
# Let's go

The bridge to possible

# Preparing for the DevNet Associate Certification

George Koukis, Leader Exam Program Manager, CCIEW#42079
Kareem Iskander, Lead Technical Advocate

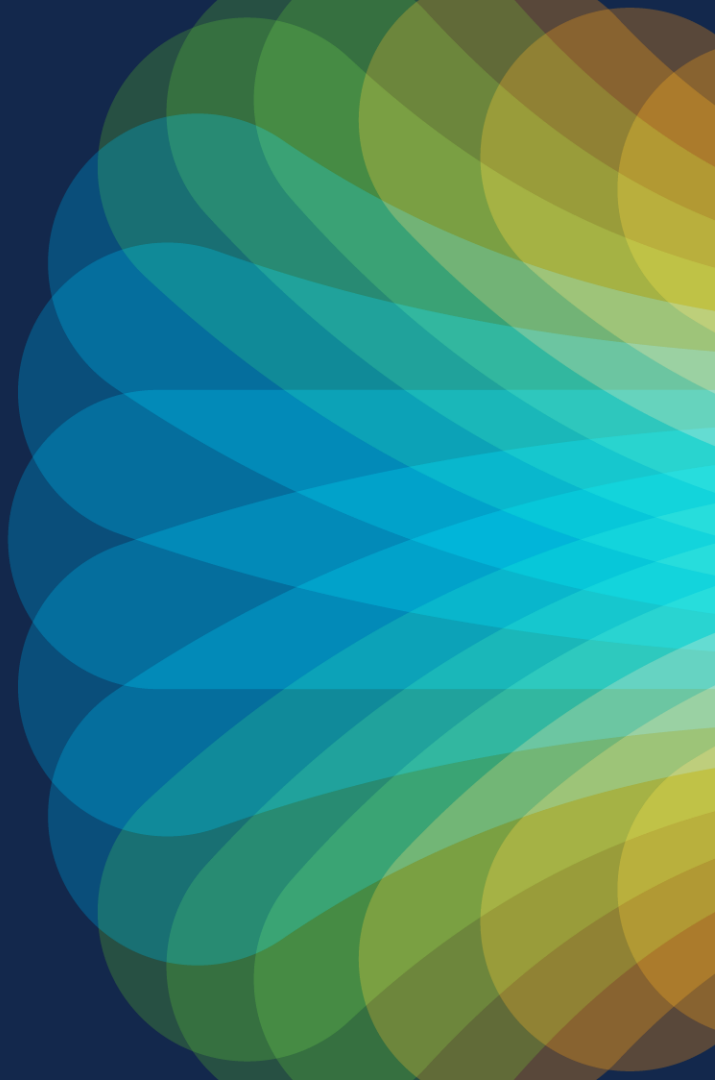
BRKCRT-1148

# Agenda

- Introduction to Cisco Certifications
- What to expect on the exam and how to prepare.
- Exam vs Real life


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Introduction to Cisco Certifications



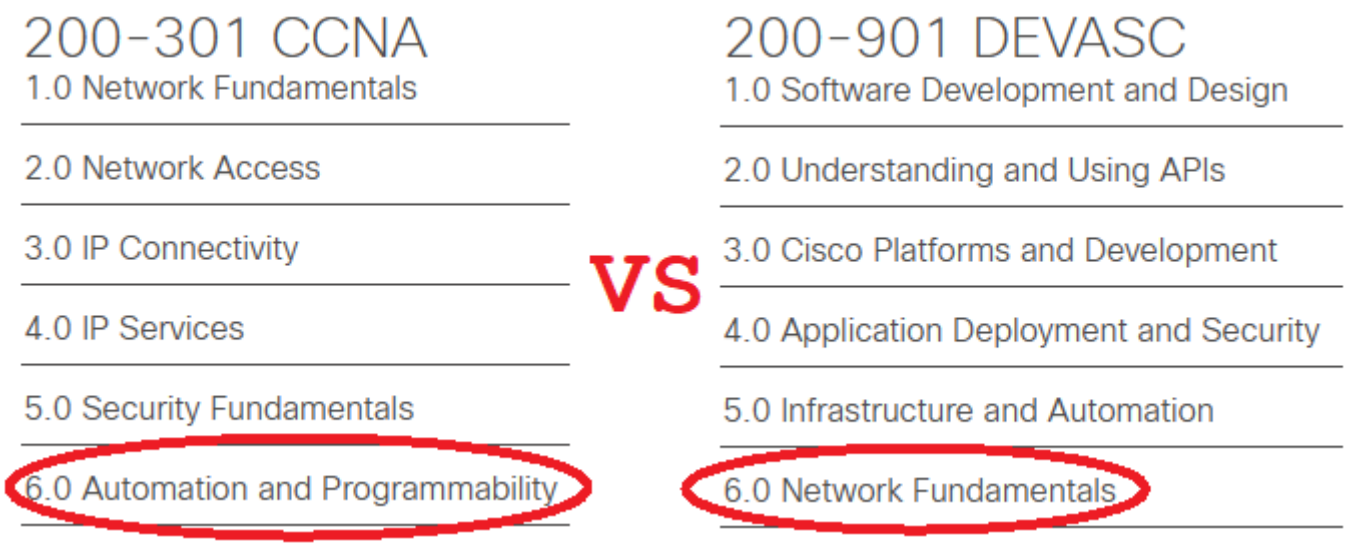
cisco Live!

