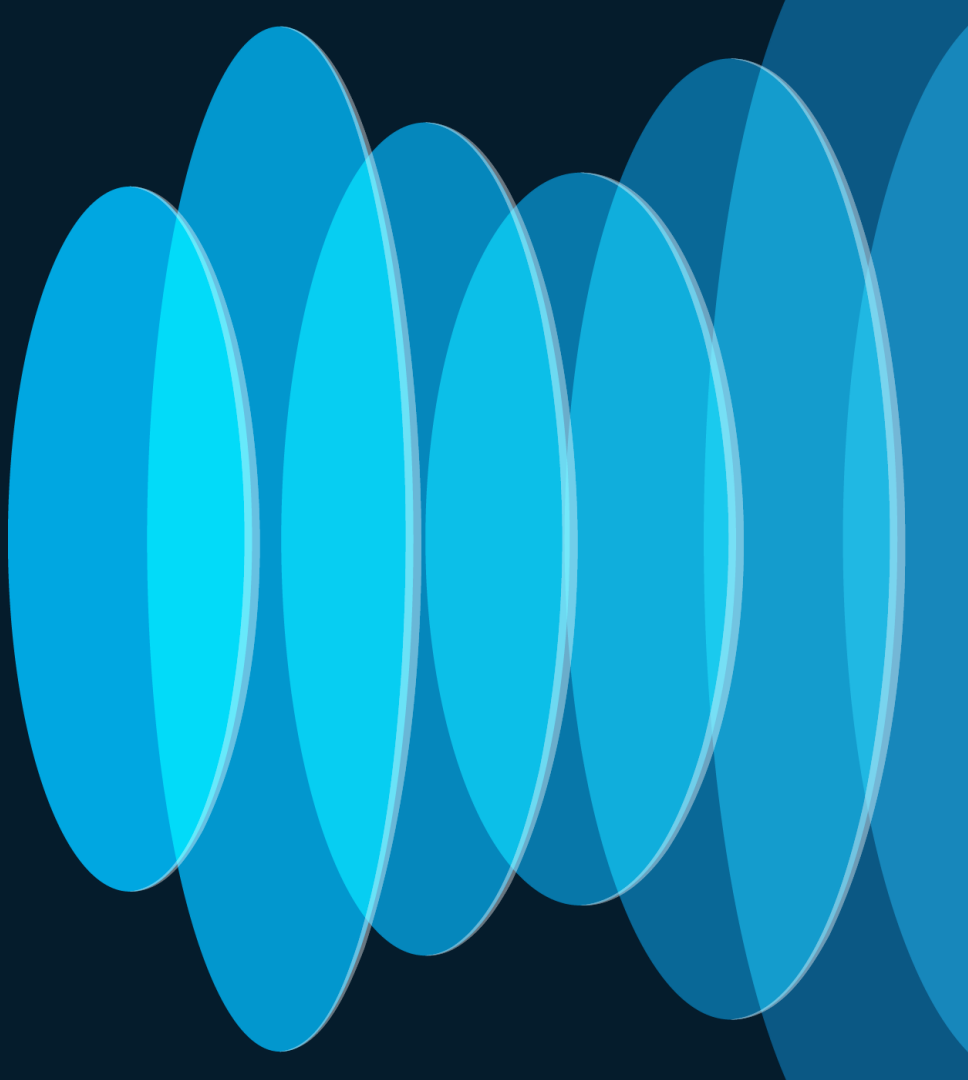
Upcoming Exam Changes

Select the technology tabs below for specific exam changes:

DevNet | CyberOps | Service Provider | CCNA | Security | **Enterprise** | Collaboration | Data Center | FAQs

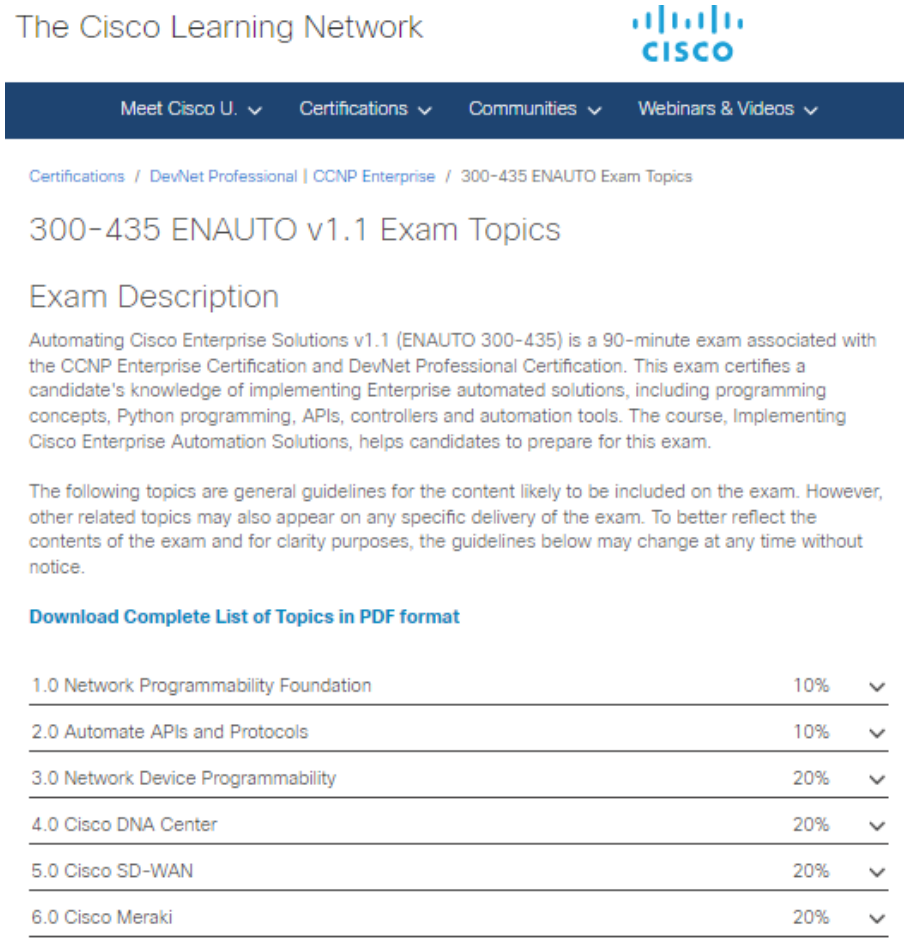
Exam Number	Release Notes	Exam Topic Blueprint	Learning Matrix	Hardware/ Software List
Cisco Enterprise Exams	CCNP Release Notes			
CCNP/CCIE Core Exam Updates				
350-401 ENCOR v1.1		Exam Topics		
CCNP Concentration Exam Updates				
300-410 ENARSI v1.1		Exam Topics		
300-415 ENSDWI v1.2		Exam Topics		
300-420 ENSLD v1.1		Exam Topics		
300-425 ENWLSD v1.1		Exam Topics		
300-430 ENWLSI v1.1		Exam Topics		
300-435 ENAUTO v1.1		Exam Topics		
300-440 ENCC v1.0		Exam Topics		

What to expect on the exam



Exam Blueprint

https://learningnetwork.cisco.com/s/enauto-exam-topics



The Cisco Learning Network

Meet Cisco U. | Certifications | Communities | Webinars & Videos

Certifications / DevNet Professional | CCNP Enterprise / 300-435 ENAUTO Exam Topics

300-435 ENAUTO v1.1 Exam Topics

Exam Description

Automating Cisco Enterprise Solutions v1.1 (ENAUTO 300-435) is a 90-minute exam associated with the CCNP Enterprise Certification and DevNet Professional Certification. This exam certifies 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.

