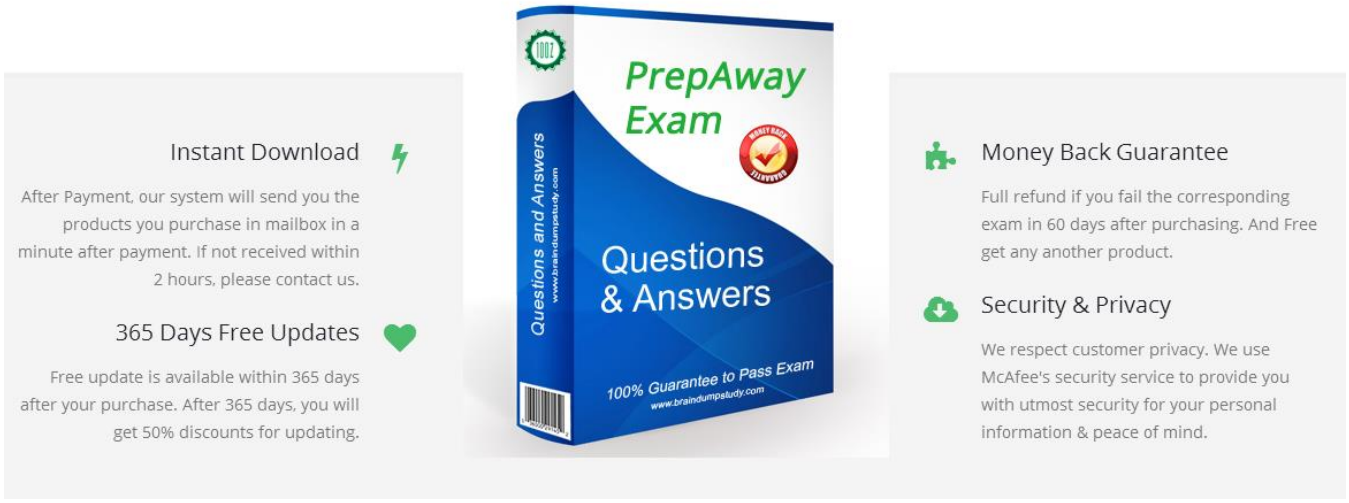
PrepAwayExam

Try **Online Engine** before you buy



http://www.prepawayexam.com/

High-efficient Exam Materials are the best high pass-rate Exam Dumps

Exam : **300-435J**

Title : Automating and Programming Cisco Enterprise Solutions (300-435日本語版)

Vendor : Cisco

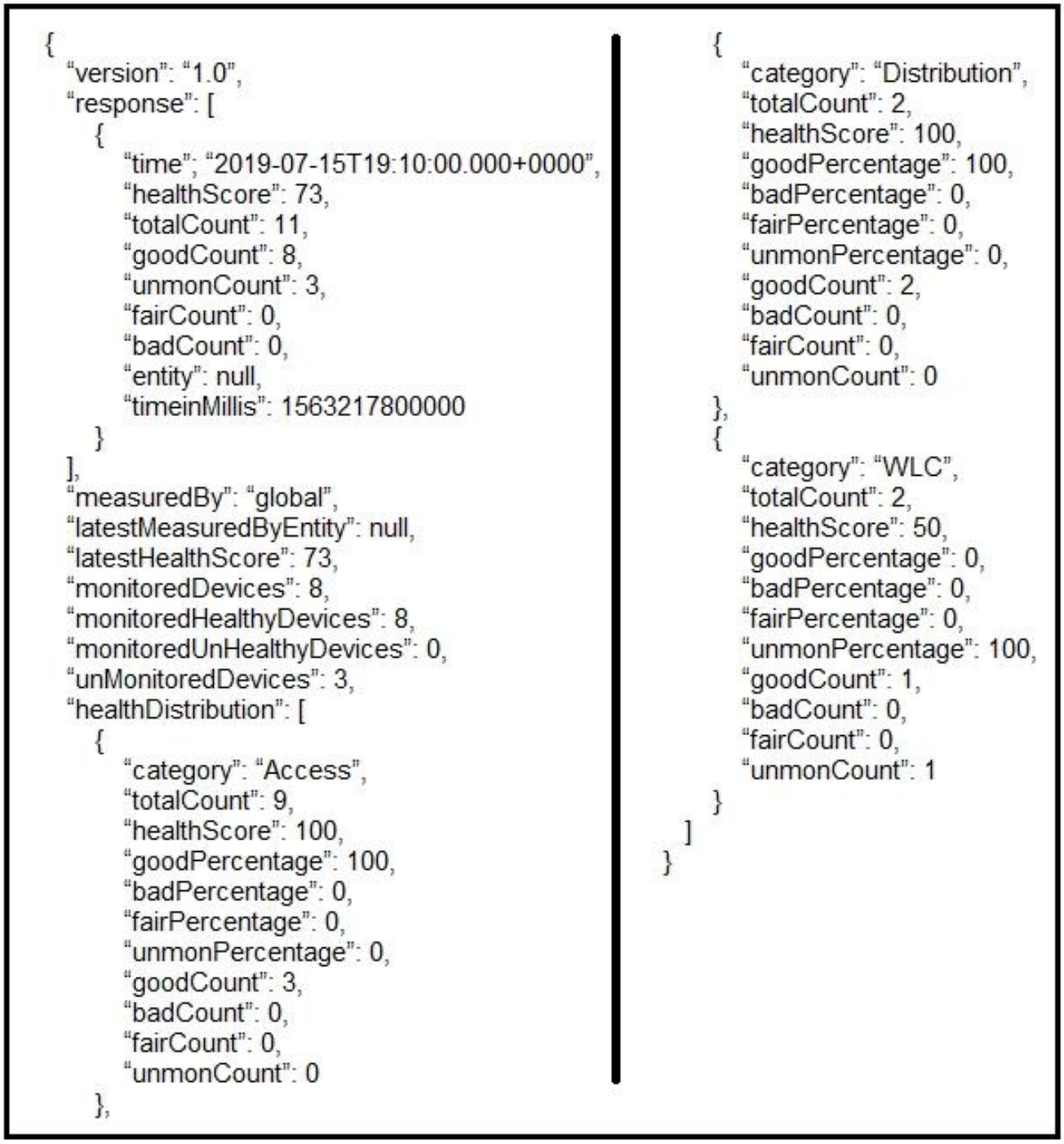
Version : DEMO

QUESTION NO: 1
添付資料を参照してください。Cisco IOS XEデバイスで静的テレメトリサブスクリプションを設定するために、CLI設定モードで設定コマンドを入力します。コマンドはデバイスに受け入れられますが、コンシューマはテレメトリデータを受信しません。コンシューマがテレメトリデータを確実に受信できるようにするには、どのような変更を加える必要がありますか？

```
telemetry ietf subscription 154
 encoding encode-tdl
 filter xpath /memory-ios-xe-oper:memory-statistics/memory-statistic
 source-vrf Mgmt-intf
 stream yang-push
 update-policy periodic 6000
```

A. 受信機の IP アドレスを設定する必要があります。
B. ストリーム タイプを YANG に設定する必要があります。
C. 更新ポリシー期間を短縮する必要があります。
D. 送信元 IP アドレスを設定する必要があります。
Answer: A
Explanation:
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1610/b_1610_programmability_cg/model_driven_telemetry.html

QUESTION NO: 2
図を参照してください。どのタイプのデバイスが劣化状態で機能していますか？



```
{
  "version": "1.0",
  "response": [
    {
      "time": "2019-07-15T19:10:00.000+0000",
      "healthScore": 73,
      "totalCount": 11,
      "goodCount": 8,
      "unmonCount": 3,
      "fairCount": 0,
      "badCount": 0,
      "entity": null,
      "timeinMillis": 1563217800000
    }
  ],
  "measuredBy": "global",
  "latestMeasuredByEntity": null,
  "latestHealthScore": 73,
  "monitoredDevices": 8,
  "monitoredHealthyDevices": 8,
  "monitoredUnHealthyDevices": 0,
  "unMonitoredDevices": 3,
  "healthDistribution": [
    {
      "category": "Access",
      "totalCount": 9,
      "healthScore": 100,
      "goodPercentage": 100,
      "badPercentage": 0,
      "fairPercentage": 0,
      "unmonPercentage": 0,
      "goodCount": 3,
      "badCount": 0,
      "fairCount": 0,
      "unmonCount": 0
    },
    {
      "category": "Distribution",
      "totalCount": 2,
      "healthScore": 100,
      "goodPercentage": 100,
      "badPercentage": 0,
      "fairPercentage": 0,
      "unmonPercentage": 0,
      "goodCount": 2,
      "badCount": 0,
      "fairCount": 0,
      "unmonCount": 0
    },
    {
      "category": "WLC",
      "totalCount": 2,
      "healthScore": 50,
      "goodPercentage": 0,
      "badPercentage": 0,
      "fairPercentage": 0,
      "unmonPercentage": 100,
      "goodCount": 1,
      "badCount": 0,
      "fairCount": 0,
      "unmonCount": 1
    }
  ]
}
```


A. アクセスポイント
B. 配布スイッチ
C. アクセススイッチ
D. 無線LANコントローラ
Answer: D
Explanation:
The WLC (Wireless LAN Controller) category shows a healthScore of 50, with only 1 out of 2 devices marked as "good" and 1 as "unmonitored," indicating degraded health compared to other device categories with a healthScore of 100.