# “DevNet professional, DEVASC and Automation skills Required”



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

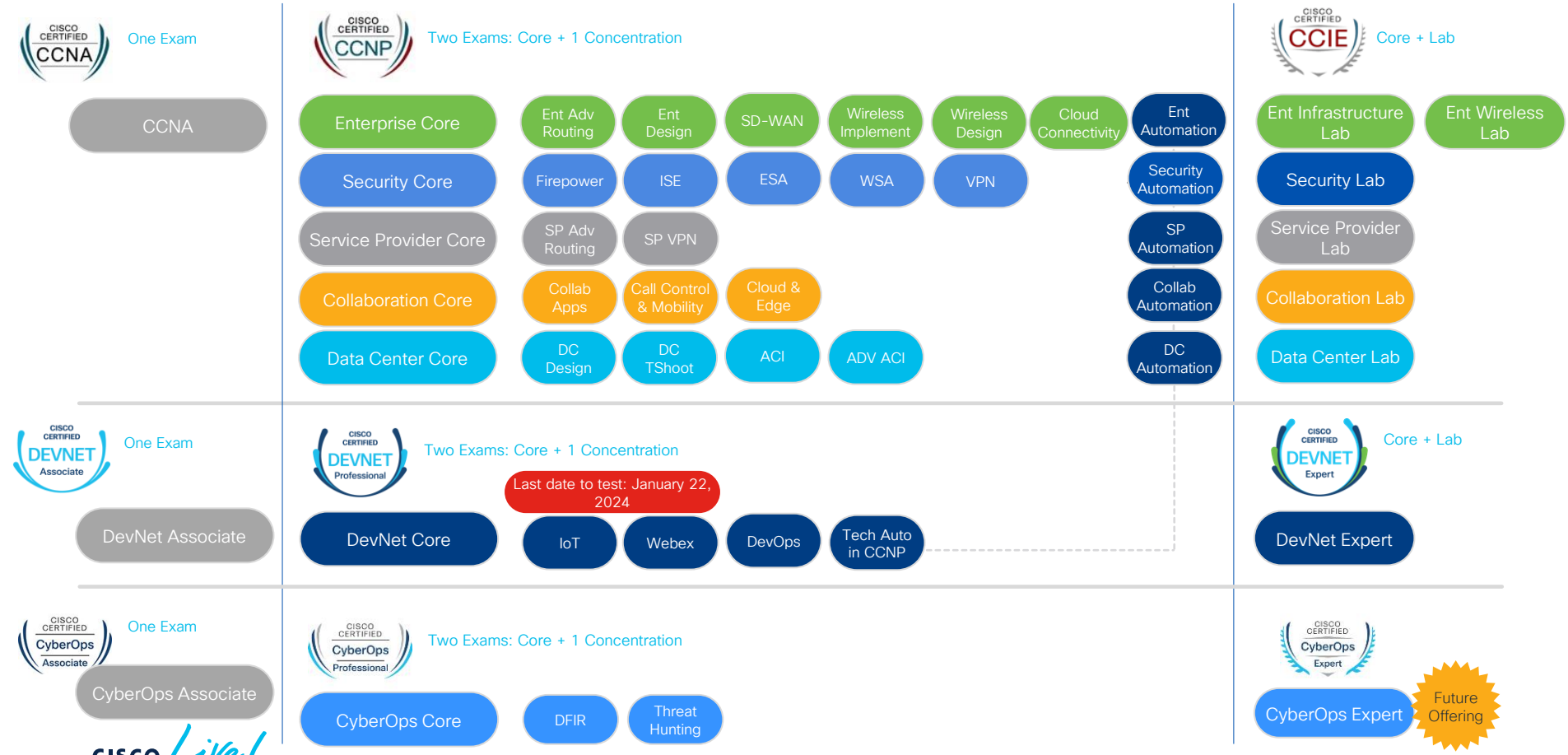
# Network skills or Developer skills





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

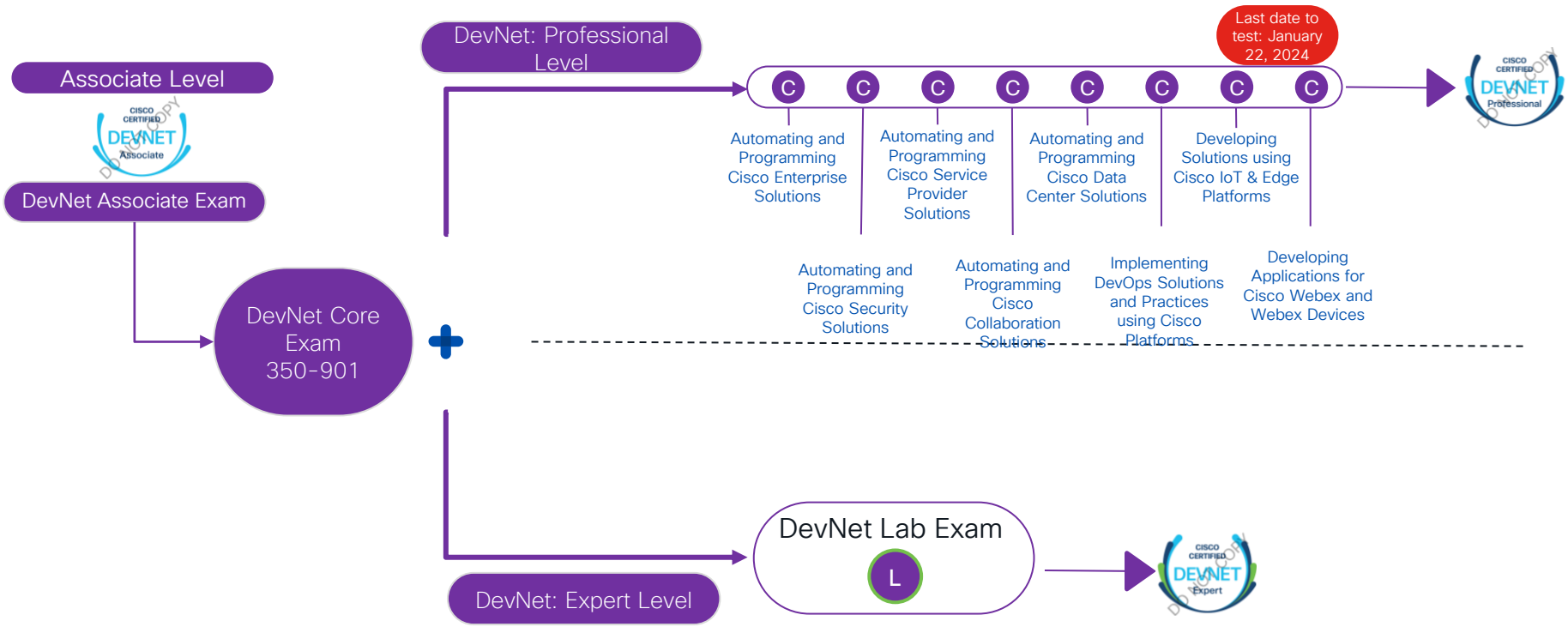
# Cisco Certifications






© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Cisco DevNet Certification Track





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

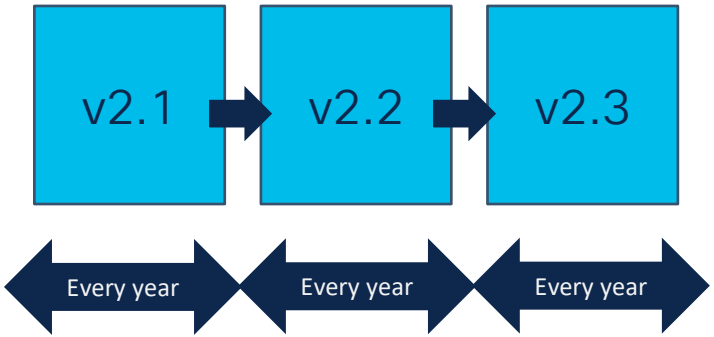
# Revision Framework

## Major Revision
(Traditional Revision Model)

Blueprint version number
v2.1 ➡ v3.0


- 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)


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# 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.