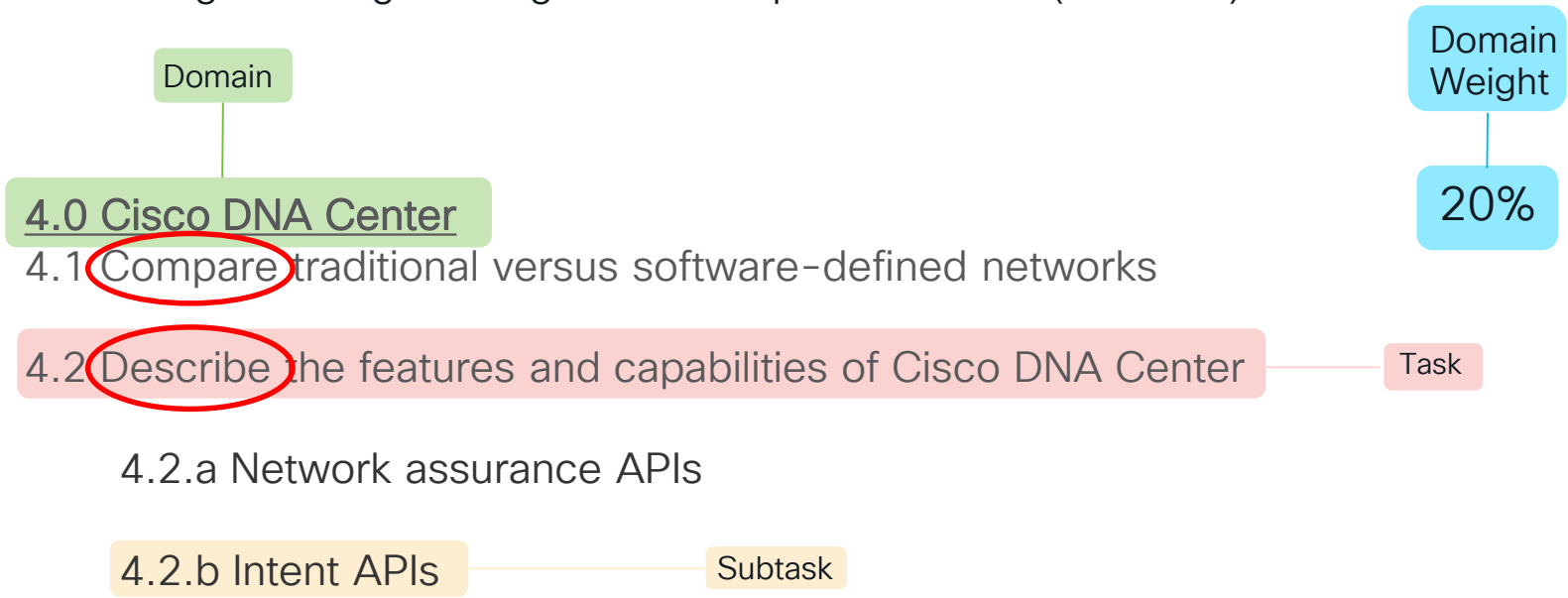
The following topics are general guidelines for the content likely to be included on the exam. However, other related topics may also appear on any specific delivery of the exam. To better reflect the contents of the exam and for clarity purposes, the guidelines below may change at any time without notice.

Download Complete List of Topics in PDF format

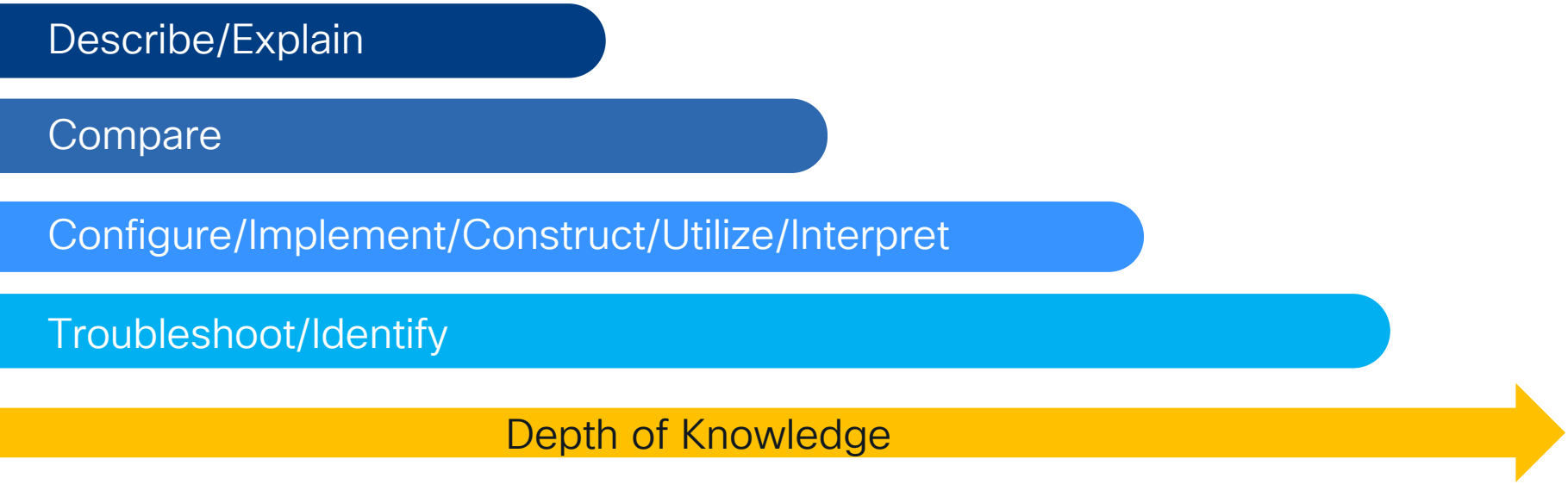
1.0 Network Programmability Foundation	10%
2.0 Automate APIs and Protocols	10%
3.0 Network Device Programmability	20%
4.0 Cisco DNA Center	20%
5.0 Cisco SD-WAN	20%
6.0 Cisco Meraki	20%

Interpret the Blueprint:

Automating and Programming Cisco Enterprise Solutions (300-435)



Blueprint Verbs



Types of questions

Multiple choice

Drag and drop

Lablets are coming...

Describe question

1.5 Describe the benefits of Python virtual environments

What is a benefit of Python virtual environments?

A. separates project dependencies

B. automates software upgrades

C. enhances DDos security

D. adds version control functionality

Describe question

1.5 Describe the benefits of Python virtual environments

What is a benefit of Python virtual environments?

*A. separates project dependencies

B. automates software upgrades

C. enhances DDos security

D. adds version control functionality

Compare question

4.1 Compare traditional versus software-defined networks

What is the difference between traditional and software-defined networks?

*A. Traditional networks have fixed capacity, whereas software-defined networks are flexible and agile.

B. Traditional networks are centralized controlled, whereas software-defined are hardware-centric..

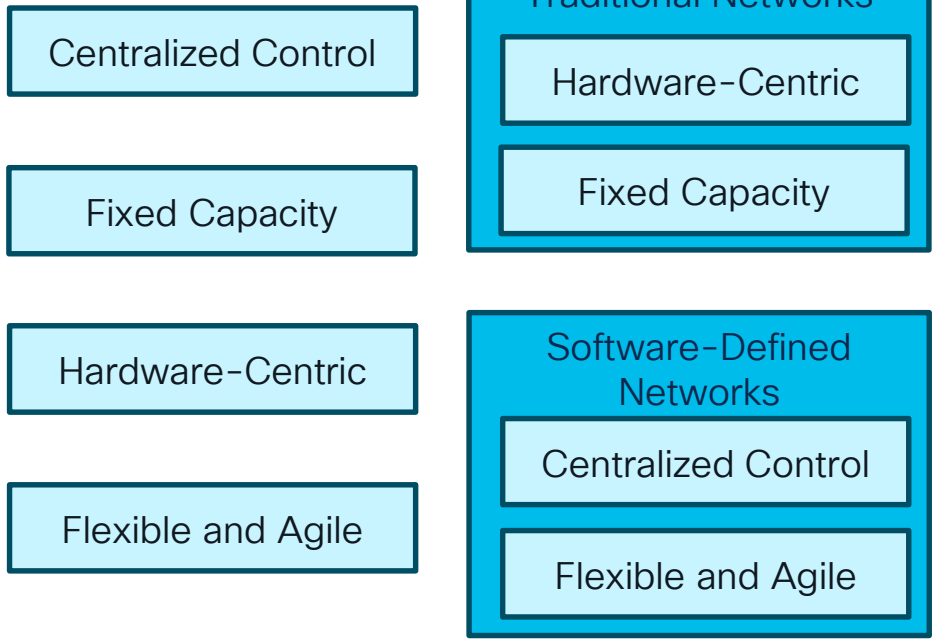
C. Traditional networks ...
D. Traditional networks ...