QUESTION NO: 3

図を参照してください。ネットワークエンジニアは、ネットワーク内のスイッチの電源に障害が発生するたびにアラートを通知するスクリプトを作成する必要があります。このタスクを実行するために、ネットワークエンジニアはPythonスクリプトでCisco Catalyst Center（旧DNA Center）のイベントWebhookを使用しています。イベントをサブスクライブするには、コードのボックスにどのコードスニペットを追加する必要がありますか？

```
payload = [
    {
        "name": "Webhook for power supply failure",
        "description": "Power supply failure on switch",
        "subscriptionEndpoints": [
            {
                "instanceId": INSTANCEID,
                "subscriptionDetails": {
                    "connectorType": "REST",
                    [                    ]
                }
            }
        ],
        "filter": {
            "eventIds": [
                "NETWORK-DEVICES-2-201"
            ]
        }
    }
]
```

A. "connectorMethod": "POST"

B. "method": "POST"

C. "subscribeTo": "POST"

D. "connector": "POST"

Answer: B

Explanation:

When configuring event subscription endpoints in Cisco Catalyst Center (formerly DNA Center), the correct field to specify the HTTP method for REST webhooks is: "method": "POST" This ensures that events such as power supply failures are pushed to the designated endpoint using the POST method.

QUESTION NO: 4

ドラッグアンドドロップ問題

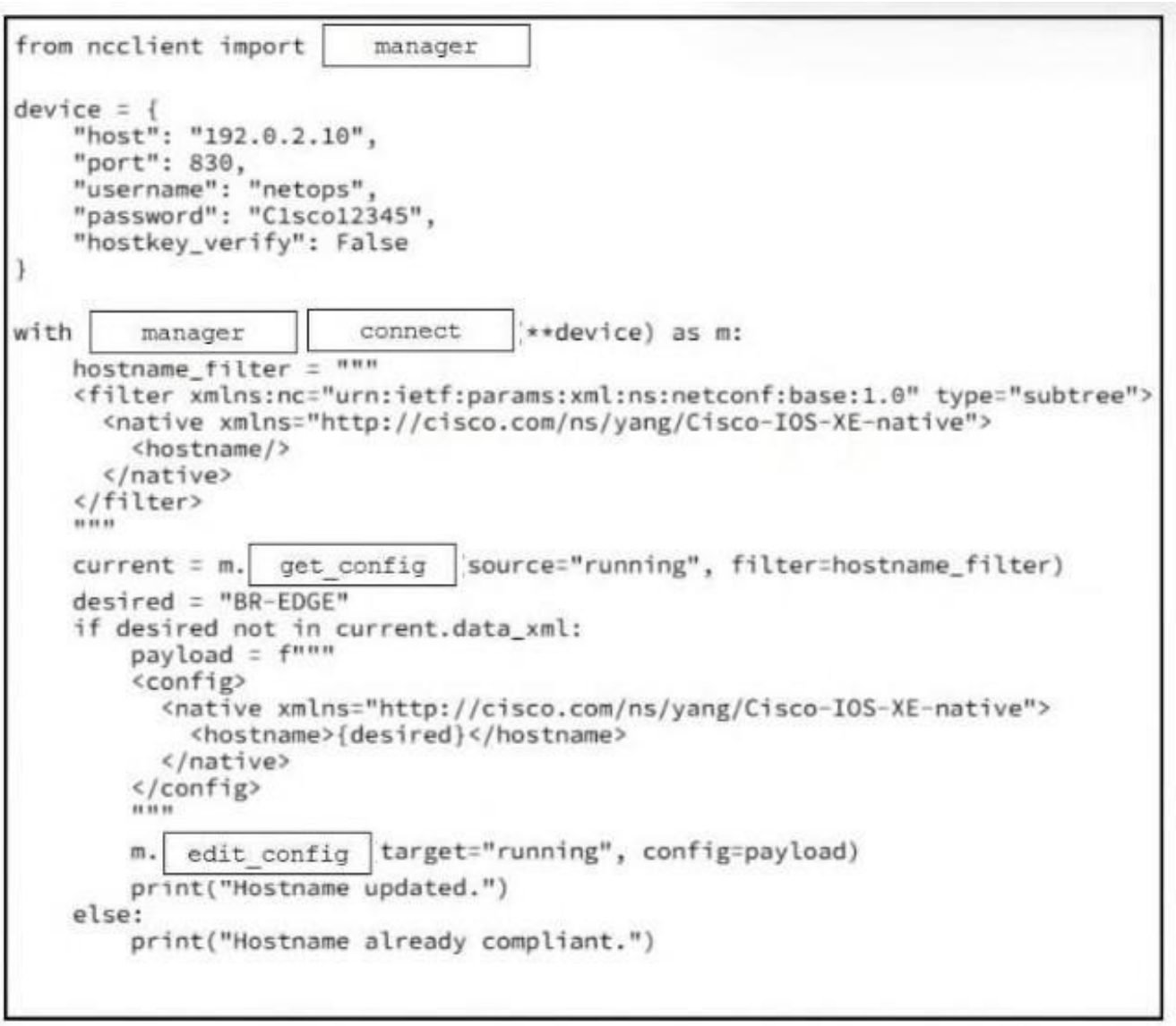
エンジニアは、PythonとNETCONFを使用してCisco IOS XEルータ上でホスト名の準拠を強制する必要があります。エンジニアはNETCONFを介して現在のホスト名を読み取り、一致しない場合は、目的のホスト名を実行中のデータストアにプッシュする必要があります。下部にあるコードスニペットをコード内のボックスにドラッグアンドドロップして、成果物を構築してください。すべてのオプションが使用されるわけ

ではありません。オプションは複数回使用できます。

```
from ncclient import [          ]

device = {
    "host": "192.0.2.10",
    "port": 830,
    "username": "netops",
    "password": "C1sco12345",
    "hostkey_verify": False
}

with [          ].[          ](**device) as m:
    hostname_filter = """
    <filter xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" type="subtree">
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <hostname/>
      </native>
    </filter>
    """
    current = m.[          ](source="running", filter=hostname_filter)
    desired = "BR-EDGE"
    if desired not in current.data_xml:
        payload = f"""
        <config>
          <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
            <hostname>{desired}</hostname>
          </native>
        </config>
        """
        m.[          ](target="running", config=payload)
        print("Hostname updated.")
    else:
        print("Hostname already compliant.")
```

connect_ssh	get_config	merge_config	edit_config
connect	manager	get	

Answer:



Explanation:
NETCONF sessions with nccclient are established through the manager module, and the connection is opened with manager.connect. To read the current hostname from the running datastore, the script uses get_config with the subtree filter. If the hostname is not compliant, the change is pushed to the running datastore with edit_config, which applies the NETCONF configuration payload.

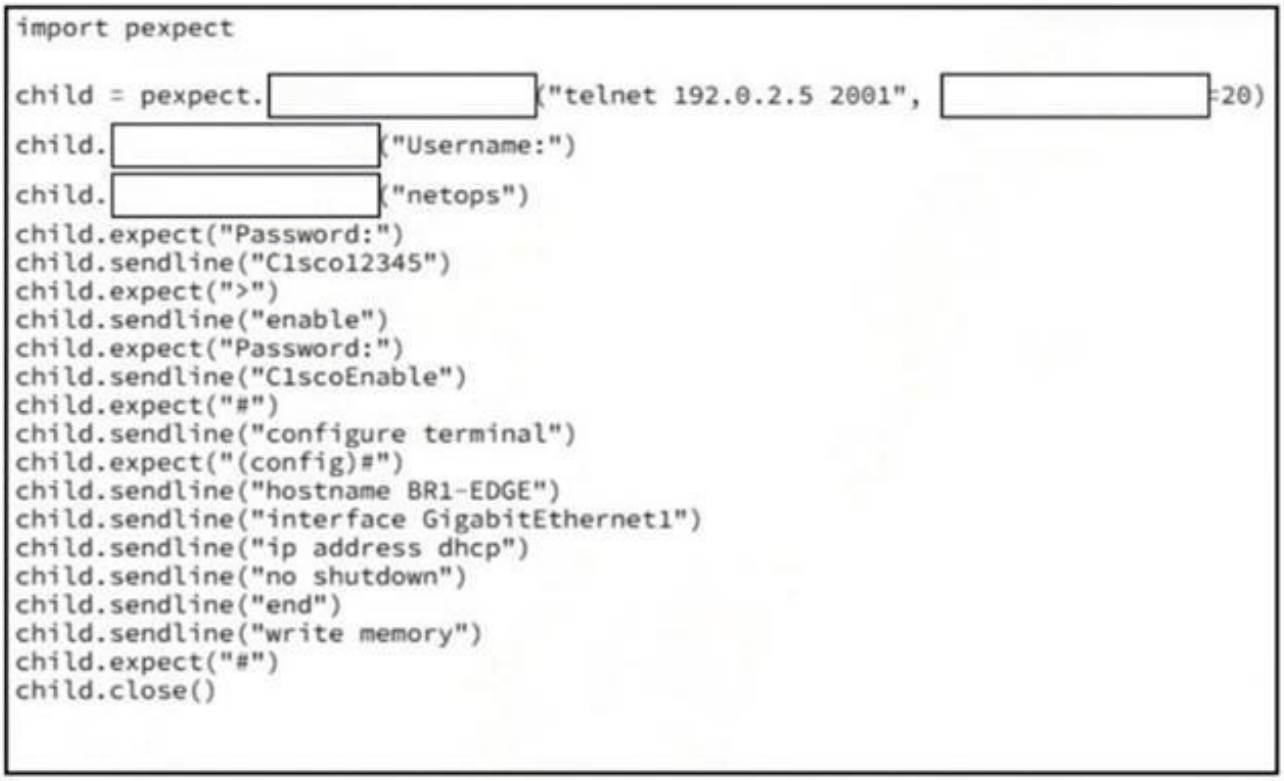
QUESTION NO: 5
展示を参照してください。ネットワークエンジニアは、Cisco IOS XEデバイスのループバックインターフェースから不要なIPアドレスを自動削除するAnsibleプレイブック実行タスクを作成します。
コード内のボックスにどのコード スニペットを追加する必要がありますか?

```
---
- name: DELETE LOOPBACK INTERFACES
  hosts: CSR1kv
  gather_facts: false
  connection: network_cli

  tasks:
  - name: DELETE UNNECESSARY IP ADDRESSES
    [          ]
      config:
        - name: Loopback10
        - name: Loopback20
        - name: Loopback30
      state: deleted
```