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# DEVASC v1.0 VS DEVASC v1.1

3.3 Describe the capabilities of Cisco compute management platforms and APIs (UCS Manager, ~~UCS Director~~, and Intersight)

3.4 Describe the capabilities of Cisco collaboration platforms and APIs (Webex ~~Teams~~, Webex devices, Cisco Unified Communication Manager including AXL and UDS interfaces, and Finesse)

3.5 Describe the capabilities of Cisco security platforms and APIs (XDR, Firepower, Umbrella, Secure endpoint~~AMP~~, ISE, and Secure Malware Analytics~~ThreatGrid~~)

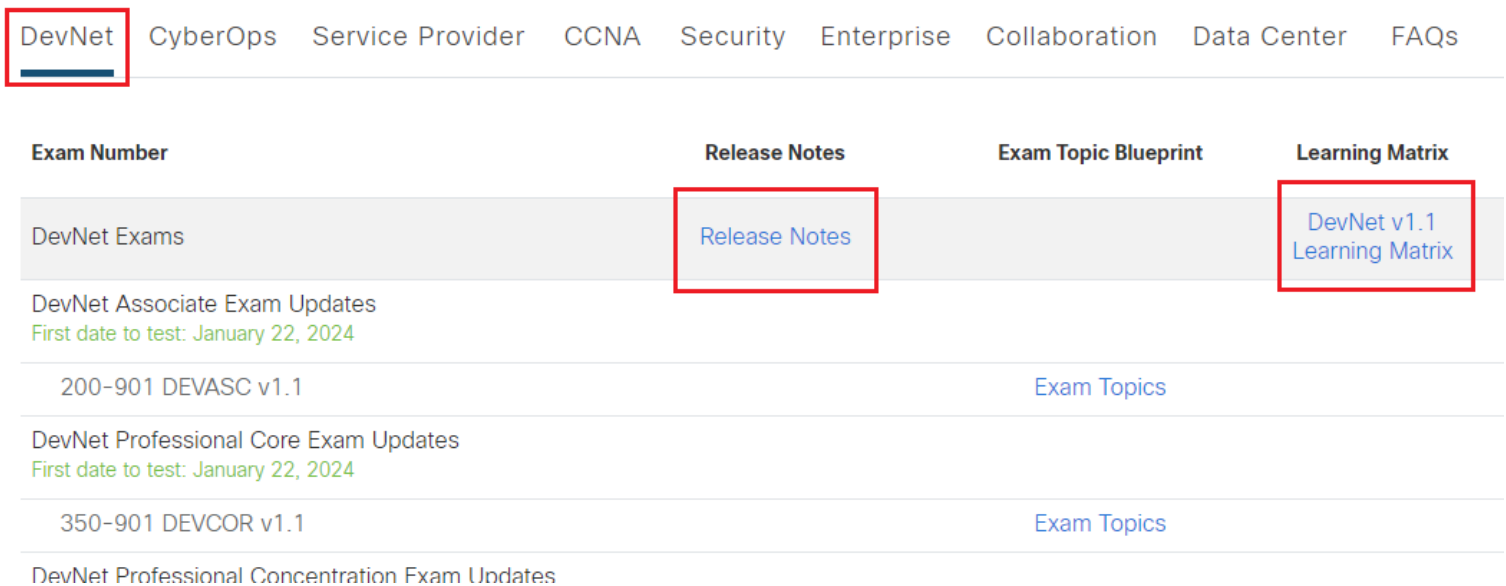
3.9.b Manage spaces, participants, and messages in Webex ~~Teams~~

5.3 Describe the use and roles of network simulation and test tools (such as Cisco Modeling Labs~~VIRL~~ and pyATS)

5.6 Describe the capabilities of automation tools such as Ansible, Terraform~~Chef~~, and Cisco NSO


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# https://learningnetwork.cisco.com/s/cisco-certification-roadmaps



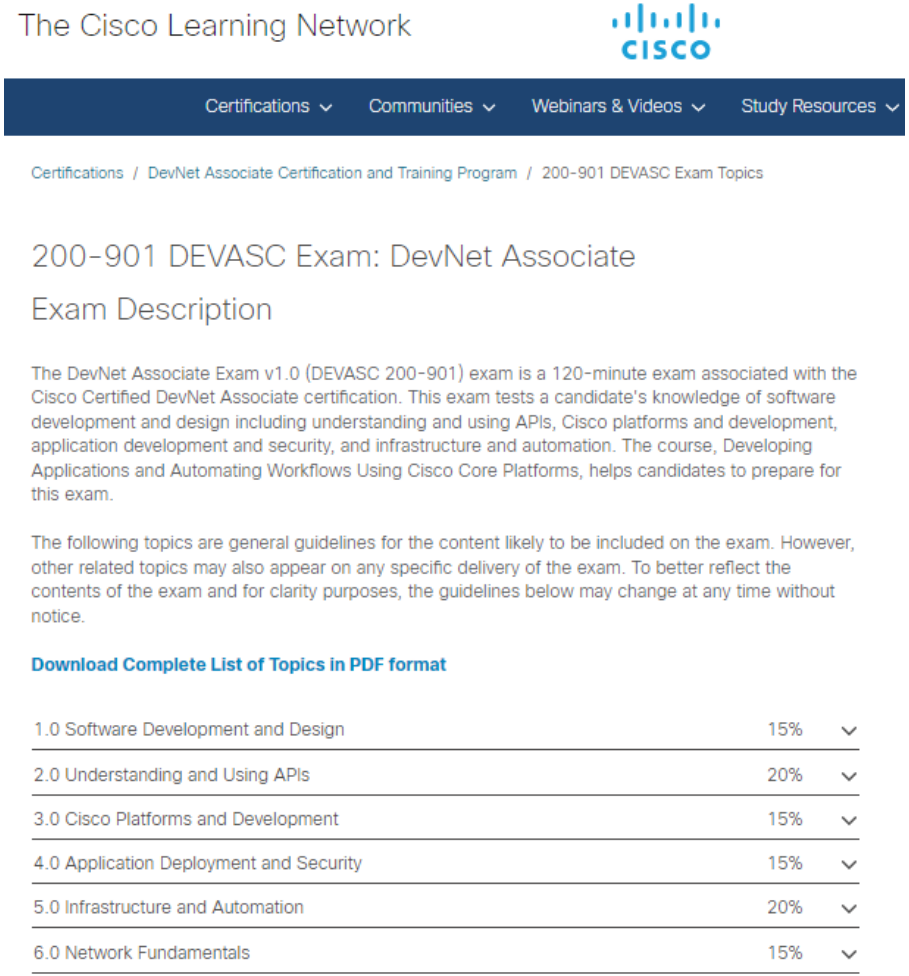


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# What to expect on the exam