Compare question

4.1 Compare traditional versus software-defined networks

OR like this

Drag and drop the characteristics from the left onto the network types on the right.



How to prepare

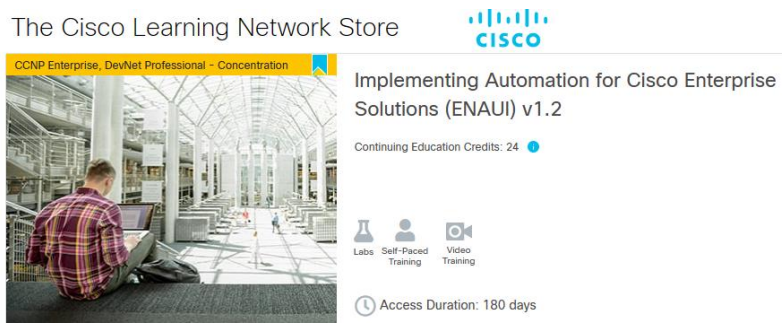
Describe/Explain

Compare

Configure/Implement/Construct/Utilize/Interpret

Troubleshoot/Identify

Depth of Knowledge



Configure question

3.3 Configure device using RESTCONF API utilizing Python requests library

Drag and drop the code from the bottom onto the box where the code is missing to remove loopback100 using RESTCONF. Not all options are used.

```
import requests
import json

url = "https://ios-xe-mgmt-latest.cisco.com:9443/restconf/ \
       data/ietf-interfaces:interfaces/interface=Loopback100"

[          ] = {}

headers = {
  'Content-Type': 'application/yang-data+json',
  '[          ]': 'application/yang-data+json',
  'Authorization': '[          ] ZGV2ZWxvcGVyOkMxc2NvMTIzNDU='
}

response = requests.request("[          ]", url, headers=headers, data=payload)

print(response.text)
```

Bearer	DELETE	payload	data
Basic	GET	Accept	

Configure question

3.3 Configure device using RESTCONF API utilizing Python requests library

Drag and drop the code from the bottom onto the box where the code is missing to remove loopback100 using RESTCONF. Not all options are used.

```
import requests
import json

url = "https://ios-xe-mgmt-latest.cisco.com:9443/restconf/ \
       data/ietf-interfaces:interfaces/interface=Loopback100"

 [          ] = {}

headers = {
  'Content-Type': 'application/yang-data+json',
  '[          ]': 'application/yang-data+json',
  'Authorization': '[          ] ZGV2ZWxvcGVyOkMxc2NvMTIzNDU='
}

response = requests.request("[          ]", url, headers=headers, data=payload)

print(response.text)
```

Bearer | DELETE | payload | data

Basic | GET | Accept

How to prepare

Describe/Explain

Compare

Configure/Implement/Construct/Utilize/Interpret

Troubleshoot/Identify

Troubleshoot question

5.6 Troubleshoot a Cisco SD-WAN deployment using vManage APIs

```
import requests

url = "https://sandbox-sdwan-2.cisco.com:443/dataservice/device/monitor"

payload = {
'Cookie': 'JSESSIONID=VlGlzAg8MQaHRpvyshPVk9hTkCRe5PLo1rx2RuK7.81ac6722-a226-4411-9d5d-45c0ca7d567b'
}
headers = {
  'X-XSRF-TOKEN': '154A22D9C79DE811BE7A2693BEE288C3',
}

response = requests.request("GET", url, headers=headers, data=payload, verify=False)

print(response.text)
```

Refer to the exhibit. A network engineer is tasked to retrieve device statuses on a SD-WAN deployment. To accomplish the requirement, the network creates a script utilizing the vManage API. When the script is run a login error appears. What must the engineer change on the script to fix the issue?

A. move the cookie in the headers
B. move the x-xsrf-token in the payload
C. remove the cookie from the code
D. remove the x-xsrf-token from the code

Troubleshoot question

5.6 Troubleshoot a Cisco SD-WAN deployment using vManage APIs

```
import requests

url = "https://sandbox-sdwan-2.cisco.com:443/dataservice/device/monitor"

payload = {
'Cookie': 'JSESSIONID=VlGlzAg8MQaHRpvyshPVk9hTkCRe5PLo1rx2RuK7.81ac6722-a226-4411-9d5d-45c0ca7d567b'
}
headers = {
  'X-XSRF-TOKEN': '154A22D9C79DE811BE7A2693BEE288C3',
}

response = requests.request("GET", url, headers=headers, data=payload, verify=False)

print(response.text)
```

Refer to the exhibit. A network engineer is tasked to retrieve device statuses on a SD-WAN deployment. To accomplish the requirement, the network creates a script utilizing the vManage API. When the script is run a login error appears. What must the engineer change on the script to fix the issue?

*A. move the cookie in the headers
B. move the x-xsrf-token in the payload
C. remove the cookie from the code
D. remove the x-xsrf-token from the code