A. ios_l3_interfaces
B. ios_command
C. ios_config
D. ios_vrf
Answer: A
Explanation:

The correct module to use for managing Layer 3 interfaces (such as loopbacks) on Cisco IOS XE devices in an Ansible playbook is . This module allows configuration or deletion of L3 interfaces using structured data, as shown in the playbook. The state deleted ios_l3_interfaces along with a list of interface names is a valid use case for ios_l3_interfaces.

QUESTION NO: 6
ドラッグアンドドロップ問題
エンジニアは、Day-0セットアップのためにターミナルサーバーに接続されたルーターに初期CLI設定を適用する必要があります。エンジニアは、pexpectを使用してTelnet経由でコンソールポートにログインして設定を自動化し、変更を保存する必要があります。下部にあるコードスニペットをコード内のボックスにドラッグアンドドロップして、成果物を構築してください。すべてのオプションが使用されるわけではありません。



Answer:

Explanation:
In pexpect, the Telnet session is started with spawn, and the session timeout is set with the timeout argument. The script then waits for the Username prompt with expect and sends the username with sendline. This enables the automated console login flow before applying the initial router configuration and saving it.

QUESTION NO: 7
展示を参照してください。Cisco SD-WAN vManage Device Inventory API を呼び出して vEdge のリストを取得する Python スクリプトが作成されています。Python ディクショナリに返される JSON
データは変換され、「d」という変数に代入されています。JSON
の一部が展示に示されています。ホスト名にアクセスするために、式 hostname= を完成させるコードはどれですか？

```
{
  'data':
    [
      {
        'availableVersions':[]
        'chasisNumber': '4af9e049-0052-47e9-83af-81a5825f7ffe',
        'deviceIP': '4.4.4.60',
        'deviceModel': 'vedge-cloud',
        'deviceType': 'vedge',
        'host-name': 'vedge01',
        ...
```

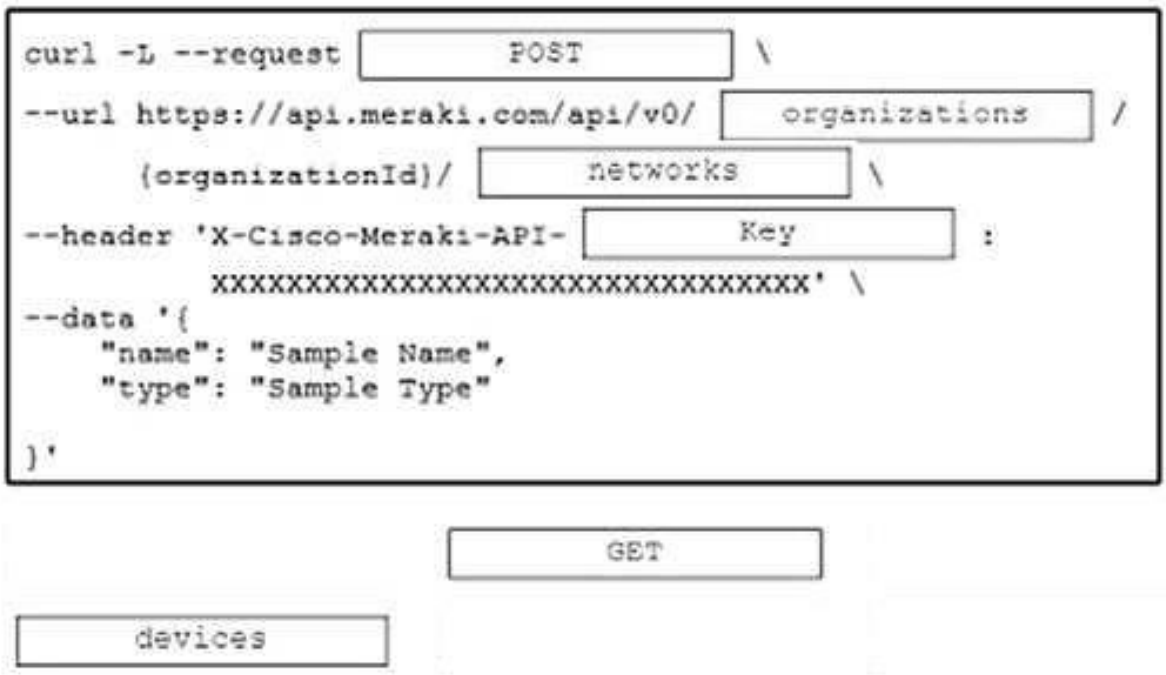
A. d["データ"][0]["ホスト名"]
B. d[データ][0][ホスト名]
C. d("データ")[0]("ホスト名")
D. d["ホスト名"]["データ"]{"0"}
Answer: A
Explanation:
The double-quotations are a necessary syntax of Python. And for the json portion doesnt use parentheses. It always uses brackets. d["data"][0]["host-name"] is the only logical answer.

QUESTION NO: 8
ドラッグアンドドロップの質問
下部のコードスニペットをコード内の空白部分にドラッグ＆ドロップして、新しいCisco Meraki組織のネットワークを作成します。すべてのオプションが使用されていませんか？

```
curl -L --request [          ] \
--url https://api.meraki.com/api/v0/ [          ] /
      {organizationId}/ [          ] \
--header 'X-Cisco-Meraki-API-[          ] :
          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX' \
--data '{
    "name": "Sample Name",
    "type": "Sample Type"

}'
```

Key	GET	organizations
devices	POST	networks

Answer:



QUESTION NO: 9
従来のネットワークとソフトウェア定義ネットワークを比較するとどうなりますか?
A. ソフトウェア定義ネットワークは Intent API
の使用をサポートし、従来のネットワークは通常ハードウェアベースです。
B.
ソフトウェア定義ネットワークにより分散ネットワーク構成の使用が可能になり、従来のネットワークは集中化されます。
C. 従来のネットワークでは動的アクセス
リストが使用され、ソフトウェア定義ネットワークでは個別のハードウェアをプログラムする必要があります。
D.
従来のネットワークは集中型のネットワーク管理を提供しますが、ソフトウェア定義ネットワークはデバイスに依存します。
Answer: A
Explanation:
Software-defined networks (SDN) support the use of Intent APIs to enable programmability and automation, while traditional networks are typically hardware-based and managed manually.

QUESTION NO: 10
ドラッグアンドドロップの質問
下部のコードスニペットをコード内のボックスにドラッグ & ドロップして、運用データを有効にし、インターフェースデータを取得します。すべてのオプションが使用されるわけではありません。

```
from ncclient import manager

def get_interface_state(host, port, user, passwd, filename):
    with manager.connect(host, port, user, passwd) as m:
        with open [          ] as f:
            rpc_reply = m.[          ](target='running', config=f.read())
        rpc_reply = m.[          ](filter=('subtree', "<interfaces-state/>"))
    return(rpc_reply)
```

(filen)	edit_setup	get
put	(filename)	edit_config

Answer:

```
from ncclient import manager

def get_interface_state(host, port, user, passwd, filename):
    with manager.connect(host, port, user, passwd) as m:
        with open (filename) as f:
            rpc_reply = m.edit_config(target='running', config=f.read())
        rpc_reply = m.get(filter=('subtree', "<interfaces-state/>"))
    return(rpc_reply)
```

(filen)	edit_setup
put	

Explanation:

```
with open(filename) as f:
    rpc_reply = m.edit_config(target='running', config=f.read())
rpc_reply = m.get(filter=('subtree', "<interfaces-state/>"))
```

(filename) is used to open the file specified by the function parameter.
edit_config is the NETCONF operation for editing the running config.
get retrieves operational state data such as interface state.