# Exam Blueprint

https://learningnetwork.cisco.com

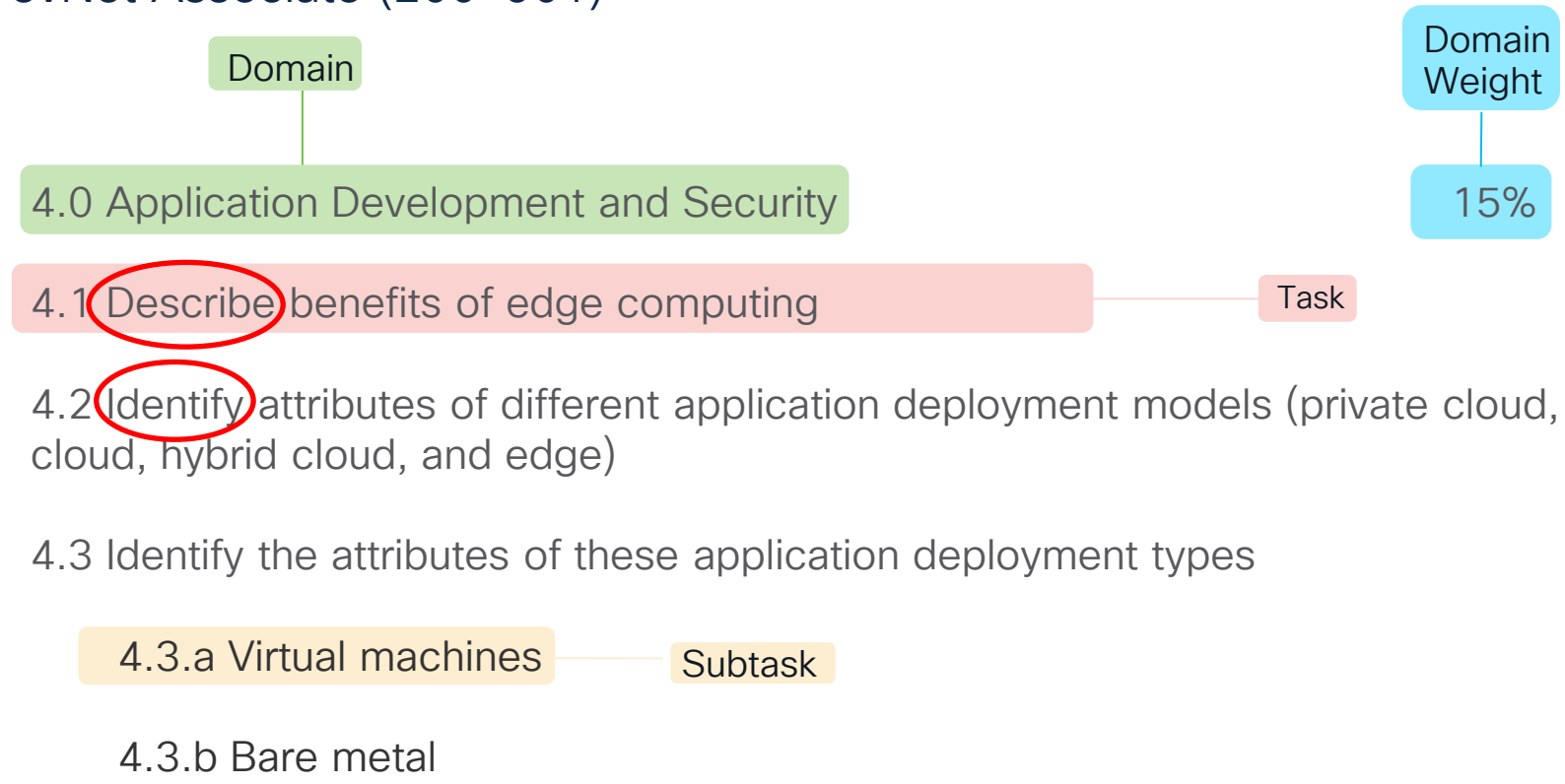




© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Interpret the Blueprint:

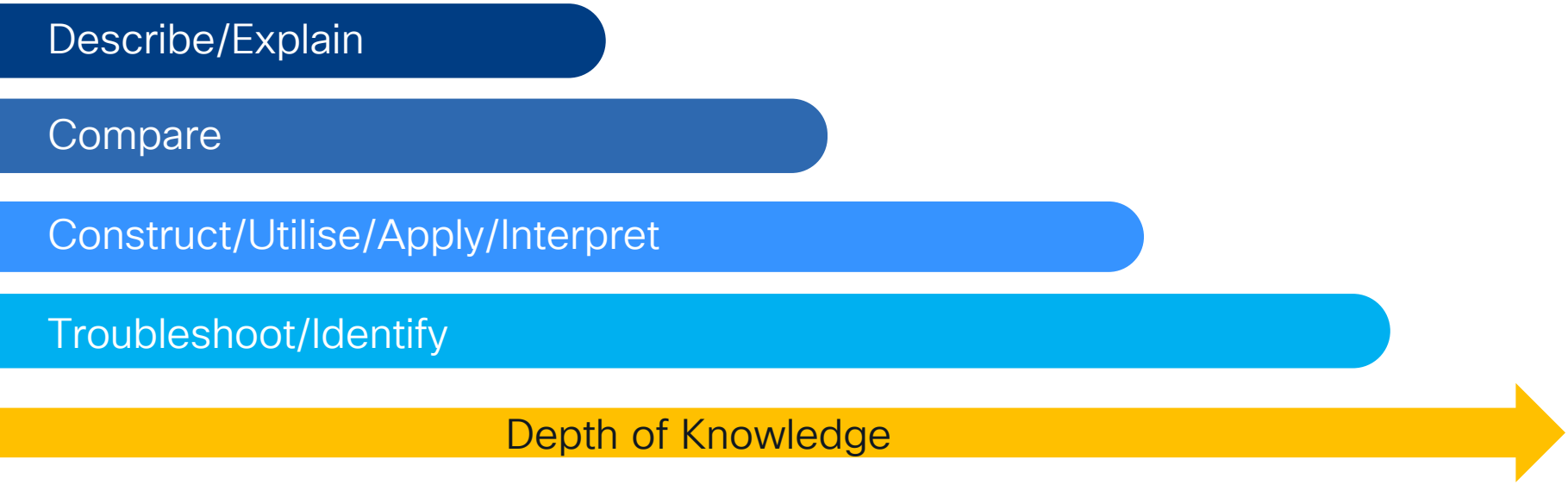
## DevNet Associate (200-901)





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

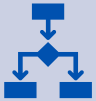
# Blueprint Verbs





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Types of questions



Multiple choice

Drag and drop

Fill in the blanks


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# 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


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# 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


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Compare question

**Example:** Compare common 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 | **Rest APIs** |
| stateless | stateless |
| uses XML | uses XML |
| uses YAML | **RPC APIs** |
| | stateful |
| | uses YAML |


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Compare question

**Example:** Compare common 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 | **Rest APIs** |
| stateless | stateless |
| uses XML | uses XML |
| uses YAML | **RPC APIs** |
| | stateful |
| | uses YAML |


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# How to prepare

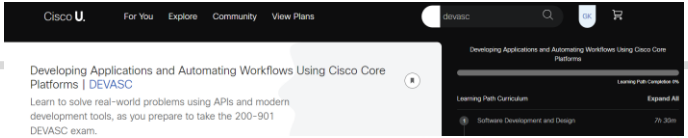
Describe/Explain

Compare

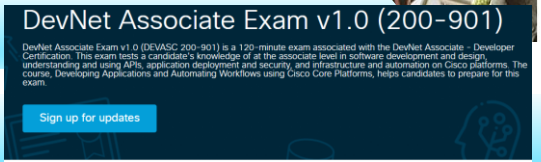
Construct/Utilize/Apply/Interpret

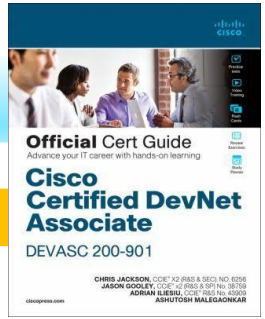
Troubleshoot/Identify

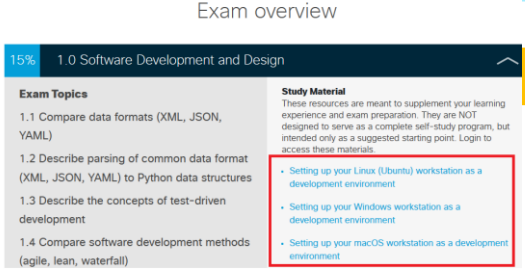
Depth of Knowl









© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# 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('tear down class')

