

Efficient Cisco Popular 300-535 Exams Are Leading Materials & The Best 300-535: Automating and Programming Cisco Collaboration Solutions



Now, let us show you why our 300-535 exam questions are absolutely your good option. First of all, in accordance to the fast-paced changes of bank market, we follow the trend and provide the latest version of 300-535 study materials to make sure you learn more knowledge. Secondly, since our 300-535 training quiz appeared on the market, seldom do we have the cases of customer information disclosure. We really do a great job in this career!

Cisco 300-535 certification exam is designed to provide an in-depth understanding of the automation and programming concepts, tools, and techniques used in Cisco Collaboration solutions. 300-535 Exam is ideal for IT professionals who want to develop their skills in automating and programming Cisco Collaboration solutions and become experts in this field.

>> Popular 300-535 Exams <<

300-535 Prep Guide | 300-535 Valid Test Question

With our customer-oriented 300-535 actual question, you can be one of the former exam candidates with passing rate up to 98 to 100 percent. You will pay just a small amount of money on our 300-535 exam guide but harvest colossal success with potential bright future. And we have confidence that your future aims will come along with this successful exam as the beginning. So choosing 300-535 actual question is choosing success.

Cisco Automating and Programming Cisco Collaboration Solutions Sample Questions (Q38-Q43):

NEW QUESTION # 38

Fill in the blank to complete the statement about NETCONF and Python libraries.

_____ is a Python library that facilitates client-side scripting and deploying changes to the network using the NETCONF protocol.

Answer:

Explanation:

ncclient

Reference:

<https://pypi.org/project/ncclient/>

https://www.ic.unicamp.br/~edmundo/proceedings/html/fullPapers/88577_1.pdf

NEW QUESTION # 39

```
def configure_ip_address(interface, ip, length):
    url = BASE_URL + "/data/ietf-interfaces:interfaces/interface={i}".format(
        i = interface
    )
    data = OrderedDict(
        [
            (
                "ietf-interfaces:interface",
                OrderedDict(
                    [
                        ("name", interface),
                        ("type", "iana-if-type:ethernetCsmacd"),
                        (
                            "ietf-ip:ipv6",
                            OrderedDict(
                                [
                                    (
                                        "address",
                                        [OrderedDict([("ip", ip), ("prefix-length", length)]),
                                    ],
                                ),
                            ),
                        ),
                    ],
                ),
            ),
        ],
    )

    response = requests.put(
        url, auth=(USERNAME, PASSWORD), headers=HEADERS, verify=False, json=data
    )
    print(response.status_code)

configure_ip_address("GigabitEthernet2", "2001:db8:636c:6179:2063:7572:7469:7300", "64")
```

Refer to the exhibit. What is the effect of the script on the device?

- A. It merges the new configuration with the existing configuration on the device using RESTCONF.
- B. It compares the configuration to the device. If it matches, the device sends back an HTTP 204 status code.
- C. It replaces the entire configuration for GigabitEthernet2 on the device using RESTCONF.
- D. All interfaces except GigabitEthernet2 are reset to their default configurations.

Answer: A

Explanation:

Section: 2.0 Automation APIs and Protocols

NEW QUESTION # 40

Refer to the exhibit. Which lines of code need to be placed on the snippet where the code is missing in the script to connect to a Cisco IOS XR device via NETCONF over port 10000 and download the running-config in XML format?

```
#!/usr/bin/env python3
```

```
from ncclient import manager
```

```
host = "10.10.10.240"
```

```
PORT = 10000
```

```
USER = "admin"
```

```
PASSWORD = "Cisco12345"
```

```
with open("%s.xml" % host, 'w') as f:  
    f.write(c)
```

- A.

```
with ncclient.connect(host,port=PORT,  
    username=USER,  
    password=PASSWORD,  
    hostkey_verify=True,  
    device_params={'name':'iosxr'}) as m:  
    c = m.get_config(source='CMD'.data xml
```

```
with manager.connect(host,port=PORT,  
    username=USER,  
    password=PASSWORD,  
    hostkey_verify=False,  
    device_params={'name':'iosxr'}) as m:  
    c = m.get_config(source='running'.data xml
```

- B.
- C.

```
with manager.connect(host,port=10000,  
    username=admin,  
    password=Cisco12345,  
    hostkey_verify=False,  
    device_params={'name':'iosxr'}) as m:  
    c = m.get_config(source='running'.data xml
```

```
with ncclient.connect(host,port=10000,  
    username="admin",  
    password="Cisco12345",  
    hostkey_verify=True,  
    device_params={'name':'iosxr'}) as m:  
    c = m.get_config(source='CMD'.data xml
```

- D.

Answer: B

Explanation:

This option uses the variables host, PORT, USER, and PASSWORD as defined earlier in the script, and retrieves the running configuration in XML format from a Cisco IOS XR device using NETCONF.

NEW QUESTION # 41

Drag and Drop Question

Drag and drop the code snippets from the bottom onto the blanks in the code to run the displayed ansible playbook, which configures an interface on a Cisco device. Not all options are used.

```
admin@dev:/$ cat mgmt_iface.yml
---
- name: Provision Management Interface
  hosts: csrl
  gather_facts: false

  tasks:
    - name: Enable Management Interface
      ios_interfaces:
        config:
          - name: "{{ iface_name }}"
            description: Management access
            mtu: 1500
            enabled: true
```

```
admin@dev:/$
```

```
-i
```

```
-u admin -e
```

mgmt_iface.yml

-I

inventory.yml

ansible-playbook

("iface_name":"GigabitEthernet4")

ansible

inventory.yml,

iface_name=GigabitEthernet4

Answer:

Explanation:

```
admin@dev:/$ cat mgmt_iface.yml
```

```
---
- name: Provision Management Interface
  hosts: csrl
  gather_facts: false

  tasks:
    - name: Enable Management Interface
      ios_interfaces:
        config:
          - name: "{{ iface_name }}"
            description: Management access
            mtu: 1500
            enabled: true
```

```
admin@dev:/$ ansible-playbook
```

```
-i inventory.yml
```

```
-u admin -e iface_name=GigabitEthernet4
```

```
mgmt_iface.yml
```

```
-I
```

```
ansible
```

```
inventory.yml,
```

```
("iface_name": "GigabitEthernet4")
```

Explanation:

```
ansible-playbook \
-i inventory.yml \
-u admin -e iface_name=GigabitEthernet4 \
mgmt_iface.yml
```

ansible-playbook runs the playbook.

-i inventory.yml specifies the inventory file.

-e iface_name=GigabitEthernet4 passes the variable.

NEW QUESTION # 42

An engineer needs to configure network devices in an automated way. Which two ways are used to create structured data using YANG to provide REST-like APIs to enable programmability access? (Choose two.)

- A. E.XML
- B. JSON-RPC
- C. YAML
- D. JSON
- E. GPB

Answer: A,D

Explanation:

Reference: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/166/b_166_programmability_cg/restconf_prog_int.pdf

NEW QUESTION # 43

.....

For candidates who are going to buy the 300-535 training materials online, they have the concern of the safety of the website. Our 300-535 training materials will offer you a clean and safe online shopping environment, since we have professional technicians to examine the website and products at times. In addition, 300-535 Training Materials have 98.75% pass rate, and you can pass the exam. We also pass guarantee and money back guarantee if you fail to pass the exam.

300-535 Prep Guide: <https://www.fast2test.com/300-535-premium-file.html>

- [illegible]