QUESTION NO: 11
ドラッグアンドドロップ問題
エンジニアは、エグゼクティブダッシュボード用に、Cisco Catalyst Center

からサイトレベルのヘルススナップショットを作成する必要があります。エンジニアは、トークンヘッダーとミリ秒単位のタイムスタンプクエリパラメータを使用して、サイトヘルスリソースを認証し、クエリを実行する必要があります。下部にあるコードスニペットをコード内のボックスにドラッグアンドドロップして、成果物を作成します。すべてのオプションが使用されるわけではありません。

```
import requests, time

base = "https://dnac.example.local"
user, pwd = ("apiuser", "apipass")

# token
t = requests.post(f"{base}/[          ]", auth=(user, pwd), verify=False)
t.raise_for_status()
token = t.json()["Token"]

# site-health
ts = int(time.time() * 1000)
headers = {"[          ]": token}
r = requests.get(f"{base}/dna/intent/api/v1/[          ]?[          ]{ts}", headers=headers, verify=False, timeout=15)
r.raise_for_status()
site_data = r.json()
print("Sites reported:", len(site_data.get("response", [])))
```

x-auth-token | site-health | startTime | network-health | timestamp | auth/token | Authorization

Answer:

```
import requests, time

base = "https://dnac.example.local"
user, pwd = ("apiuser", "apipass")

# token
t = requests.post(f"{base}/ auth/token ', auth=(user, pwd), verify=False)
t.raise_for_status()
token = t.json()["Token"]

# site-health
ts = int(time.time() * 1000)
headers = {" x-auth-token ': token}
r = requests.get(f"{base}/dna/intent/api/v1/ site-health ' timestamp 'ts}", headers=headers, verify=False, timeout=15)
r.raise_for_status()
site_data = r.json()
print("Sites reported:", len(site_data.get("response", [])))
```

startTime | network-health | Authorization

Explanation:
Cisco Catalyst Center authentication is performed against the auth/token endpoint to obtain the API token. Subsequent requests must pass that token in the x-auth-token header. The site-level health snapshot is retrieved from the site-health resource, and the query uses the timestamp parameter with the required millisecond value to return the health view for that point in time.

QUESTION NO: 12
ドラッグアンドドロップの質問
下部のコードスニペットをコード内のボックスにドラッグ＆ドロップして、Cisco Meraki Webhookからの受信イベントを処理するPythonのHTTPレシーバーを実装します。すべてのオプションが使用されるわけではありません。

```
from flask import [          ] , request, jsonify

app = Flask(__name__)

@app.route("/", methods=[" [          ] "])
def webhook():
    data = [          ].json
    print(f"{data['alertLevel'].upper()}: Alarm Type {data['alertType']}")
    return jsonify({"status": "success"})

if __name__ == "__main__":
    [          ].run(host="0.0.0.0", port=5000, threaded=True,
ssl_context=("serv.crt", "serv.key"))
```

http	request	Flask
GET	app	POST

Answer:

```
from flask import [ Flask ] , request, jsonify

app = Flask(__name__)

@app.route("/", methods=[" [ POST ] "])
def webhook():
    data = [ request ].json
    print(f"{data['alertLevel'].upper()}: Alarm Type {data['alertType']}")
    return jsonify({"status": "success"})

if __name__ == "__main__":
    [ app ].run(host="0.0.0.0", port=5000, threaded=True,
ssl_context=("serv.crt", "serv.key"))
```

http
GET

Explanation:

```python
from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route("/", methods=["POST"])
def webhook():
    data = request.json
    print(f"{data['alertLevel'].upper()}: Alarm Type {data['alertType']}")
    return jsonify({"status": "success"})

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000, threaded=True, ssl_context=("serv.crt", "serv.key"))
```

Flask is the class to create the app.
"POST" is the method for receiving webhook events.
request.json retrieves the incoming JSON data.
app.run(...) starts the Flask app.

QUESTION NO: 13
コントローラーベースのプラットフォームにおいて、トラフィック分析はどのような役割を果たすのでしょうか？
A.分離のためのパケットフィルタリング
B. 冗長性のためのアーカイブ検索
C. 制限に関するポリシーの適用
D.最適化のための流量測定
Answer: D
Explanation:
Traffic analytics in controller-based platforms is used to measure and analyze network flows so the controller can understand usage patterns, congestion, and performance trends. That visibility supports optimization decisions such as improving path selection, balancing traffic, and tuning network behavior.

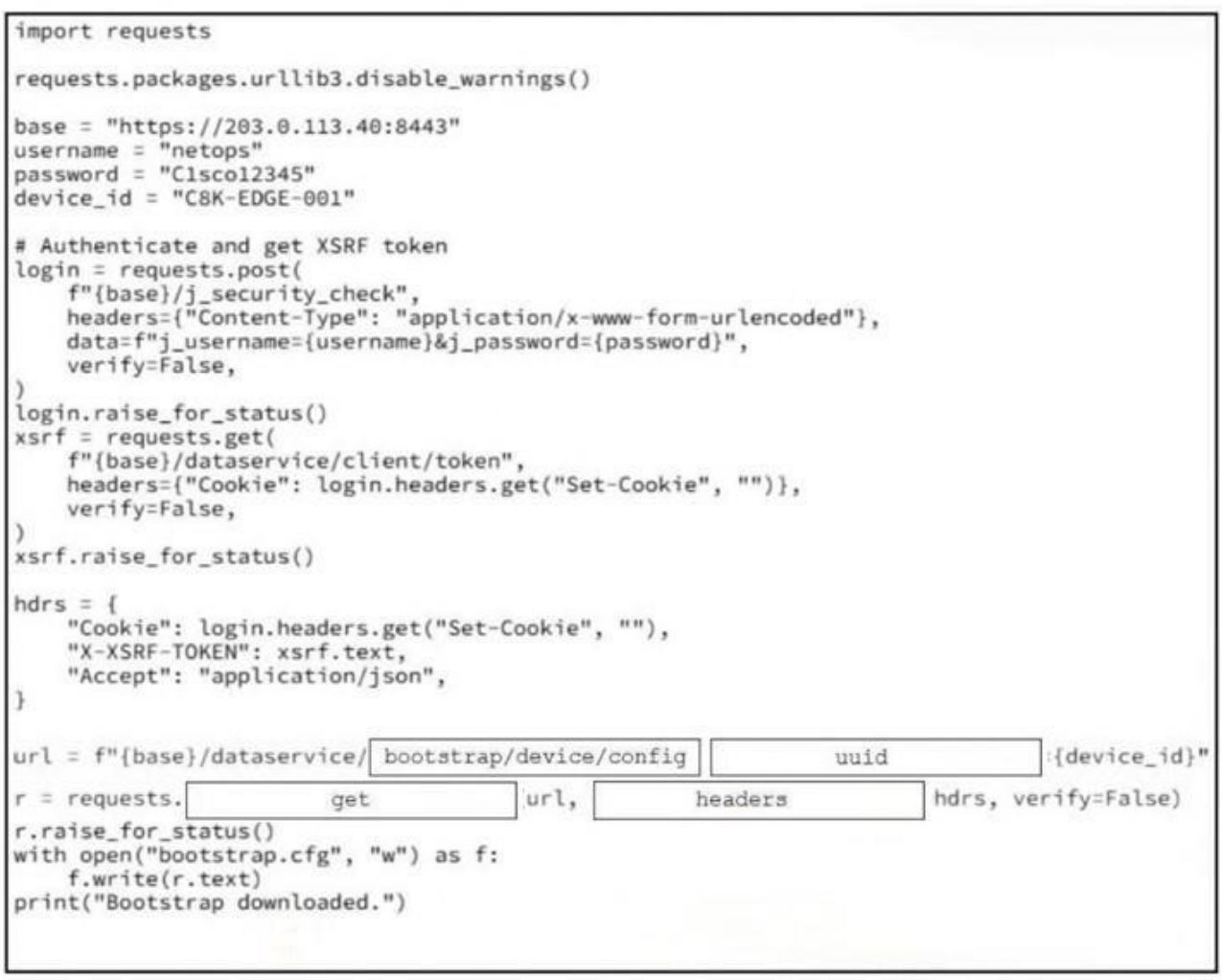
QUESTION NO: 14
エンジニアは、MerakiAPIを使用して新しいネットワークを作成する必要があります。 URL https://api.meraki.com/api/v0/organizations/<new_org_id>/networksへのどのHTTPアクションが201応答コードになりますか？
A.GET
B.POST
C.PUT
D.追加
Answer: B
Explanation:
https://httpstatuses.com/201