[ ] test_two(self):
        log.info('two')

    @classmethod
    def setUpClass(self):
        log.info('set up Class')
```

class

unittest

log

def

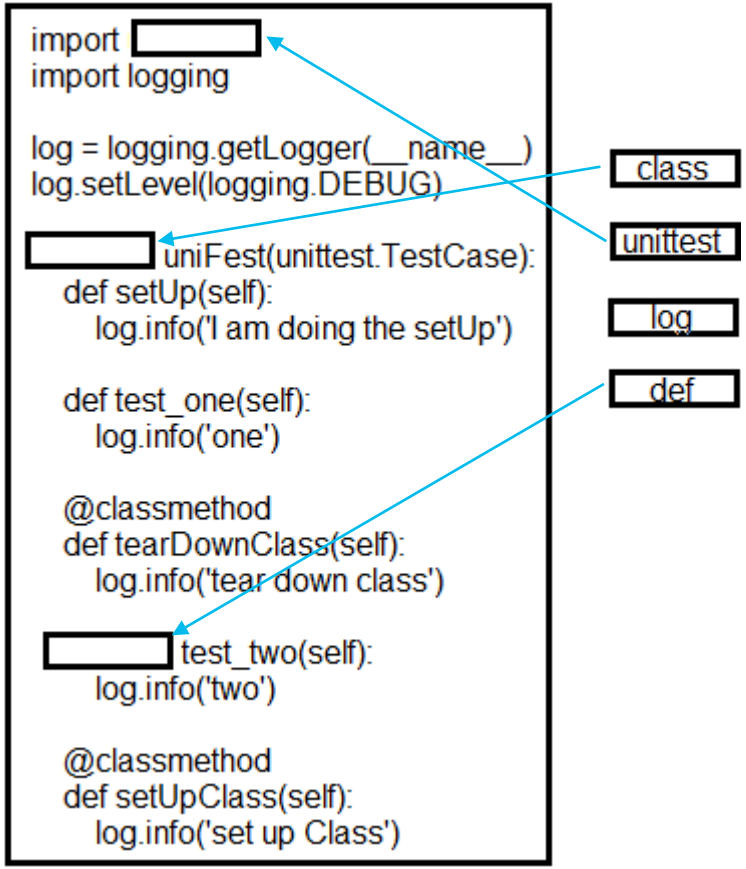

© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# 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.





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# How to prepare

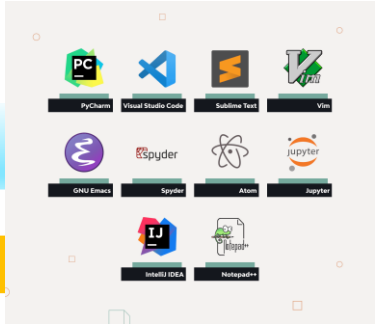
Describe/Explain

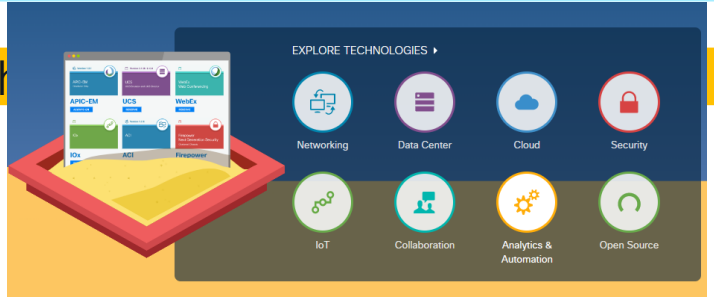
Compare

Construct/Utilize/Apply/Interpret

Troubleshoot/Identify

Depth



© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Troubleshoot question

Something is broken and we need to fix it.

**Example:** Troubleshoot a problem given the HTTP response code, request and API documentation

```
import requests
import json
url = "https://192.168.1.10:443/restconf/data/ietf-interfaces:interfaces"
payload = json.dumps({
  "ietf-interfaces:interface": {"name": "Loopback200","description":
  "External LOOPBACK200 for VTI Tunnels","type": "iana-if-type:softwareLoopback",
    "enabled": True,"ietf-ip:ipv4": {"address": [{"ip": "84.16.101.150",
          "netmask": "255.255.255.255"}]}}})
headers = {
  'Authorization': 'Basic e3t1c2VybmFtZX19Ont7cGFzc3dvcmR9fQ==',
  'Accept': 'application/yang-data+json',
  'Content-Type': 'application/yang-data+json'}
response = requests.request("POST", url, headers=headers, data=payload)
print(r.status_code)
print(response.text)
```

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 applied to the customer side, it returned 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 must 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.


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Troubleshoot question

Something is broken and we need to fix it.

**Example:** Troubleshoot a problem given the HTTP response code, request and API documentation

```
import requests
import json
url = "https://192.168.1.10:443/restconf/data/ietf-interfaces:interfaces"
payload = json.dumps({
  "ietf-interfaces:interface": {"name": "Loopback200","description":
  "External LOOPBACK200 for VTI Tunnels","type": "iana-if-type:softwareLoopback",
    "enabled": True,"ietf-ip:ipv4": {"address": [{"ip": "84.16.101.150",
          "netmask": "255.255.255.255"}]}}})
headers = {
  'Authorization': 'Basic e3t1c2VybmFtZX19Ont7cGFzc3dvcmR9fQ==',
  'Accept': 'application/yang-data+json',
  'Content-Type': 'application/yang-data+json'}
response = requests.request("POST", url, headers=headers, data=payload)
print(r.status_code)
print(response.text)
```

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 applied to the customer side, it returned 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 must 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.


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# How to prepare

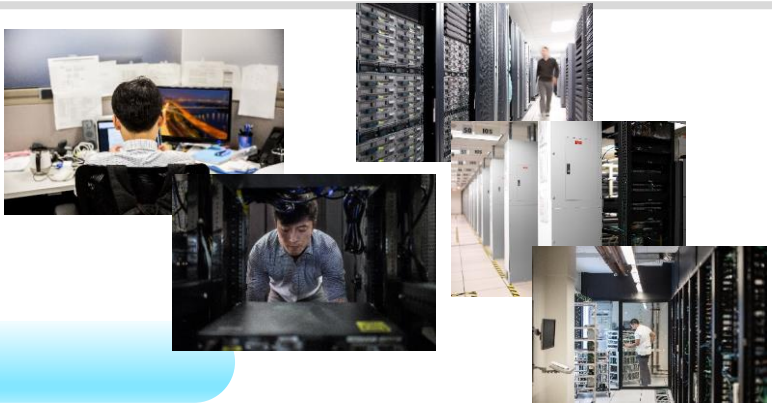
Describe/Explain

Compare

Construct/Utilize/Apply/Interpret

Troubleshoot/Identify

Depth of Knowledge




© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Exam vs Real life

# What will be covered

- 1.8 Utilize common version control operations with Git
- 2.9 Construct a Python script that calls a REST API using the requests library
- 3.9 Construct code to perform a specific operation based on a set of requirements and given API reference documentation such as these:
  - 3.9.a Obtain a list of network devices by using Meraki, Cisco DNA Center, ACI, Cisco SD-WAN, or NSO
- 4.11 Utilize Bash commands (file management, directory navigation, and environmental variables)

Sample Code:
- Scan QR code
or
- https://github.com/qsnyder/brkcrt-2080


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# What will be covered

- 1.8 Utilize common version control operations with Git
- 2.9 Construct a Python script that calls a REST API using the requests library
- 3.9 Construct code to perform a specific operation based on a set of requirements and given API reference documentation such as these:
  - 3.9.a Obtain a list of network devices by using Meraki, Cisco DNA Center, ACI, Cisco SD-WAN, or NSO
- 4.11 Utilize Bash commands (file management, directory navigation, and environmental variables)


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Utilize question

## 1.8 Utilize common version control operations with Git

## 1.8.a Clone

