300-435Q&As

Automating and Programming Cisco Enterprise Solutions (ENAUTO)

Pass Cisco 300-435 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/300-435.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Cisco Official Exam Center

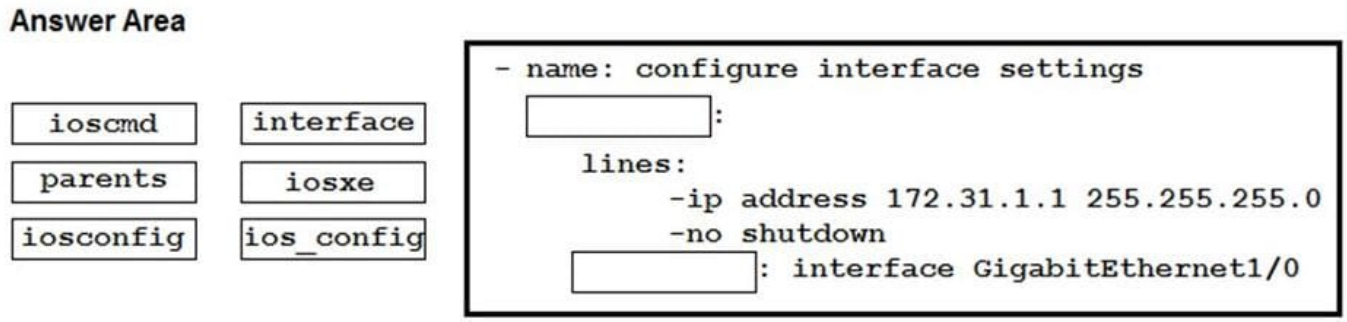
- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

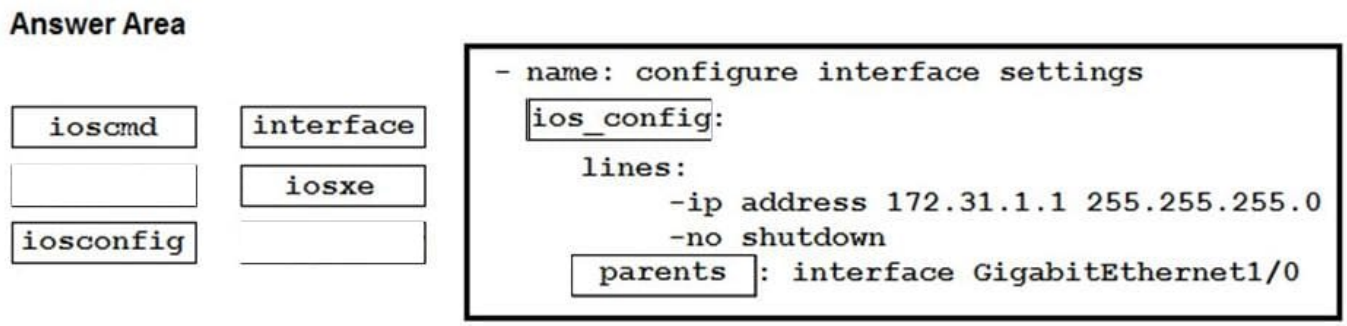
DRAG DROP

Drag and drop the commands to the Ansible playbook that applies configuration to an interface on a Cisco IOS XE device. Not all options are used.

Select and Place:



Correct Answer:



Reference: http://imxing.cn/?p=464

QUESTION 2

```
import requests
import sys

requests.package.urllib3.disable_warnings()

HOST = '10.1.2.3'
PORT = 9443
USER = 'user'
PASS = 'password'

def main():
    url = "https://{h}:{p}/restconf/data/Cisco-IOS-XE-native:native/
hostname".format(h=HOST, p=PORT)

    headers = {'Content-Type': 'application/ [          ] ',
               'Accept': 'application/ [          ] '}
    response = requests.get(url, auth=(USER,PASS),
                            headers=headers, verify=False)
    print(response.text)

if __name__ == '__main__':
    sys.exit(main())
```

Refer to the exhibit. An engineer creates a Python script using RESTCONF to display hostname information. The code must be completed so that it can be tested. Which string completes the highlighted areas in the exhibit?

A. yang-data+json

B. yang +json

C. yang.data+json

D. json

Correct Answer: A

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

QUESTION 3

Which two features are characteristics of software-defined networks when compared to traditional infrastructure? (Choose two.)

A. configured box-by-box

B. changed manually

C. use overlay networks

D. designed to change

E. require software development experience to manage

Correct Answer: CD

Reference: https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/VMDC/SDN/SDN.html

QUESTION 4


```
return_val=
{
  "alertId": "6434517967656725l6",
  "alertType": "appliances went down",
  "deviceMac": "e0:55:3d:6c:c1:7a",
  "deviceName: "MX65 c1:7a",
  "deviceSerial": "Q2QN-58EA-XXXX",
  "deviceUrl": "https://n143.meraki.com/Branch-1/n/.../manage/nodes/new_wired_status",
  "networkId": "L_1234567890",
  "networkName": "Branch 1",
  "networkUrl": "https://n143.meraki.com/Branch-1/n/.../manage/nodes/wired_status",
  "occuredAt": "2018-11-10T18:45:20.000000Z",
  "organizationId": "1234567",
  "organizationName": "Meraki Demo",
  "organizationUrl": "https://n143.meraki.com/o/.../manage/organization/overview",
  "sentAt: "2018-11-10T18:50:30.479982Z",
  "SharedSecret": "asdf1234",
  "version": "0.1"
}
```


Refer to the exhibit. The task is to create a Python script to display an alert message when a Meraki MX Security Appliance goes down. The exhibit shows sample data that is received. Which Python snippet displays the device name and the time at which the switch went down?

A.
```
with return_val:
    print("The Switch: "+deviceName+ ",
    went down at: "+occurredAt)
```

B.
```
print("The Switch: "+return_val.deviceName+ ", \
went down at: "+return_val.occurredAt)
```

C.
```
print("The Switch: "+return_val['deviceName']+ ", \
went down at: "+return_val['occurredAt']")
```

D.
```
with items as return_val:
    print("The Switch: "+items.deviceName+ ",
    went down at: "+items.occurredAt)
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: B

QUESTION 5

Which Python snippet receives a Meraki webhook request?

A.
```
@app.route('/mynet/webhook', methods=['PUT'])
@app.accept_body(WebhookSchema)
def receive_webhook(**kwargs):
        send_sms_alert(kwargs['alertType'])
```

B.
```
@app.route('/mynet/webhook', methods=['GET'])
@app.accept_body(WebhookSchema)
def receive_webhook(**kwargs):
        send_sms_alert(kwargs['alertType'])
```

C.
```
@app.route('/mynet/webhook', methods=['PATCH'])
@app.accept_body(WebhookSchema)
def receive_webhook(**kwargs):
        send_sms_alert(kwargs['alertType'])
```

D.
```
@app.route('/mynet/webhook', methods=['POST'])
@app.accept_body(WebhookSchema)
def receive_webhook(**kwargs):
        send_sms_alert(kwargs['alertType'])
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: D

Reference: https://github.com/CiscoDevNet/dnav3-code/blob/master/intro-meraki/meraki-07-webhooks/webhookreceiver.py