QUESTION NO: 15
ドラッグアンドドロップ問題
エンジニアは、vManageからWANエッジブートストラップ構成をエクスポートして、SD-WAN Day
0の準備を行う必要があります。エンジニアはvManageで認証を行い、XSRFトークンを取得し、特定のデバイスIDのブートストラップファイルをGETする必要があります。下部にあるコードスニペットをコード内のボックスにドラッグアンドドロップして、成果物を構築してください。すべてのオプションが使用されるわけではありません。

```
import requests

requests.packages.urllib3.disable_warnings()

base = "https://203.0.113.40:8443"
username = "netops"
password = "C1sco12345"
device_id = "C8K-EDGE-001"

# Authenticate and get XSRF token
login = requests.post(
    f"{base}/j_security_check",
    headers={"Content-Type": "application/x-www-form-urlencoded"},
    data=f"j_username={username}&j_password={password}",
    verify=False,
)
login.raise_for_status()
xsrf = requests.get(
    f"{base}/dataservice/client/token",
    headers={"Cookie": login.headers.get("Set-Cookie", "")},
    verify=False,
)
xsrf.raise_for_status()

hdrs = {
    "Cookie": login.headers.get("Set-Cookie", ""),
    "X-XSRF-TOKEN": xsrf.text,
    "Accept": "application/json",
}

url = f"{base}/dataservice/[ ]?[ ]={device_id}"
r = requests.[ ](url, [ ]=hdrs, verify=False)
r.raise_for_status()
with open("bootstrap.cfg", "w") as f:
    f.write(r.text)
print("Bootstrap downloaded.")
```

post	bootstrap/device/config	uuid
deviceId	device/bootstrap/config	headers
get		

Answer:



```
import requests

requests.packages.urllib3.disable_warnings()

base = "https://203.0.113.40:8443"
username = "netops"
password = "C1sco12345"
device_id = "C8K-EDGE-001"

# Authenticate and get XSRF token
login = requests.post(
    f"{base}/j_security_check",
    headers={"Content-Type": "application/x-www-form-urlencoded"},
    data=f"j_username={username}&j_password={password}",
    verify=False,
)
login.raise_for_status()
xsrf = requests.get(
    f"{base}/dataservice/client/token",
    headers={"Cookie": login.headers.get("Set-Cookie", "")},
    verify=False,
)
xsrf.raise_for_status()

hdrs = {
    "Cookie": login.headers.get("Set-Cookie", ""),
    "X-XSRF-TOKEN": xsrf.text,
    "Accept": "application/json",
}

url = f"{base}/dataservice/[bootstrap/device/config]?[uuid]={device_id}"
r = requests.[get](url, [headers]=hdrs, verify=False)
r.raise_for_status()
with open("bootstrap.cfg", "w") as f:
    f.write(r.text)
print("Bootstrap downloaded.")
```

[post]

[deviceId]

[device/bootstrap/config]

Explanation:
After authenticating to vManage and retrieving the XSRF token, the bootstrap configuration must be downloaded with an HTTP GET request. The request must include the session cookie and XSRF token in the headers parameter, and the target bootstrap export endpoint is built with the bootstrap configuration path plus the device identifier passed as the uuid query value.

QUESTION NO: 16
展示を参照してください。Pythonコードから期待される出力は何ですか？

```
# Simple Application to run a few commands on a Cisco Device
ipaddresses = ['192.168.0.1', "192.168.0.5", "10.10.10.10"]
username = "admin"
password = "cisco123"
commands_to_run=["show ver", "show ip interface brief"]
Debug = True


for device in ipaddresses:
      print ("Logging into "+device+", using "+username+"/"+password)

      # We want to execute commands on our device only if Debug=True

      for commands in commands_to_run:
            print ("    Executing "+commands+" on device: "+device)
```

A.

```
Logging into 192.168.0.1, using admin/cisco123
Logging into 192.168.0.5, using admin/cisco123
Logging into 10.10.10.10, using admin/cisco123
      Executing show ver on device: 192.168.0.1
      Executing show ip interface brief on device: 192.168.0.1
      Executing show ver on device: 192.168.0.5
      Executing show ip interface brief on device: 192.168.0.5
      Executing show ver on device: 10.10.10.10
      Executing show ip interface brief on device: 10.10.10.10
```

B.

```
Logging into 192.168.0.1, using admin/cisco123
Logging into 192.168.0.5, using admin/cisco123
Logging into 10.10.10.10, using admin/cisco123
```

C.

```
Simple Application to run a few commands on a Cisco Device
Loggig into 192.168.0.1, using admin/cisco123
We want to execute commands on our device only if Debug=True
      Executing show ver on device: 192.168.0.1
      Executing show ip interface brief on device: 192.168.0.1
Logging into 192.168.0.5, using admin/cisco123
We want to execute commands on our device only if Debug=True
      Executing show ver on device: 192.168.0.5
      Executing show ip interface brief on device: 192.168.0.5
Logging into 10.10.10.10, using admin/cisco123
We want to execute commands on our device only if Debug=True
      Executing show ver on device: 10.10.10.10
      Executing show ip interface brief on device: 10.10.10.10
```

D.

```
Logging into 192.168.0.1, using admin/cisco123
      Executing show ver on device: 192.168.0.1
      Executing show ip interface brief on device: 192.168.0.1
Logging into 192.168.0.5, using admin/cisco123
      Executing show ver on device: 192.168.0.5
      Executing show ip interface brief on device: 192.168.0.5
Logging into 10.10.10.10, using admin/cisco123
      Executing show ver on device: 10.10.10.10
      Executing show ip interface brief on device: 10.10.10.10
```

Answer: D

QUESTION NO: 17
空欄を埋める
空白を埋めて、デバイスにログインしているユーザーのリストを取得する Cisco SD-WAN API の URL を完成させます。

http://<vmanage-ip-address>/dataservice/device/ [ ] deviceid=<deviceid>>

Answer: users?
Explanation:
https://developer.cisco.com/docs/sdwan/#!device-realtime-monitoring/users API call for real-time monitoring of users logged in to the device.
Device Users
Display the users currently logged in to the device.
GET https://{vmanage-ip-address}/dataservice/device/users?deviceId=deviceId

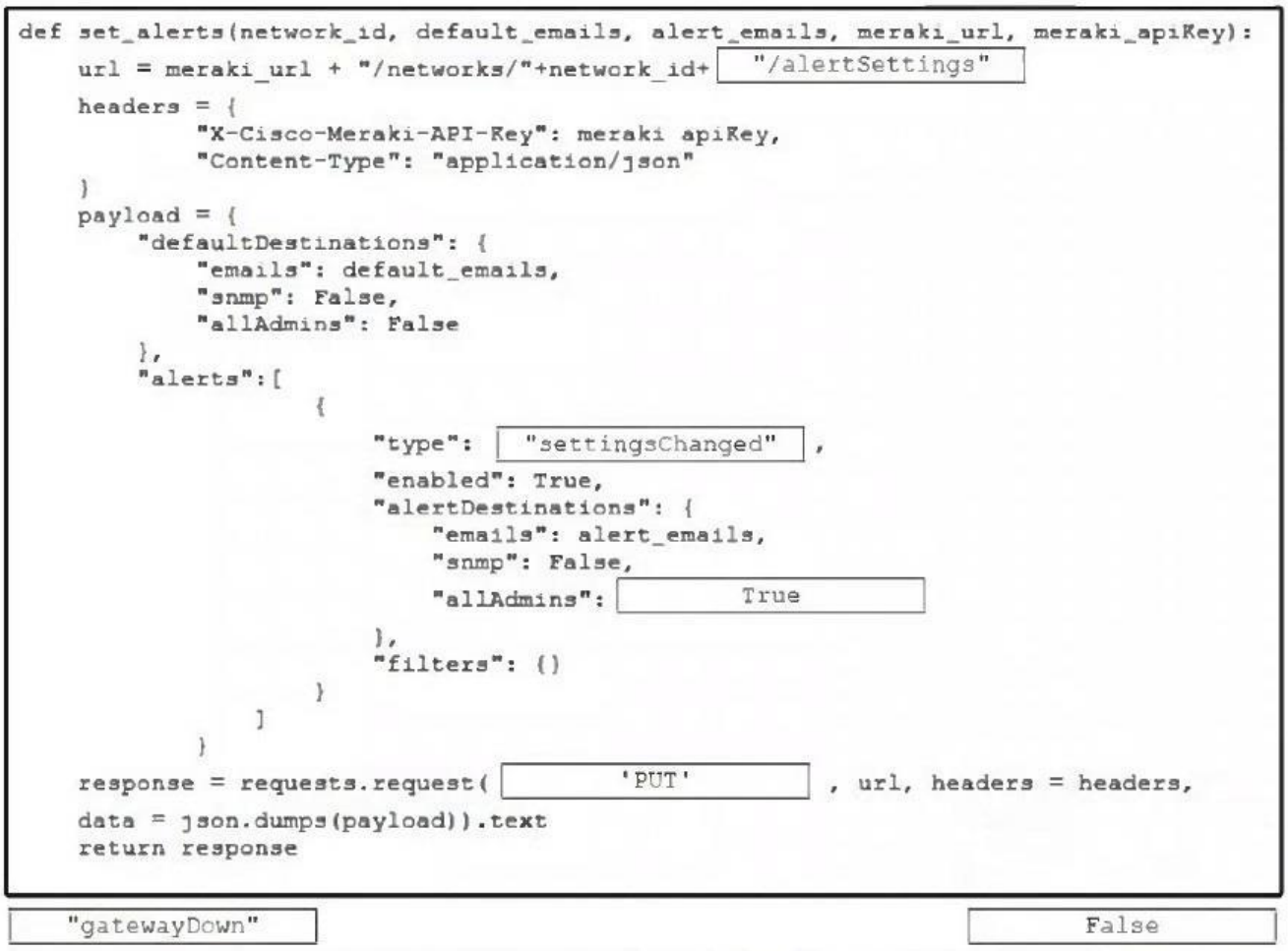
QUESTION NO: 18
ドラッグアンドドロップの質問
下部のコードスニペットをコード内のボックスにドラッグ＆ドロップして、Cisco Meraki API を使用して設定が変更された場合にすべての管理者にメールを送信するように