How to prepare

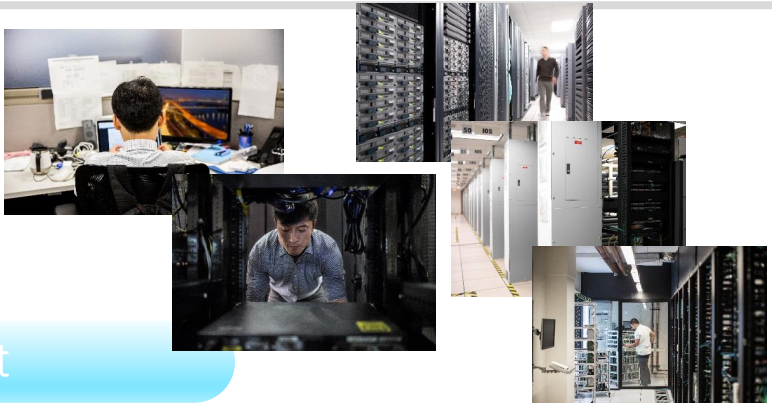
Describe/Explain

Compare

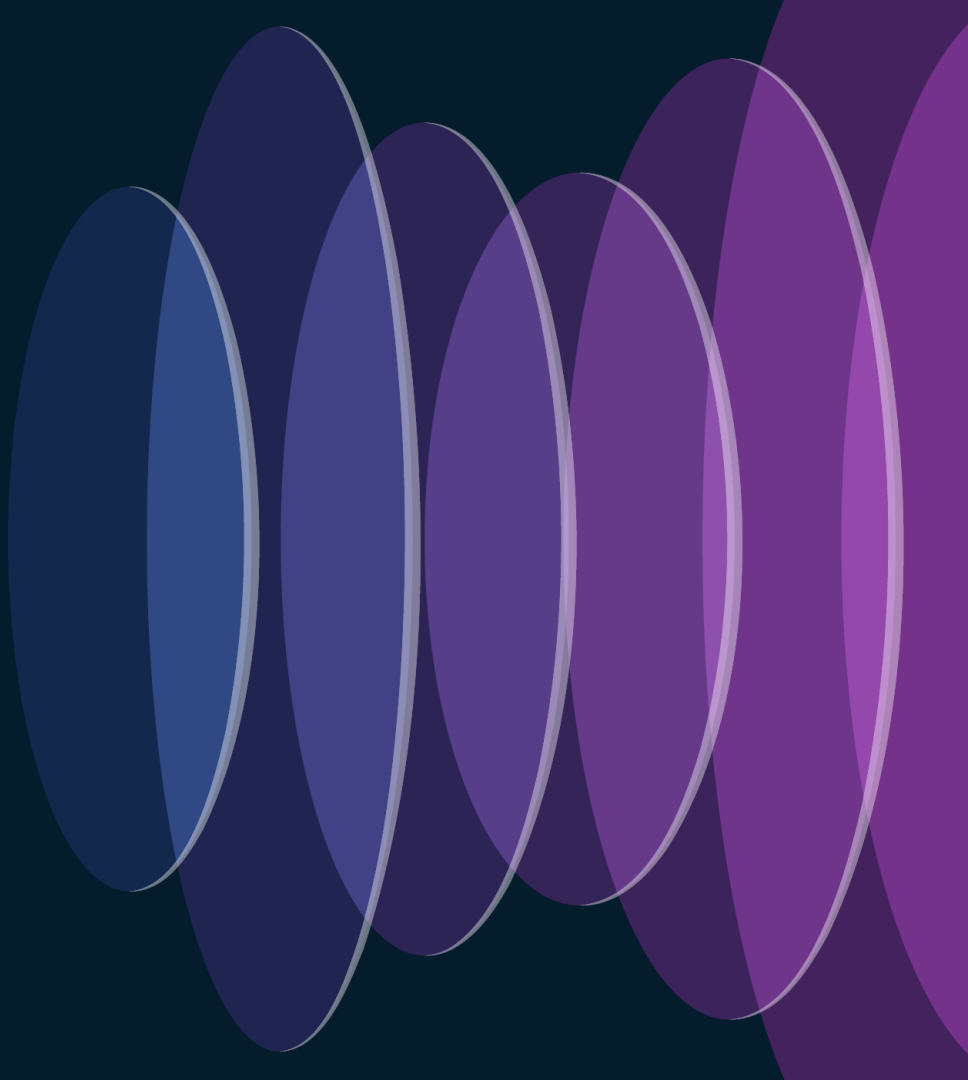
Configure/Implement/Construct/Utilize/Interpret

Troubleshoot/Identify

Depth of Knowledge



Exam vs Real life



What will be covered

2.3 Interpret a YANG module tree generated per RFC8340

3.4 Utilize Ansible to configure an IOS XE device

Implement API requests for Cisco DNA Center to accomplish network management tasks using these APIs

4.5.a Network discovery and device APIs

5.3 Construct API requests for Cisco SD-WAN vManage Administration APIs

What will be covered

2.3 Interpret a YANG module tree generated per RFC8340

3.4 Utilize Ansible to configure an IOS XE device

Implement API requests for Cisco DNA Center to accomplish network management tasks using these APIs

4.5.a Network discovery and device APIs

5.3 Construct API requests for Cisco SD-WAN vManage Administration APIs

```
$ pyang enauto-device-module.yang -f tree
module: enauto-device-module

    +--rw device
         +--rw hostname              string
         +--rw vrfs*                  string
         +--rw router-id?         inet:ipv4-address
         +--rw vlans* [id]
            +--rw id       uint16
            +--rw name?    string
```

Refer to the exhibit. Which partial YANG model is represented by the pyang tree?

A.

```
leaf router-id {
    type inet:ipv4-address;
}
list vlans {
    key id;
    leaf id {
      type uint16;
      mandatory true;
    }
```

B.

```
leaf router-id {
    type ipv4-address;
}
list vlans {
    key id;
    leaf id {
      type uint16;
      mandatory true;
    }
```

C.

```
leaf router-id {
    type inet:ipv4-address;
}
list vlans {
    leaf id {
      type uint16;
    }
```

D.

```
leaf router-id {
    type inet:ipv4-address;
}
leaf-list vlans {
    key id;
    leaf id {
      type string;
      mandatory true;
    }
```

YANG models and `pyang`

- RFC8430 defines YANG tree diagrams
  - YANG tree provides visualization of YANG module, nesting, types, and options
  - *Does not include* specific restrictions/validators within the types
- Can use `pyang` utility to generate trees
  - Installed via pip in Python
  - Called using `pyang <module> -f tree`

```
2.6.  Node Representation

   Each node in a YANG module is printed as:

     <status>--<flags> <name><opts> <type> <if-features>

       <status> is one of:
         +  for current
         x  for deprecated
         o  for obsolete

       <flags> is one of:
         rw  for configuration data nodes and choice nodes
         ro  for non-configuration data nodes and choice nodes,
             output parameters to rpcs and actions, and
             notification parameters
         -w  for input parameters to rpcs and actions
         -u  for uses of a grouping
         -x  for rpcs and actions
         -n  for notifications
         mp  for nodes containing a "mount-point" extension statement

         Case nodes do not have any <flags>.
```

```
       <name> is the name of the node
         (<name>) means that the node is a choice node
        :(<name>) means that the node is a case node

         If the node is augmented into the tree from another module,
         its name is printed as <prefix>:<name>, where <prefix> is the
         prefix defined in the module where the node is defined.

         If the node is a case node, there is no space before the
         <name>.

       <opts> is one of:
         ?  for an optional leaf, choice, anydata, or anyxml
         !  for a presence container
         *  for a leaf-list or list
         [<keys>] for a list's keys
         /  for a top-level data node in a mounted module
         @  for a top-level data node of a module identified in a
            mount point parent reference

       <type> is the name of the type for leafs and leaf-lists
```

From https://datatracker.ietf.org/doc/html/rfc8340

What will be covered

2.3 Interpret a YANG module tree generated per RFC8340

3.4 Utilize Ansible to configure an IOS XE device

Implement API requests for Cisco DNA Center to accomplish network management tasks using these APIs

4.5.a Network discovery and device APIs

5.3 Construct API requests for Cisco SD-WAN vManage Administration APIs

```
---
  - name: PLAY 1 - DEPLOYING SNMP CONFIGURATIONS ON IOS
    [          ]

    - name: TASK 1 in PLAY 1 - Modifying the SNMP configuration
      cisco.ios.ios_snmp_server:
        config:
          communities:
          -    name: BRKCRT-2014-COMMUNITY
               ro: true
          contact: THE_STAFF
          location: THE_HOTEL_CONFERENCE_ROOM
        state: merged
    - name: "TASK 2 in PLAY 1 - Verify the SNMP configuration exists"
      cisco.ios.ios_snmp_server:
        state: gathered
```

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: "csr1"
connection: ssh
gather_facts: no
tasks:
```

C.
```
hosts: "CSR"
connection: RESTCONF
gather_facts: no
tasks:
```

D.
```
hosts: "csr1"
connection: HTTPS
gather_facts: no
tasks:
```

Demo Overview: Using Ansible to Configure IOS-XE

```
[iosxe]
csr1 ansible_host=10.10.20.48 ansible_network_os=cisco.ios.ios

[all:vars]
ansible_user=developer
ansible_ssh_pass=C1sco12345
ansible_connection=network_cli
```

```
---
  - name: PLAY 1 - DEPLOYING SNMP CONFIGURATIONS ON IOS
    hosts: "csr1"
    connection: network_cli
    gather_facts: no
    tasks:
      - name: TASK 1 in PLAY 1 - Modifying the SNMP configuration
        cisco.ios.ios_snmp_server:
          config:
            communities:
            -   name: CLUS-DEMO
                ro: true
            contact: BEARDED_GUY
            location: LAS_VEGAS
          state: merged
      - name: "TASK 2 in PLAY 1 - Verify the SNMP configuration exists"
        cisco.ios.ios_snmp_server:
          state: gathered
```

- The Ansible inventory file defines the device credentials, IP address and device type within the project
- The Ansible playbook lists a series of tasks to be executed upon the device `csr1`, defined in inventory
- 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

- `-v` flag indicates additional output; command output
- Orange text: change
- Green text: no change; output