```
git [          ] ssh://user@example.com/path/to/application.git
```

Refer to the exhibit. An engineer is working on a new feature for an existing application. The application is already live and the development must take place locally on the engineer's machine. Which Git command must the engineer use to create a copy of the application on the local machine?

A. clone

B. copy

C. pull

D. add


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Utilize question

## 1.8 Utilize common version control operations with Git

## 1.8.a Clone

```
git [          ] ssh://user@example.com/path/to/application.git
```

Refer to the exhibit. An engineer is working on a new feature for an existing application. The application is already live and the development must take place locally on the engineer's machine. Which Git command must the engineer use to create a copy of the application on the local machine?

A. clone

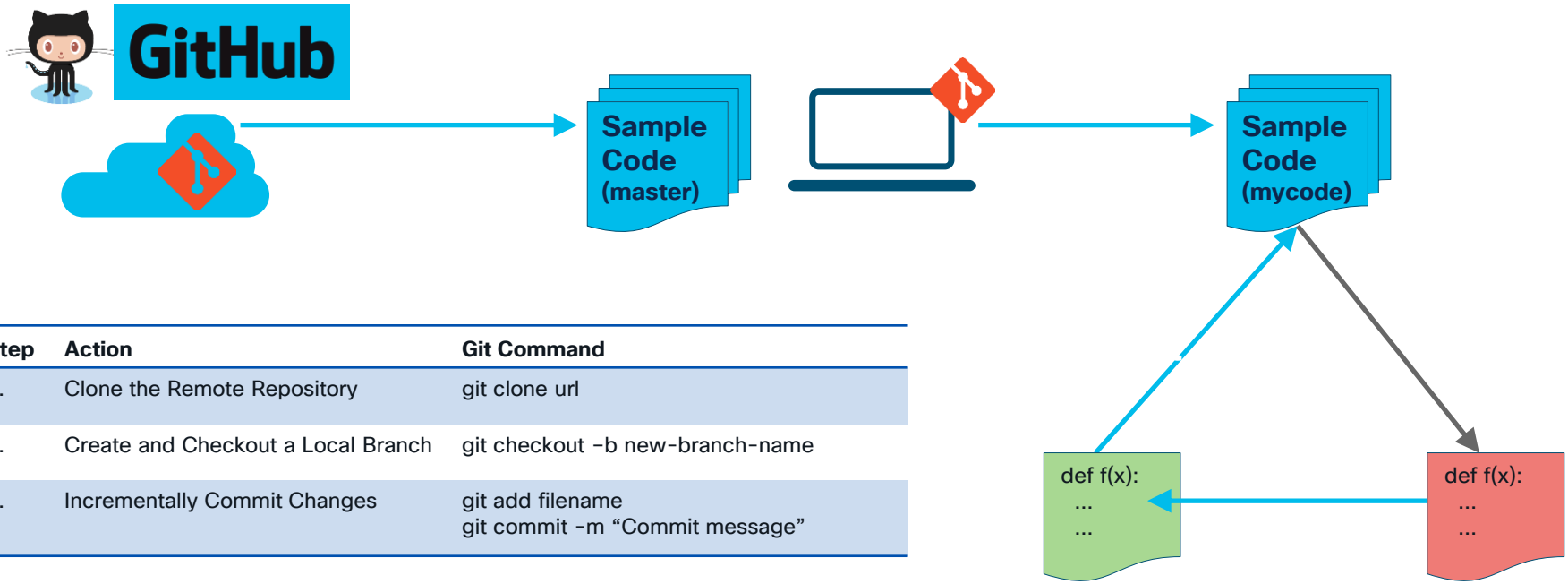
B. copy

C. pull

D. add


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# DevNet Sample-Code Workflow




| Step | Action | Git Command |
|---|---|---|
| 1. | Clone the Remote Repository | git clone url |
| 2. | Create and Checkout a Local Branch | git checkout -b new-branch-name |
| 3. | Incrementally Commit Changes | git add filename<br>git commit -m “Commit message” |


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Git Hands-on Walkthrough

# What will be covered

- 1.8 Utilize common version control operations with Git
- **2.9 Construct a Python script that calls a REST API using the requests library**
- 3.9 Construct code to perform a specific operation based on a set of requirements and given API reference documentation such as these:
  - 3.9.a Obtain a list of network devices by using Meraki, Cisco DNA Center, ACI, Cisco SD-WAN, or NSO
- 4.11 Utilize Bash commands (file management, directory navigation, and environmental variables)


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Construct question

## 2.9 Construct a Python script that calls a REST API using the requests library

Drag and drop the code from the bottom onto the box where the code is missing to retrieve a list of organizations configured in the Meraki organization. Not all options are used.

```
import [          ]
MERAKI_API_KEY = '6bec40cf957de430a6f1f2baa056b99a4fac9ea0'
base_url = "https://api.meraki.com/api/v1"
endpoint = "/organizations"
header = {"X-Cisco-Meraki-API-Key": MERAKI_API_KEY}
orgs = requests.[          ](url=base_url+endpoint, headers=header)
[          ] = json.loads(orgs.text)
pprint(orgs)
```

orgs | get | patch | requests

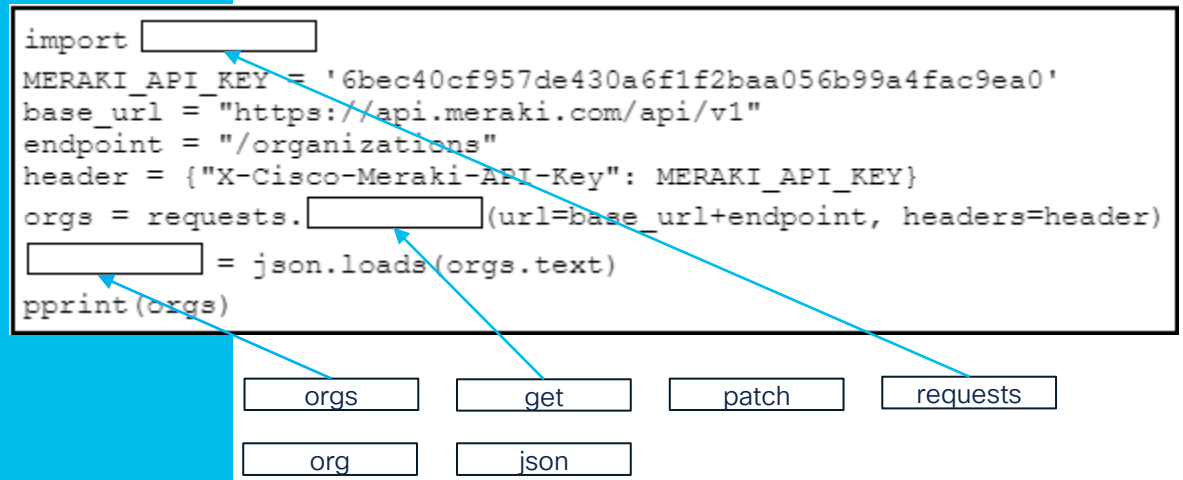
org | json


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Construct question

## 2.9 Construct a Python script that calls a REST API using the requests library

Drag and drop the code from the bottom onto the box where the code is missing to retrieve a list of organizations configured in the Meraki organization. Not all options are used.





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Demo Overview – Python & REST APIs