set_alerts 関数を設定します。すべてのオプションが使用されるわけではありません。

```
def set_alerts(network_id, default_emails, alert_emails, meraki_url, meraki_apiKey):
    url = meraki_url + "/networks/"+network_id+ [          ]
    headers = {
            "X-Cisco-Meraki-API-Key": meraki_apiKey,
            "Content-Type": "application/json"
    }
    payload = {
        "defaultDestinations": {
            "emails": default_emails,
            "snmp": False,
            "allAdmins": False
        },
        "alerts":[
                    {
                        "type": [          ],
                        "enabled": True,
                        "alertDestinations": {
                            "emails": alert_emails,
                            "snmp": False,
                            "allAdmins": [          ]
                        },
                        "filters": {}
                    }
                ]
            }
    response = requests.request( [          ] , url, headers = headers,
    data = json.dumps(payload)).text
    return response
```

"gatewayDown"	'PUT'	"/alertSettings"	False
"settingsChanged"	True		

Answer:



```
def set_alerts(network_id, default_emails, alert_emails, meraki_url, meraki_apiKey):
    url = meraki_url + "/networks/"+network_id+ "/alertSettings"
    headers = {
            "X-Cisco-Meraki-API-Key": meraki_apiKey,
            "Content-Type": "application/json"
    }
    payload = {
        "defaultDestinations": {
            "emails": default_emails,
            "snmp": False,
            "allAdmins": False
        },
        "alerts":[
                    {
                        "type": "settingsChanged" ,
                        "enabled": True,
                        "alertDestinations": {
                            "emails": alert_emails,
                            "snmp": False,
                            "allAdmins": True
                        },
                        "filters": []
                    }
                ]
            }
    response = requests.request( 'PUT' , url, headers = headers,
    data = json.dumps(payload)).text
    return response
```

"gatewayDown"

False

Explanation:
To correctly configure Meraki alert settings via the API:
The endpoint is "/alertSettings" appended to the network URL.
The alert type for configuration changes is "settingsChanged".
To notify all admins, the allAdmins field must be set to True.
The HTTP method used to modify alert settings is 'PUT'.

QUESTION NO: 19
Meraki Location Scanning
APIを使用して構築された2つのタイプのソリューションはどれですか？
（2つ選択してください。）
A. networking automation
B. mapping
C. guest Wi-Fi
D. Sense
E. wayfinder
Answer: BE
Explanation:
https://developer.cisco.com/meraki/build/wayfinding-mapwize/

QUESTION NO: 20
展示を参照してください。このYANGモジュールの有効なXMLインスタンスは何ですか？

```
module interfaces {

  typedef dotted-quad {
    type string {
      pattern
        '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'
          + '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])';
    }
    description
      "Four octets written as decimal numbers and
       separated with the '.' (full stop) character.";
  }

  container interfaces {
    list interface {
      key "name";
      leaf name {
        type string;
        mandatory "true";
        description
          "Interface name.";
      }
      leaf address {
        type dotted-quad;
        mandatory "true";
        description
          "Interface IP address.";
      }
      leaf subnet-mask {
        type dotted-quad;
        mandatory "true";
        description
          "Interface subnet mask.";
      }
      leaf enabled {
        type boolean;
        default "false";
        description
          "Enable or disable the interface.";
      }
    }
  }
}
```

```
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <interfaces xmlns="http://example.com/interfaces">
    <interface>
      <name>GigabitEthernet 0/0/0</name>
      <address>10.10.10.1</address>
      <subnet-mask>255.255.255.0</subnet-mask>
    </interface>
    <interface>
      <name>GigabitEthernet 0/0/1</name>
      <address>192.168.1.1</address>
      <subnet-mask>255.255.255.0</subnet-mask>
    </interface>
  </interfaces></data>
```

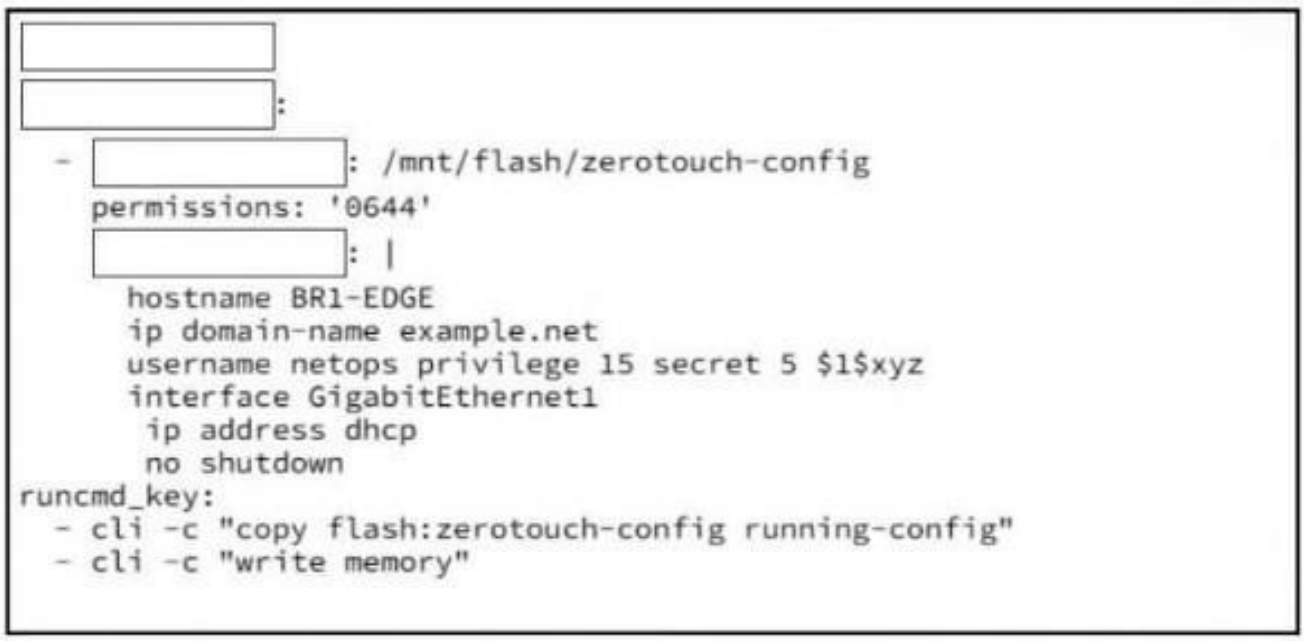
```
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <interfaces xmlns="http://example.com/interfaces">
    <interface>
      <name>GigabitEthernet 0/0/0</name>
      <address>10.10.10.1</address>
      <enabled>true</enabled>
    </interface>
    <interface>
      <name>GigabitEthernet 0/0/1</name>
      <address>192.168.1.1</address>
      <subnet-mask>255.255.255.0</subnet-mask>
      <enabled>true</enabled>
    </interface></interfaces></data>
```

```
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <interfaces xmlns="http://example.com/interfaces">
    <interface>
      <name>GigabitEthernet 0/0/0</name>
      <address> 2001:db8::2:1</address>
      <subnet-mask>255.255.255.0</subnet-mask>
    </interface>
    <interface>
      <name>GigabitEthernet 0/0/1</name>
      <address> 2001:db8::2:1</address>
      <subnet-mask>255.255.255.0</subnet-mask>
    </interface>
  </interfaces></data>
```

```
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <interfaces xmlns="http://example.com/interfaces">
    <interface>
      <address>10.10.10.1</address>
      <subnet-mask>255.255.255.0</subnet-mask>
    </interface>
    <interface>
      <address>192.168.1.1</address>
      <subnet-mask>255.255.255.0</subnet-mask>
    </interface>
  </interfaces>
</data>
```