```
[I] [brkcrt2014] ansible [main]× » ansible-playbook pb-configure-snmp.yaml -v
Using /Users/qsnyder/dev/BRKCRT-2014/ansible/ansible.cfg as config file

PLAY [PLAY 1 - DEPLOYING SNMP CONFIGURATIONS ON IOS]
*********************************************************************************
[WARNING]: ansible-pylibssh not installed, falling back to paramiko

TASK [TASK 1 in PLAY 1 - Modifying the SNMP configuration]
*********************************************************************
changed: [csr1] => {"after": {"communities": [{"name": "CLUS-DEMO", "ro": true}], "contact": "BEARDED_GUY",
"location": "LAS_VEGAS"}, "before": {}, "changed": true, "commands": ["snmp-server contact BEARDED_GUY", "snmp-
server location LAS_VEGAS", "snmp-server community CLUS-DEMO ro"]}

TASK [TASK 2 in PLAY 1 - Verify the SNMP configuration exists]
*********************************************************************
ok: [csr1] => {"changed": false, "gathered": {"communities": [{"name": "CLUS-DEMO", "ro": true}], "contact":
"BEARDED_GUY", "location": "LAS_VEGAS"}}

PLAY RECAP
*****************************************************************************************************************************
csr1                       : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

What will be covered

2.3 Interpret a YANG module tree generated per RFC8340

3.4 Utilize Ansible to configure an IOS XE device

Implement API requests for Cisco DNA Center to accomplish network management tasks using these APIs

4.5.a Network discovery and device APIs

5.3 Construct API requests for Cisco SD-WAN vManage Administration APIs

```
BASE_URL = 'https://10.10.10.100'
DISCOVERY_URL = '/dna/intent/api/v1/discovery'

response = requests.post(BASE_URL + DISCOVERY_URL, headers=headers,
                         json=discovery, verify=False)
task_id = response.json()['response']['taskId']

time.sleep(10)
response = requests.get(BASE_URL + TASK_BY_ID_URL.format(task_id=task_id),
                        headers=headers, verify=False)
discovery_id = response['progress']
```

Refer to the exhibit. An engineer must use Cisco DNA Center to discover the connected devices based on a list of IP addresses. Which line of code needs to be added to the box where the code is missing to accomplish the requirement?

A.

```
discovery = {
    "name": "ENAUTO-Discovery",
    "discoveryType": "CDP",
    "ipAddressList": "10.255.3.100-10.255.3.199",
    "timeOut": "10s",
    "retryCount": 3,
    "preferredMgmtIPMethod": "LoopbackInterface"
    ]
}
```

B.

```
discovery = {
    "name": "ENAUTO-Discovery",
    "discoveryType": "LLDP",
    "ipAddressList": "10.255.3.100-10.255.3.199",
    "timeOut": "10s",
    "retryCount": 3,
    "preferredMgmtIPMethod": "UseLoopBack"
    ]
}
```

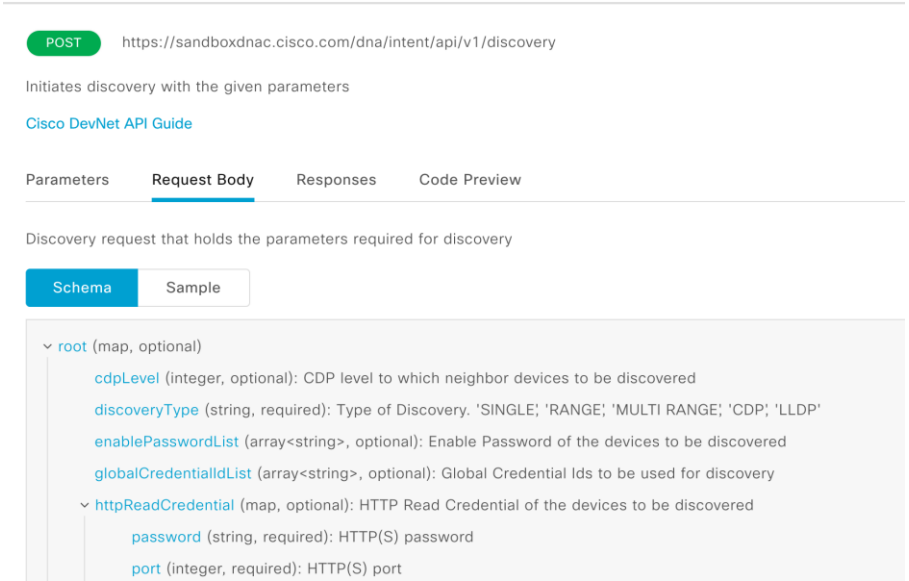
C.

```
discovery = {
    "name": "ENAUTO-Discovery",
    "discoveryType": "Range",
    "ipAddressList": "10.255.3.100-10.255.3.199",
    "timeOut": 10,
    "retryCount": 3,
    "preferredMgmtIPMethod": "UseLoopBack"
    ]
}
```

D.

```
discovery = {
    "name": "ENAUTO-Discovery",
    "discoveryType": "Range",
    "ipAddressList": "10.255.3.100-10.255.3.199",
    "timeOut": 10,
    "retryCount": 3,
    "preferredMgmtIPMethod": "LoopbackInterface"
    ]
}
```

Specific APIs and payloads



- Specific APIs on ”exam topics” should be reviewed in depth
- Should understand:
  - API paths
  - Payload required
  - HTTP verbs required for action
- Resources:
  - Sample code (DevNet)
  - Off-box API documentation
  - On-box API docs/explorer

What will be covered

2.3 Interpret a YANG module tree generated per RFC8340

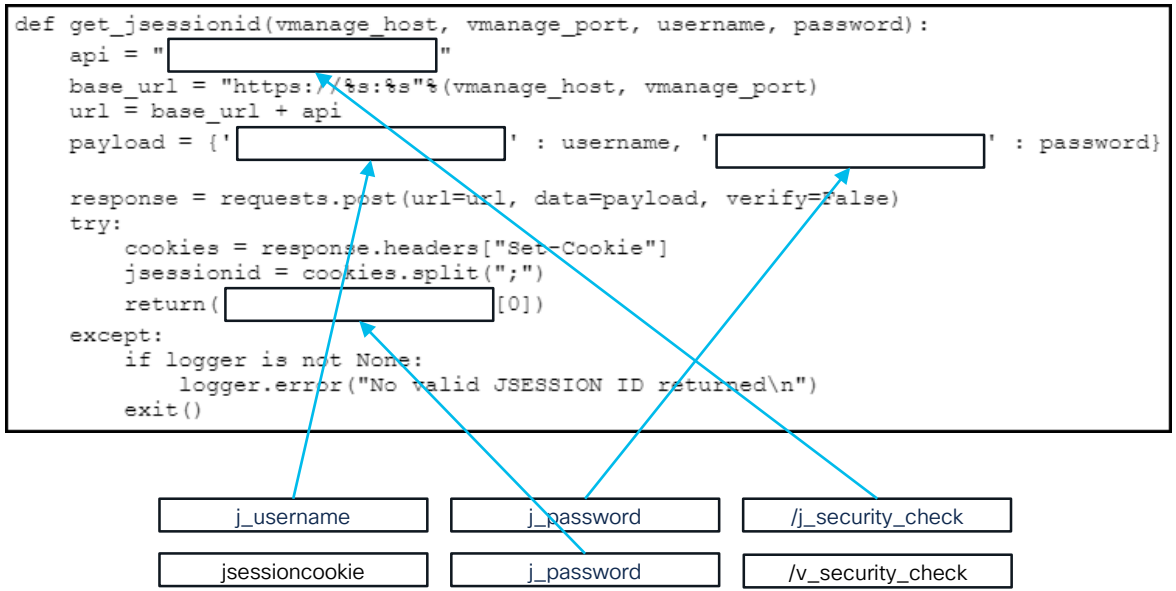
3.4 Utilize Ansible to configure an IOS XE device

Implement API requests for Cisco DNA Center to accomplish network management tasks using these APIs

4.5.a Network discovery and device APIs

5.3 Construct API requests for Cisco SD-WAN vManage Administration APIs

Drag and drop the code from the bottom onto the box where the code is missing to complete the script for a user to authenticate to a vManage controller. Not all options are used.