```python
import requests
import json
from pprint import pprint
# DevNet Sandbox Information
from requests import Response
'''
Username: devnetmeraki@cisco.com
Password: Adm!n123!
You can also use this API key for the Dashboard API: 6bec40cf957de430a6f1f2baa056b99a4fac9ea0
'''
MERAKI_API_KEY = '6bec40cf957de430a6f1f2baa056b99a4fac9ea0'
# MERAKI BASE URL
base_url = "https://api.meraki.com/api/v1"
# MERAKI Endpoint
endpoint = "/organizations"
# requests header information
header = {"X-Cisco-Meraki-API-Key": MERAKI_API_KEY}
# Make request
orgs: Response = requests.get(url=base_url+endpoint, headers=header)
orgs = json.loads(orgs.text)
pprint(orgs)


```

*Let's Test it!*

- **Requests** is an elegant and simple HTTP **library** for **Python**, built for human beings
- Define requests building blocks
- Make API call to service
- Capture returned data and doing something with it


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# What will be covered

- 1.8 Utilize common version control operations with Git
- 2.9 Construct a Python script that calls a REST API using the requests library
- 3.9 Construct code to perform a specific operation based on a set of requirements and given API reference documentation such as these:
  - 3.9.a Obtain a list of network devices by using Meraki, Cisco DNA Center, ACI, Cisco SD-WAN, or NSO
- 4.11 Utilize Bash commands (file management, directory navigation, and environmental variables)


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Construct question

3.9.a Obtain a list of network devices by using Meraki, Cisco DNA Centre, ACI, Cisco SD-WAN, or NSO SDWAN

Drag and drop the code from the bottom onto the box where the code is missing to retrieve a list of devices in the Meraki organization. Not all options are used.

```
import requests
base_url = "https://api.meraki.com/api/v1"
[          ] = {"X-Cisco-Meraki-API-Key": MERAKI_API_KEY}
def get_device_list():
    for org in orgs:
        endpoint = "/organizations/{}/devices".format(org['id'])
        print([          ])
        resp = requests.get(url=base_url + endpoint, headers=header)
        [          ] = json.loads(resp.text)
        pprint(device)
if __name__ == "__main__":
    get_device_list()
```

header | device | auth | endpoint

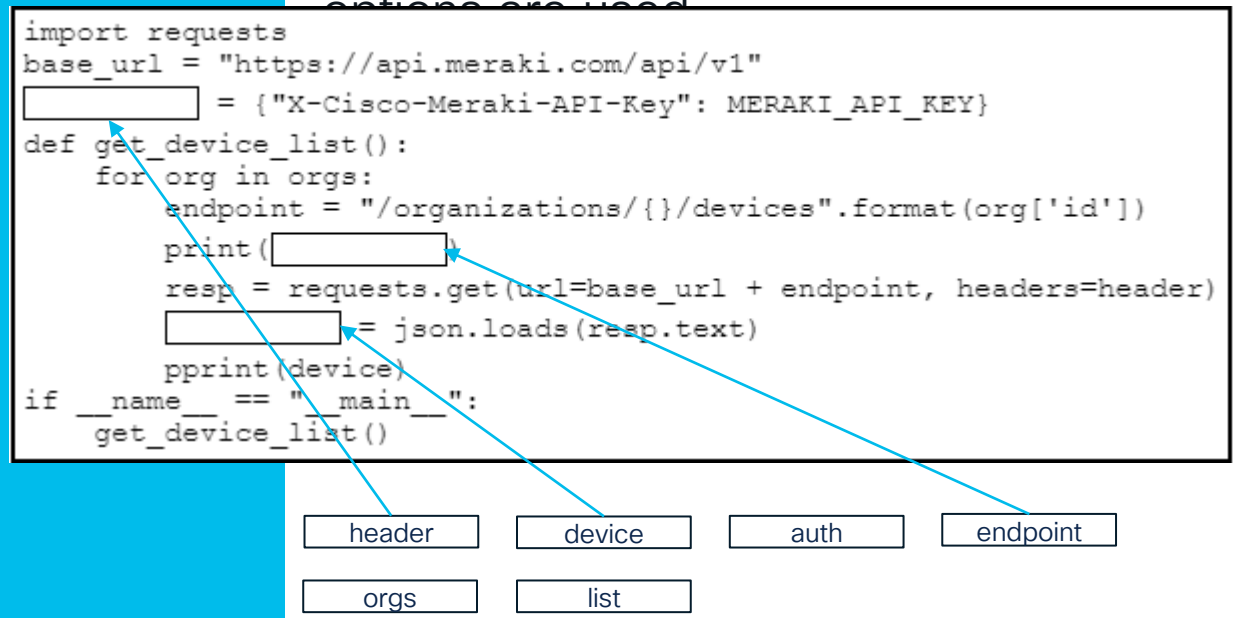
orgs | list


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Construct question

3.9.a Obtain a list of network devices by using Meraki, Cisco DNA Centre, ACI, Cisco SD-WAN, or NSO SDWAN

Drag and drop the code from the bottom onto the box where the code is missing to retrieve a list of devices in the Meraki organization. Not all options are used.





© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Demo Overview – Meraki Network Devices

```python
import requests
import json
from pprint import pprint

# DevNet Sandbox Information
'''
Username: devnetmeraki@cisco.com
Password: Adm!n123!
You can also use this API key for the Dashboard API: 6bec40cf957de430a6f1f2baa056b99a4fac9ea0
'''