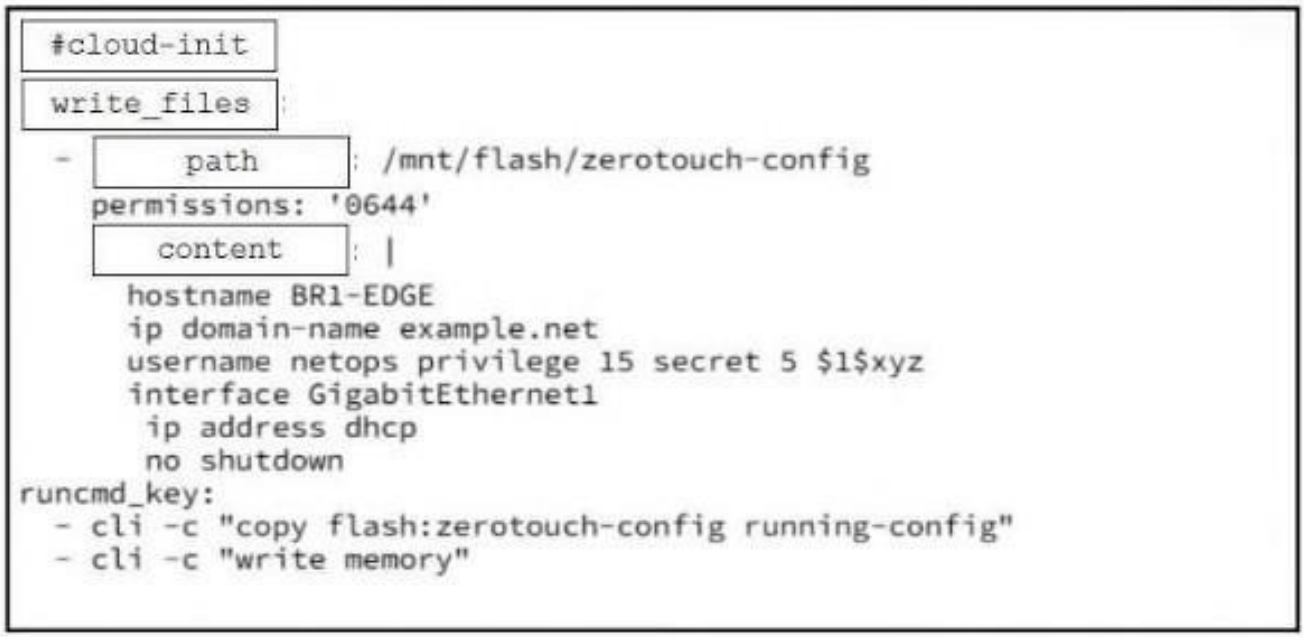
A. オプションA
B. オプションB
C. オプションC
D. オプションD
Answer: A

QUESTION NO: 21
ドラッグアンドドロップ問題
エンジニアは、ゼロタッチプロビジョニングをサポートする仮想ネットワークデバイス用のDay-0クラウド初期化ファイルを作成する必要があります。エンジニアは、起動CLIをファイルに書き込み、初回起動時にそれを実行構成にマージするコマンドを実行する必要があります。下部のコードスニペットをコード内のボックスにドラッグアンドドロップして、成果物を構築してください。すべてのオプションが使用されるわけではありません。



#cloud-init | path | #cloud-header | files_write
content | write_files | commands

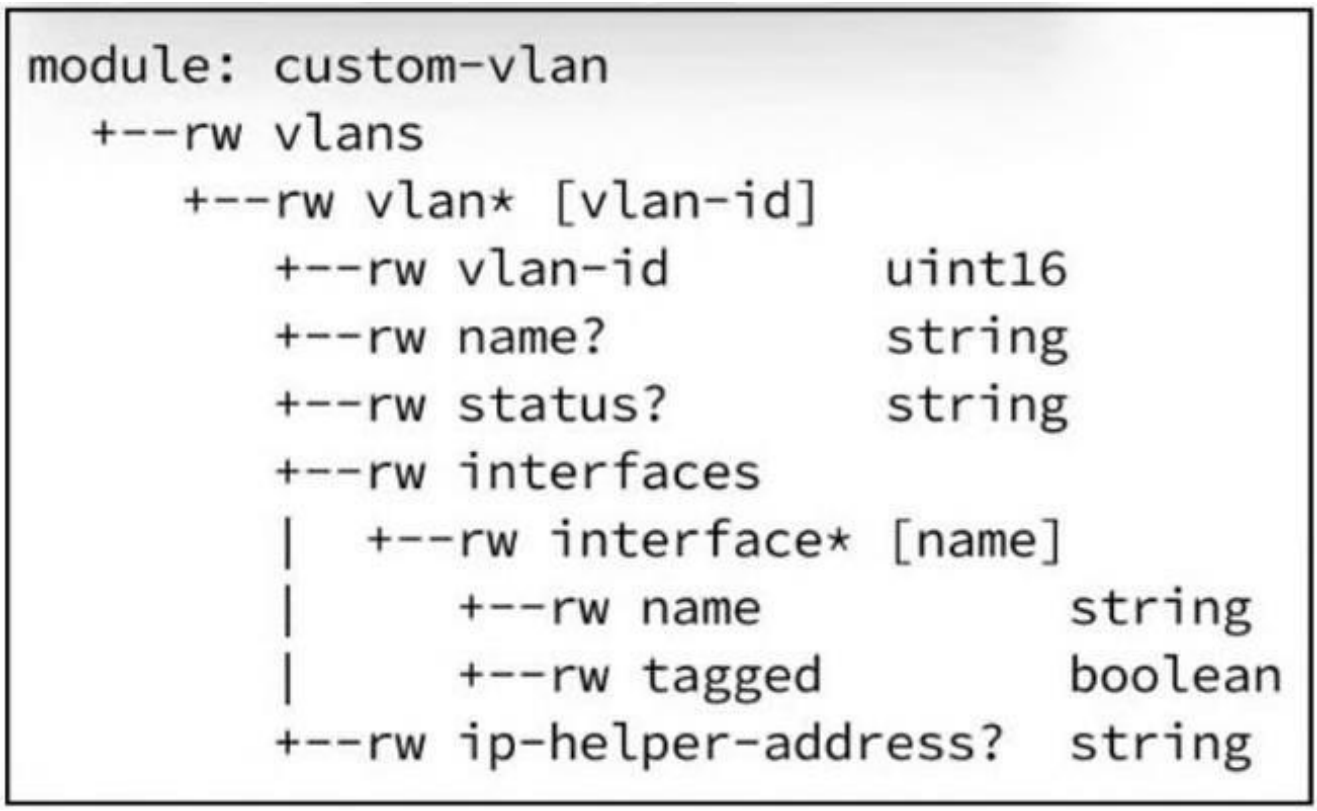
Answer:



#cloud-header | files_write
commands

Explanation:
A cloud-init Day-0 artifact must begin with the cloud-init header, then use the write_files section to place the bootstrap configuration on disk. Inside that section, path defines where the startup CLI file is written, and content provides the multiline CLI configuration that will later be merged into the running configuration by the boot-time commands.

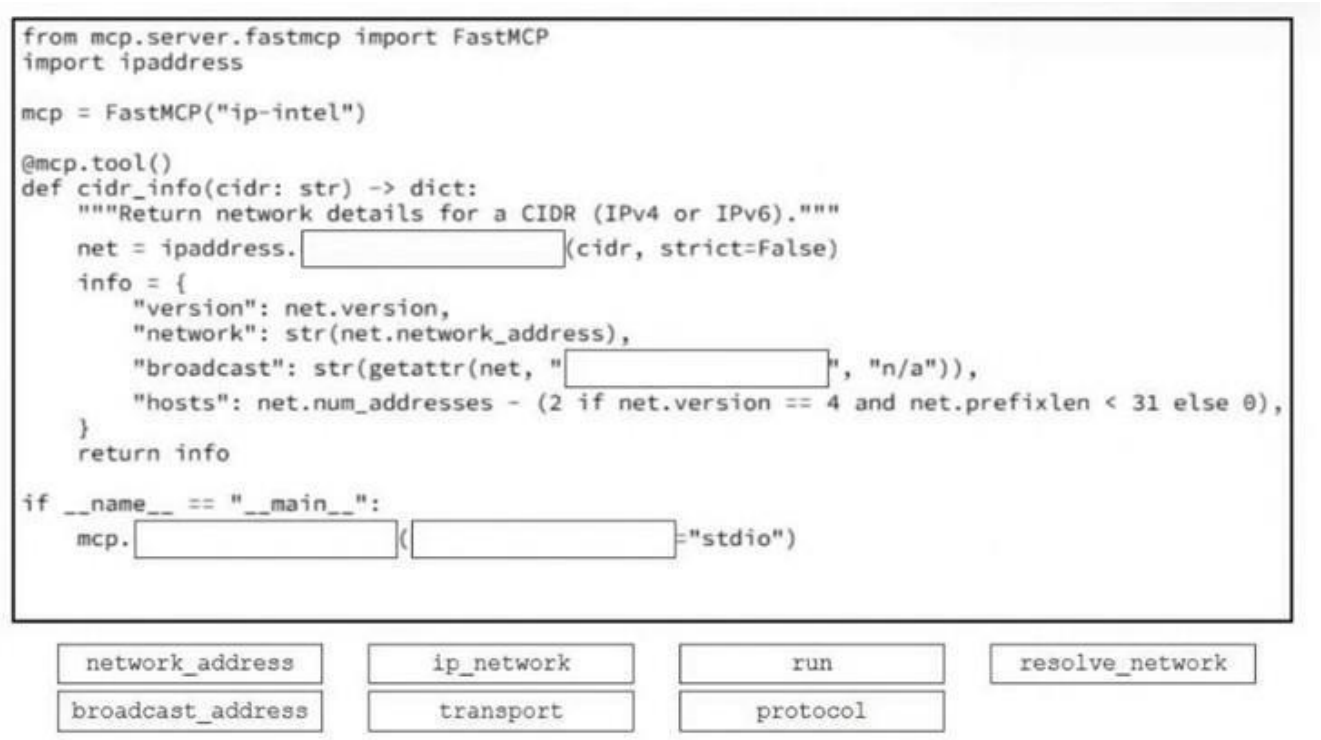
QUESTION NO: 22
図を参照してください。ネットワーク自動化エンジニアが、VLANメンバーシップとインターフェースの階層配置を解釈するために、VLAN
YANGモジュールツリーを検証しています。エンジニアは、ネットワークコントローラで正確な構成アクションを自動化したいと考えています。これらの要件を満たすノードパスはどれですか？