Using APIs requires authentication

```python
def get_dnac_jwt_token():
    response = requests.post(BASE_URL + AUTH_URL,
                             auth=HTTPBasicAuth(USERNAME, PASSWORD),
                             verify=False)
    token = response.json()['Token']
    return token
```

Cisco DNA Center: Basic auth, returns Token -> add'l calls

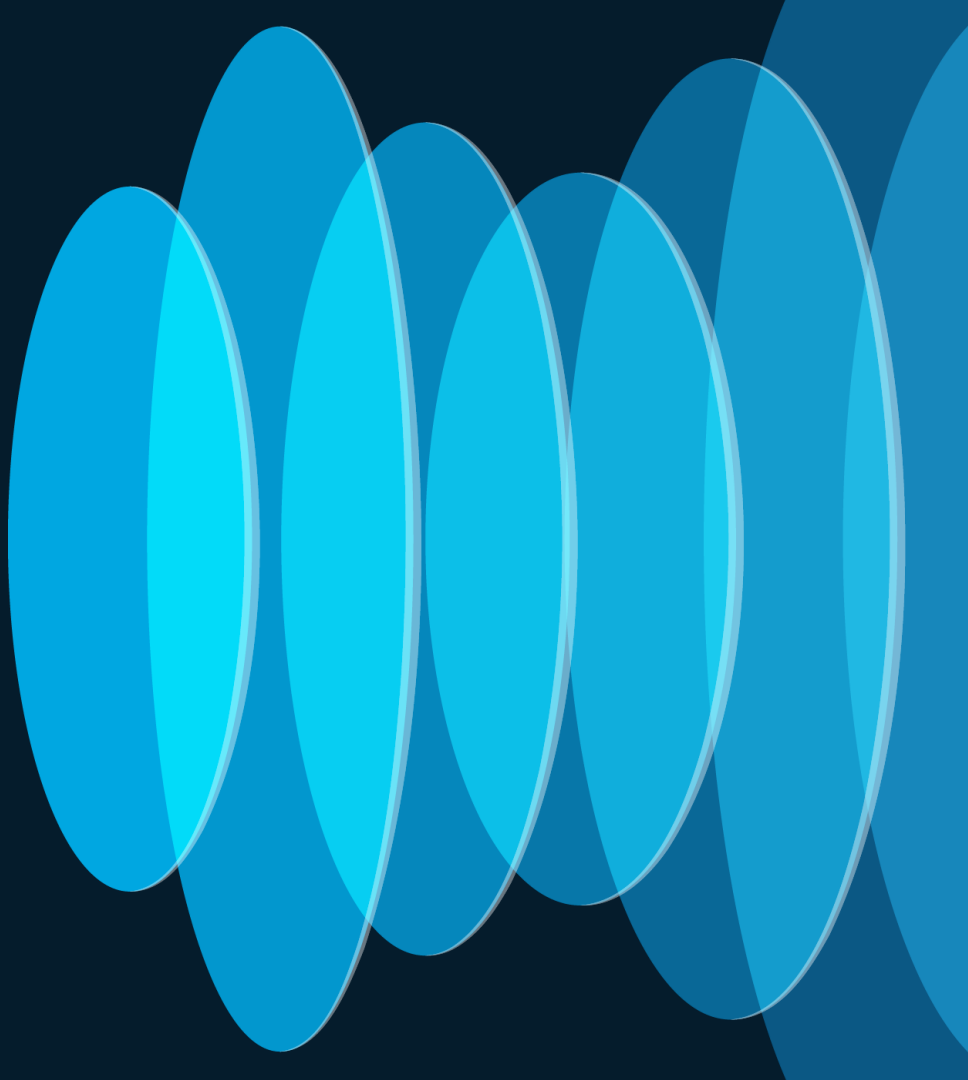
```python
from requests import Session
class NoRebuildAuthSession(Session):
    def rebuild_auth(self, prepared_request, response):
session = NoRebuildAuthSession()
API_KEY = '6bec40cf957de430a6f1f2baf056b99a4fac9ea0'
response = session.get('https://api.meraki.com/api/v1/organizations/', headers={'Authorization': f'Bearer
{API_KEY}'})
print(response.JSON())
```

Cisco Meraki Dashboard: API key
(with or without session)

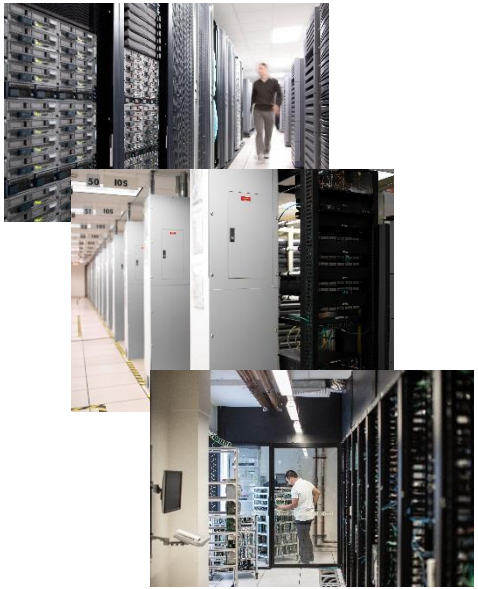
```python
class Authentication:

    @staticmethod
    def get_jsessionid(vmanage_host, vmanage_port, username, password):
        api = "/j_security_check"
        base_url = "https://%s:%s"%(vmanage_host, vmanage_port)
        url = base_url + api
        payload = {'j_username' : username, 'j_password' : password}

        response = requests.post(url=url, data=payload, verify=False)
        try:
            cookies = response.headers["Set-Cookie"]
            jsessionid = cookies.split(";")
            return(jsessionid[0])
        except:
            print("No valid JSESSION ID returned\n")
            exit()

    @staticmethod
    def get_token(vmanage_host, vmanage_port, jsessionid):
        headers = {'Cookie': jsessionid}
        base_url = "https://%s:%s"%(vmanage_host, vmanage_port)
        api = "/dataservice/client/token"
        url = base_url + api
        response = requests.get(url=url, headers=headers, verify=False)
        if response.status_code == 200:
            return(response.text)
        else:
            return None

Auth = Authentication()
jsessionid = Auth.get_jsessionid(vmanage_host,vmanage_port,vmanage_username,vmanage_password)
token = Auth.get_token(vmanage_host,vmanage_port,jsessionid)

if token is not None:
    header = {'Content-Type': "application/json",'Cookie': jsessionid, 'X-XSRF-TOKEN': token}
else:
    header = {'Content-Type': "application/json",'Cookie': jsessionid}
```