MERAKI_API_KEY = '6bec40cf957de430a6f1f2baa056b99a4fac9ea0'
# MERAKI BASE URL
base_url = "https://api.meraki.com/api/v1"
# requests header information
header = {"X-Cisco-Meraki-API-Key": MERAKI_API_KEY}

def get_org_list():
    endpoint = "/organizations"
    # Make request
    resp = requests.get(url=base_url+endpoint, headers=header)
    orgs = json.loads(resp.text)
    print(orgs)
    return orgs

def get_device_list():
    for org in orgs:
        # MERAKI Endpoint
        endpoint = "/organizations/{}/devices".format(org['id'])
        print(endpoint)
        resp = requests.get(url=base_url + endpoint, headers=header)
        device = json.loads(resp.text)
        pprint(device)

if __name__ == "__main__":
    orgs = get_org_list()
    get_device_list()
```

*Let's Test it!*

- Meraki APIs and Python Requests
- Authenticate
- Meraki Hierarchical order
  - Orgs contain Networks
  - Networks contain Devices
- Get list of orgs
- loop through each network in orgs and extract devices


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# What will be covered

- 1.8 Utilize common version control operations with Git
- 2.9 Construct a Python script that calls a REST API using the requests library
- 3.9 Construct code to perform a specific operation based on a set of requirements and given API reference documentation such as these:
  - 3.9.a Obtain a list of network devices by using Meraki, Cisco DNA Center, ACI, Cisco SD-WAN, or NSO
- 4.11 Utilize Bash commands (file management, directory navigation, and environmental variables)


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Utilize question

## 4.11 Utilize Bash commands

```
$virtualenv -p python3 <desired-path>
$source <desired-path>/bin/
```

Refer to the exhibit. An engineer needs to complete the creation and enable a Python virtual environment. Which code needs to be placed on the box where the code is missing?

A. activate

B. enable

C. allow

D. power


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Utilize question

## 4.11 Utilize Bash commands

```
$virtualenv -p python3 <desired-path>
$source <desired-path>/bin/
```

Refer to the exhibit. An engineer needs to complete the creation and enable a Python virtual environment. Which code needs to be placed on the box where the code is missing?

A. activate

B. enable

C. allow

D. power


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Virtual Environment

- You can code in Python without using virtual environments.
- The minute you update modules or Python itself, you run the risk of breaking your apps
- A virtual environment allows you to lock in the components and modules you use for your app into a single “package.”


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Enabling Python VirtualEnv

- Directory Structure
- Usually associated with a Project
- An isolated environment for installing and working with Python Pacges

```
$ python3 -m venv myvenv

$ source myvenv/bin/activate
```

**source** *environment-name*/**bin/activate**

- ✓ The activation script will modify your prompt.
- ✓ Inside a virtual environment, your interpreter will always be `**python**`.


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

cisco Live!

# Did you know?

You can have a
one-on-one session with
a technical expert!

Visit Meet the Expert in The HUB
to meet, greet, whiteboard & gain
insights about your unique questions
with the best of the best.

## Meet the Expert Opening Hours:

| | |
|---|---|
| **Tuesday** | **3:00pm – 7:00pm** |
| **Wednesday** | **11:15am – 7:00pm** |
| **Thursday** | **9:30am – 4:00pm** |
| **Friday** | **10:30am – 1:30pm** |

# Session Surveys

**We would love to know your feedback on this session!**

- Complete a minimum of four session surveys and the overall event surveys to claim a Cisco Live T-Shirt

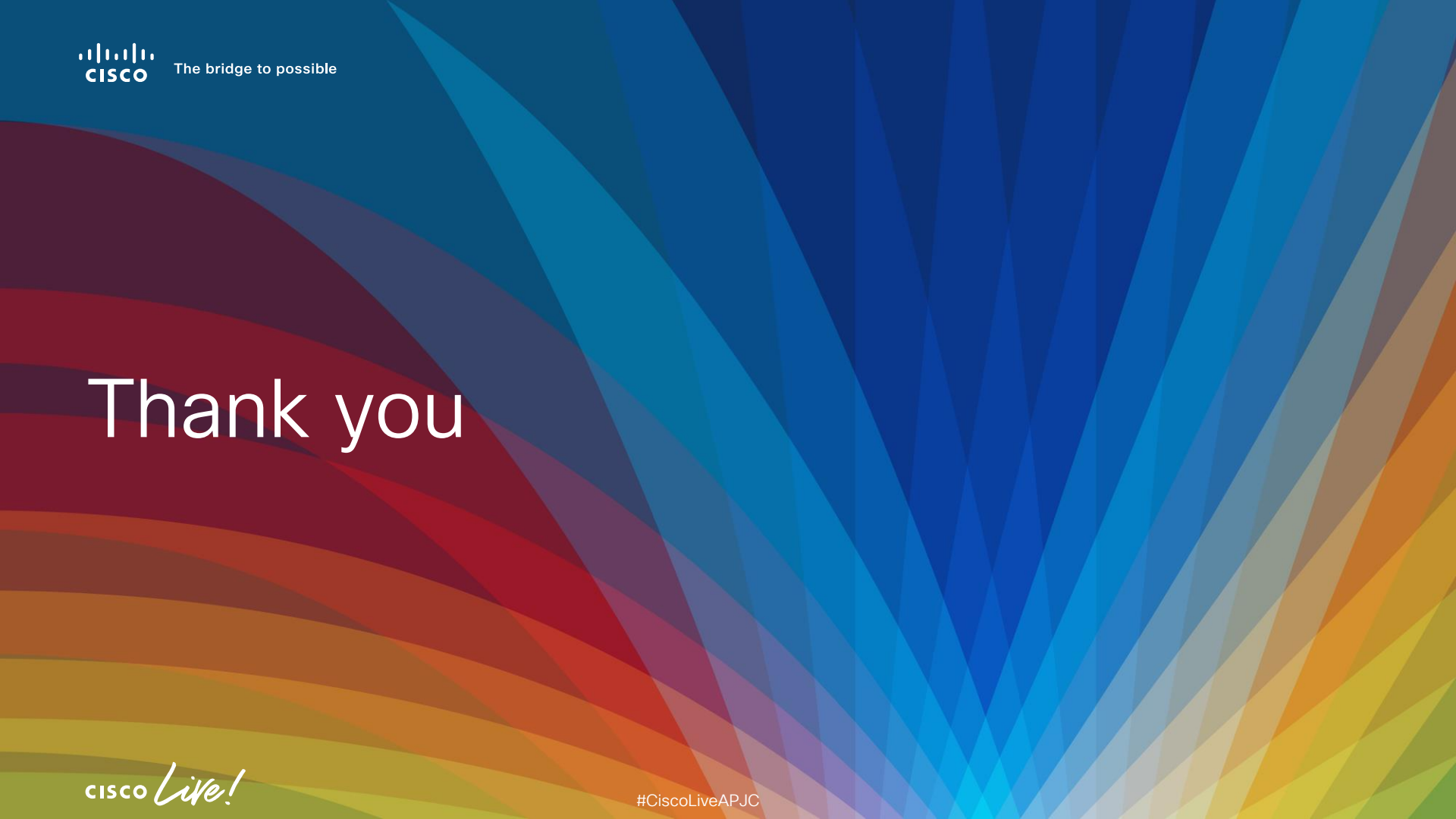


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public

# Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Expert 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


© 2023 Cisco and/or its affiliates. All rights reserved. Cisco Public


cisco
The bridge to possible
Thank you
cisco Live!
#CiscoLiveAPJC


cisco Live!
Let's go
#CiscoLiveAPJC