Cisco SDWAN: User auth -> ID -> token

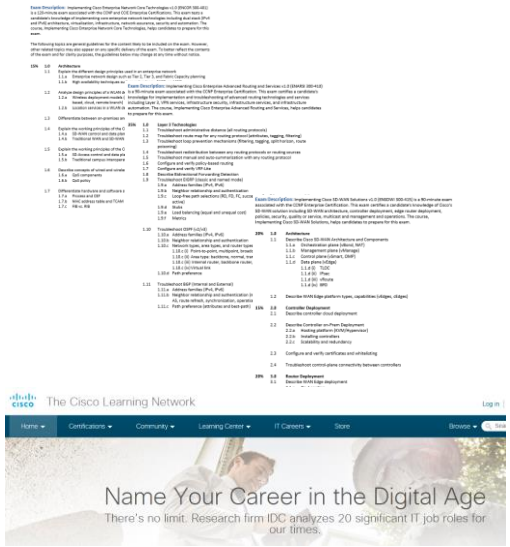
How to prepare for the exam



- Hands-on experience



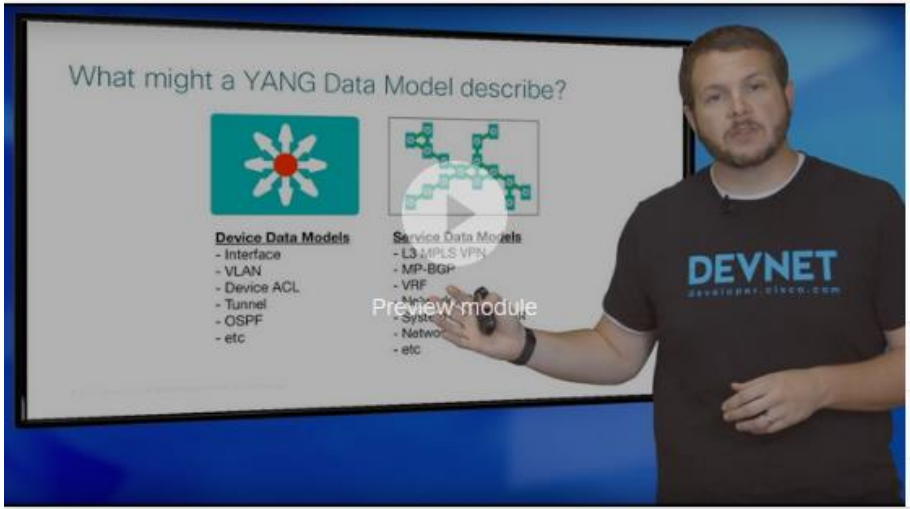
- Study the Blueprint and training. Join CLN-DevNet (study groups)



- Lab-Sandboxes-Code !!!

DevNet Video Courses

https://developer.cisco.com



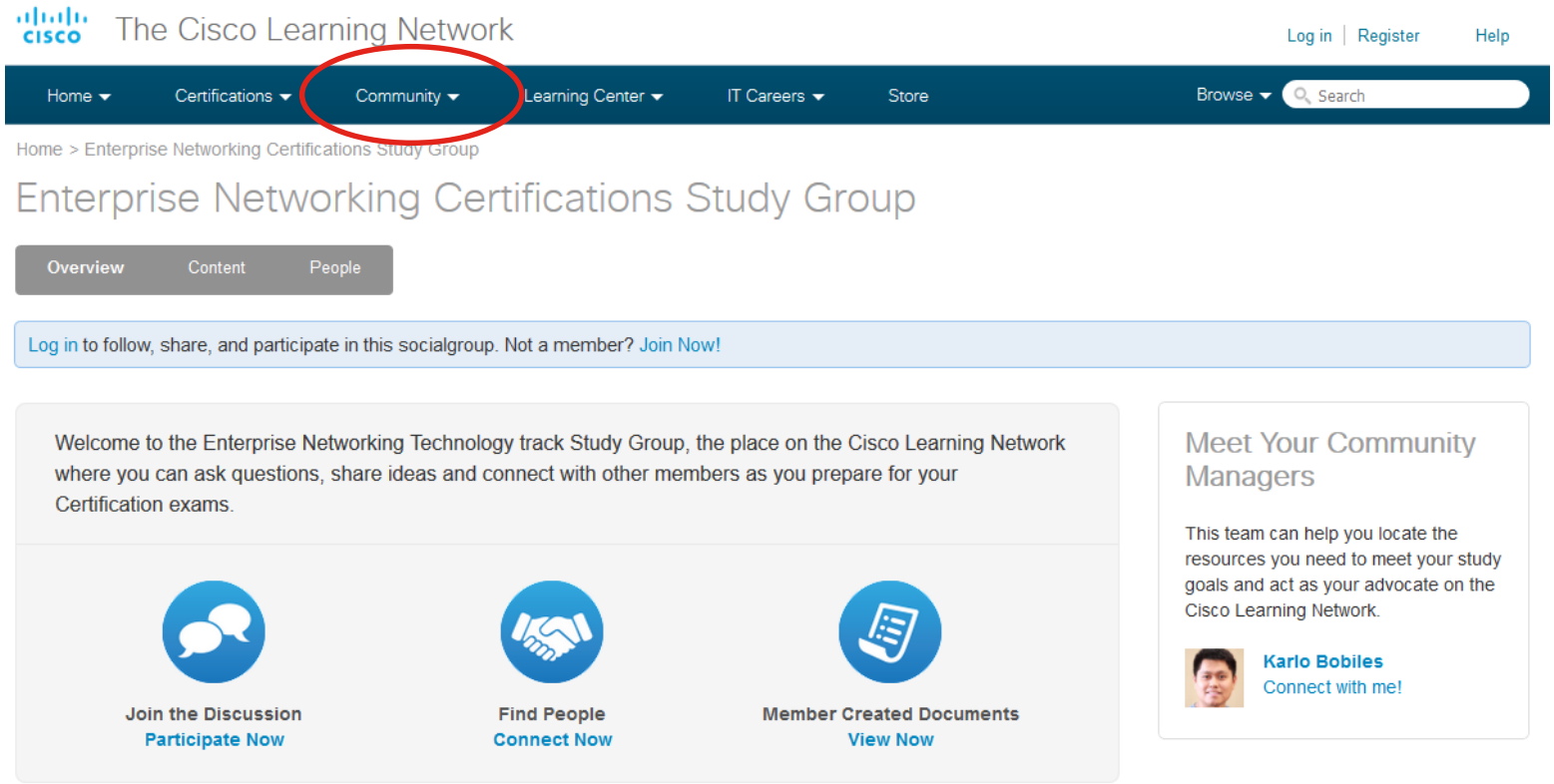
Network Device APIs

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



COURSE

Building Cisco SD-WAN Automation with Python

Intermediate 1h 30m 1 Lab

COURSE

Automating Cisco Meraki

Intermediate 1h 25m 1 Lab

COURSE

Implementing Automation in Enterprise Networks

Intermediate 1h 30m 1 Lab

COURSE

Building Cisco SD-WAN Automation with Ansible

Intermediate 2h 30m 3 Labs

COURSE

Automating Operations using Cisco DNA Center

Intermediate 2h 2 Labs

Cisco U.

COURSE

Building Cisco DNA Center Automation with Python

Intermediate 1h 15m 1 Lab

COURSE

Introducing Cisco SD-WAN Programmability

Intermediate 1h 15m 1 Lab

COURSE

Implementing Meraki Integration APIs

Intermediate 2h 5m 2 Labs

COURSE

Managing Configuration with Python and Ansible

Intermediate 1h 5m 1 Lab

COURSE

Implementing On-Box Programmability and Automation with Cisco IO…

Intermediate 1h 20m 2 Labs

COURSE

Day-Zero Provisioning with Cisco IOS XE Software

Intermediate 40m

COURSE

Automating APIs and Protocols

Intermediate 1h 20m 1 Lab

COURSE

Network Programmability Foundation

Intermediate 1h 50m 1 Lab

COURSE

Implementing Model-Driven Telemetry

Intermediate 1h 20m 1 Lab

LEARNING PATH

Implementing Automation for Cisco Enterprise Solutions | ENAUI

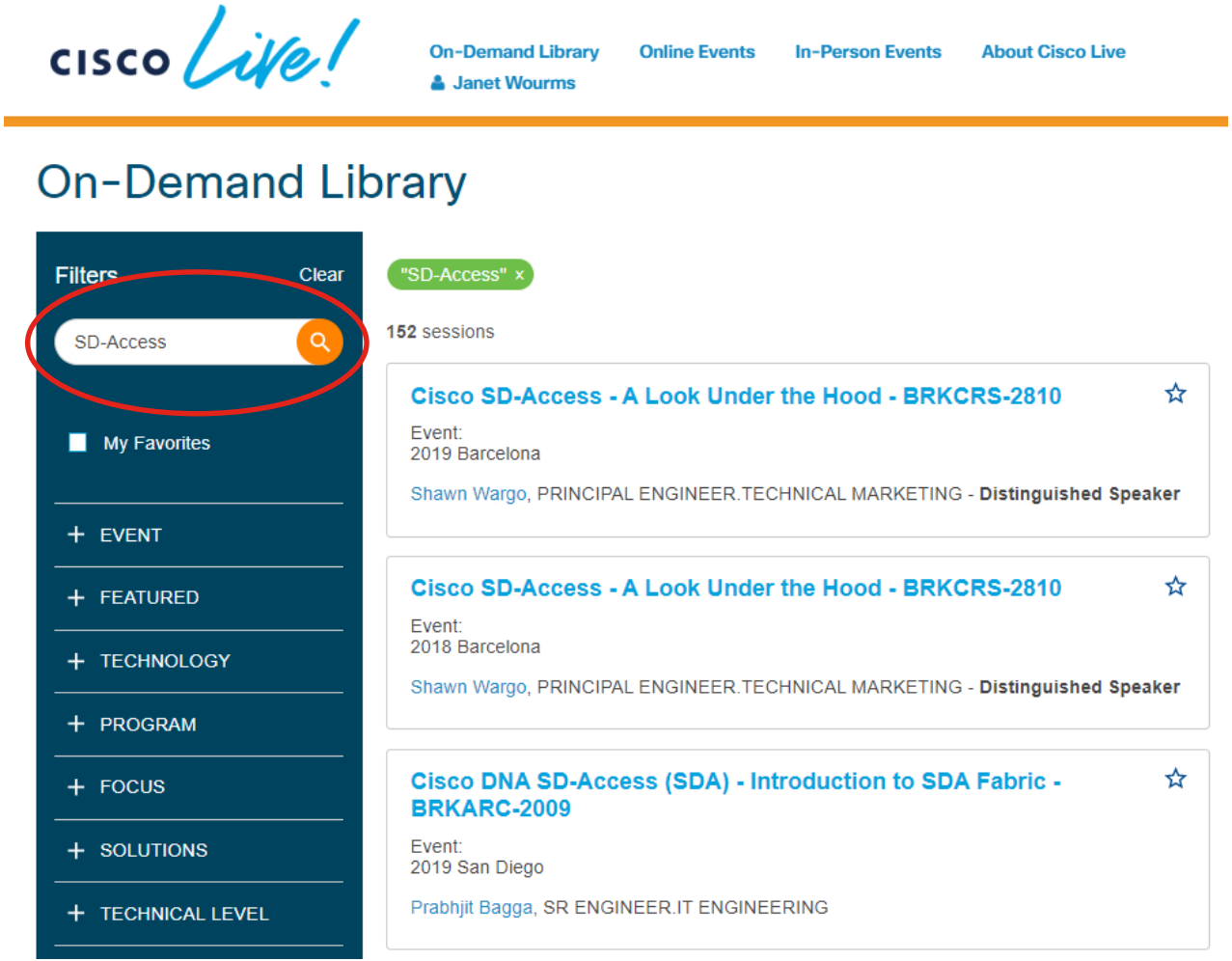
Learn to implement Cisco Enterprise automated solutions and prepare for the 300-435 ENAUTO certification exam.

Intermediate 21h 5m 24 Credits v1.2

Cisco courses from CLN

Cisco Live On-Demand Library

(www.ciscolive.com)



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

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**

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 2025.

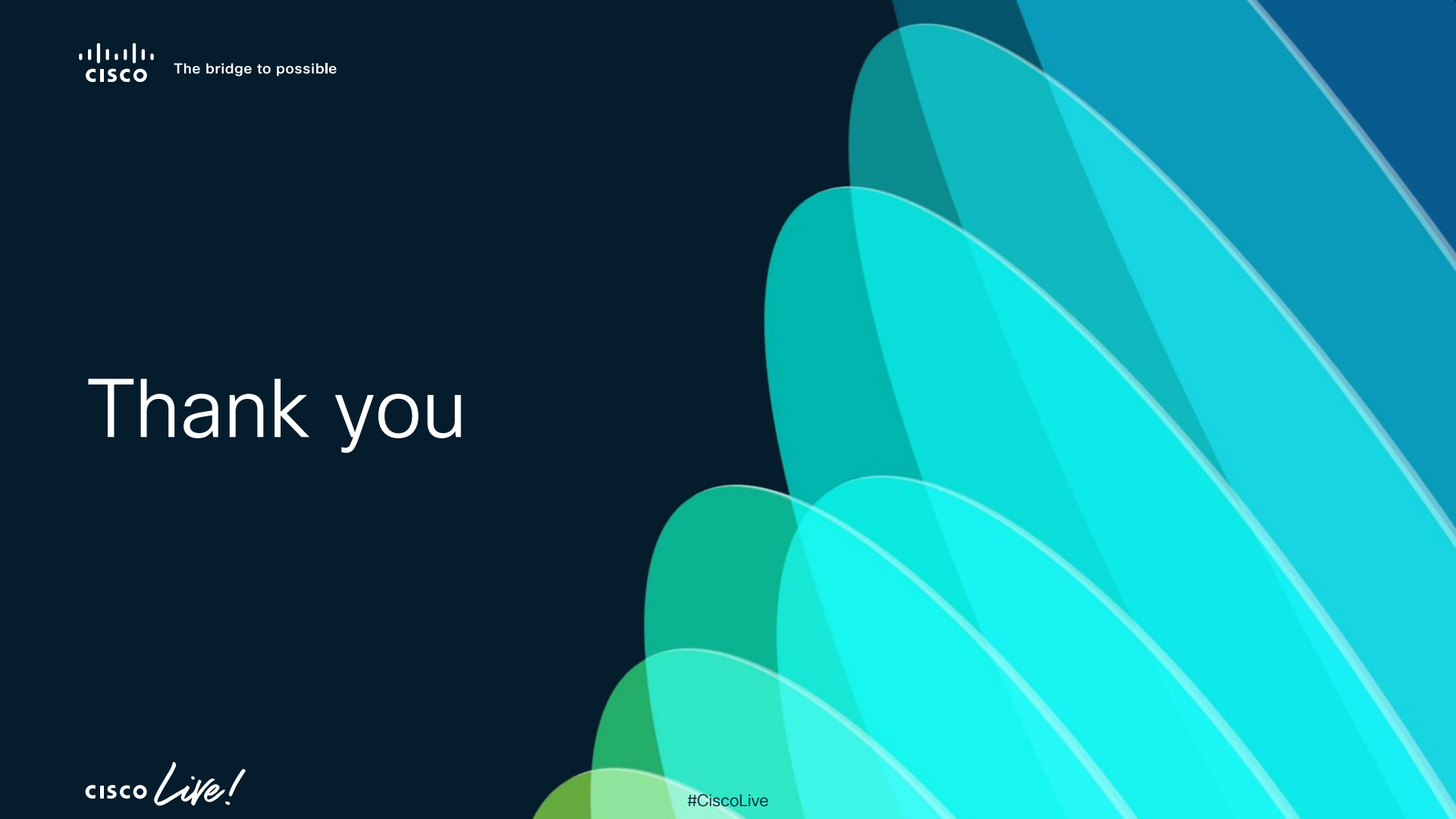
Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

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


The bridge to possible
Thank you
cisco Live!
#CiscoLive