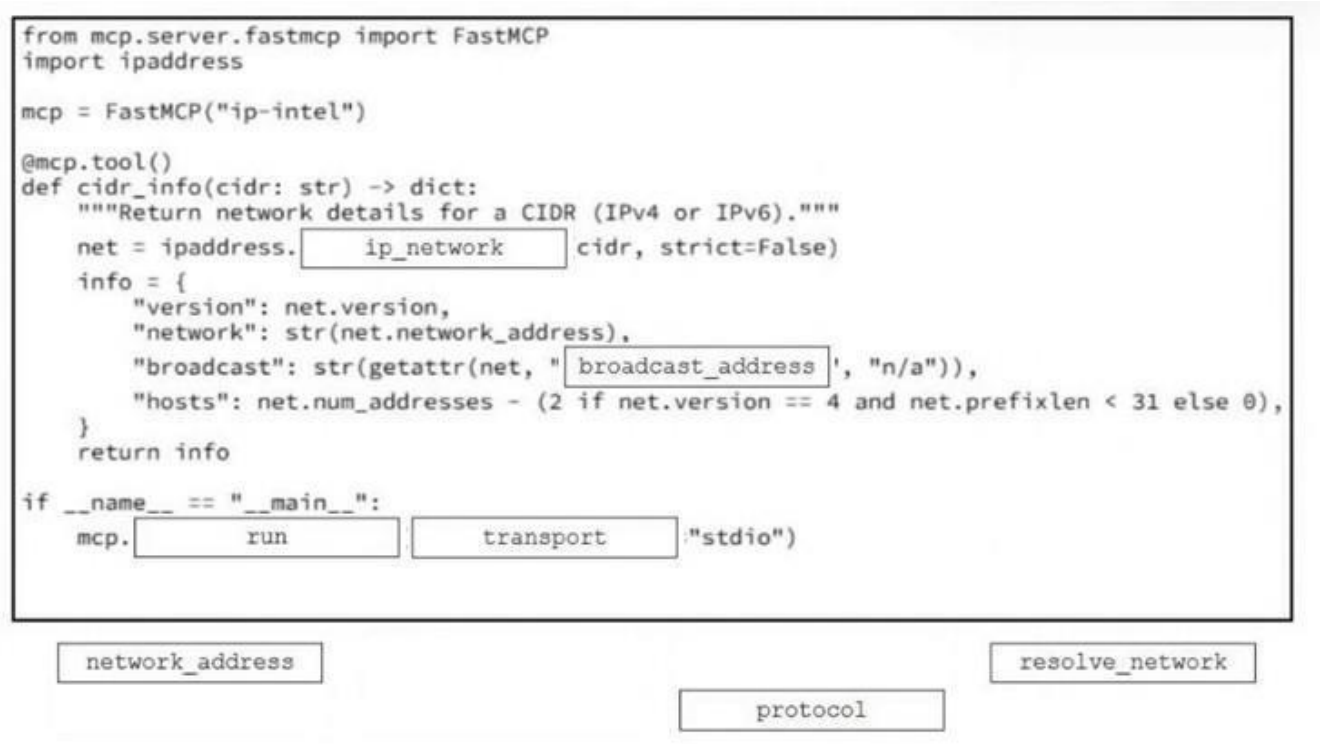
A. vlans/vlan/interfaces/interface
B. vlans/vlan/data/interface
C. VLAN/VLAN/インターフェース/名前
D. VLAN/VLAN/インターフェース/インターフェース名
Answer: A
Explanation:
The YANG tree shows that VLAN membership interfaces are organized under the interfaces container within each vlan list entry, and each member is represented as an interface list item.
That makes vlans/vlan/interfaces/interface the correct node path for automating configuration of VLAN interface membership.

QUESTION NO: 23
ドラッグアンドドロップ問題
エンジニアは、変更計画中にアドレス指定について推論できるように、AIエージェントにサブネット情報を提供する必要があります。エンジニアは、CIDRを受け取り、ネットワーク、ブロードキャスト、ホストの数を返すFastMCPツールを実装し、stdio経由でサーバーを起動する必要があります。下部にあるコードスニペットをコード内のボックスにドラッグアンドドロップして、成果物を構築してください。すべてのオプションが使用されるわけでは

ありません。



Answer:



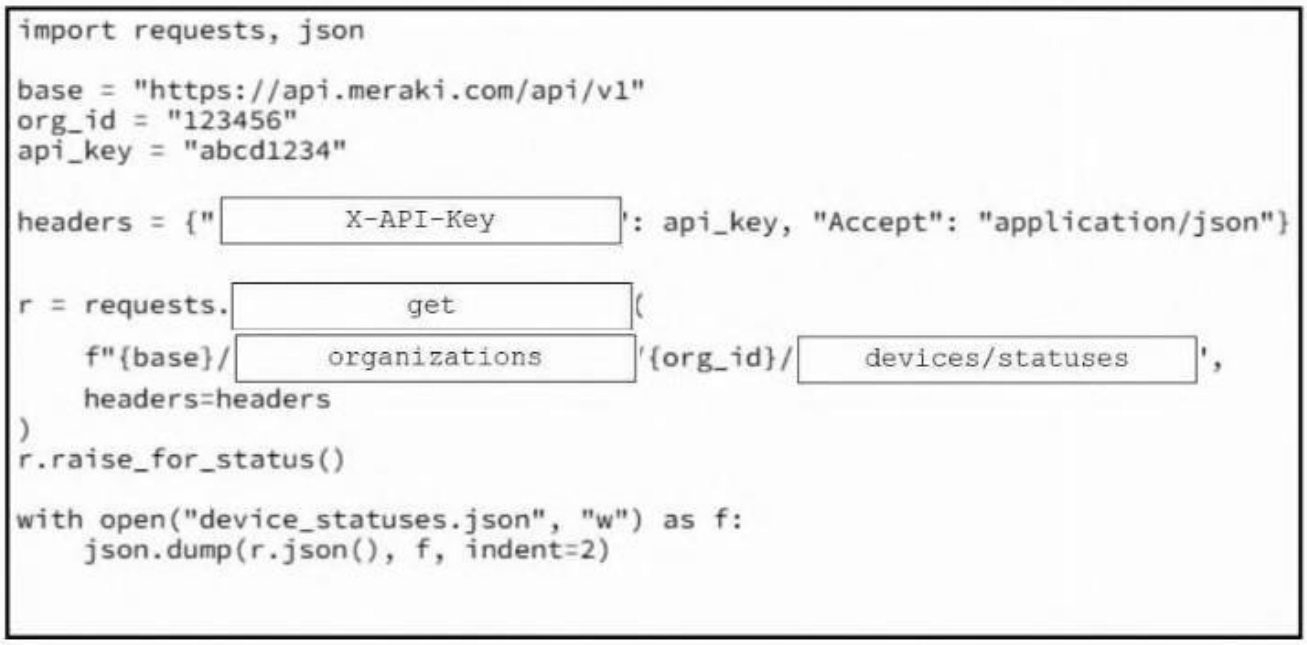
Explanation:
The ip_network function parses the CIDR into a network object that exposes the addressing attributes needed by the tool. The broadcast value is obtained from the broadcast_address attribute, which is used when present and safely handled for IPv6 with the fallback already shown. To start the FastMCP server over standard I/O, the server is launched with run and the mode is specified with the transport argument set to stdio.

QUESTION NO: 24
ドラッグアンドドロップ問題
エンジニアは、Meraki組織全体のデバイスの状態を監視する必要があります。エンジニアは、組織の
devices/statusesエンドポイントを介してデバイスの状態を取得し、JSONをアーカイブする必要があります。下部にあるコードスニペットをコード内のボックスにドラッグアンドドロップして、成果物を作成してください。すべてのオプションが使用されるわけではありません。

```
import requests, json

base = "https://api.meraki.com/api/v1"
org_id = "123456"
api_key = "abcd1234"

headers = {"                    ": api_key, "Accept": "application/json"}

r = requests.                   (
    f"{base}/                   /{org_id}/                   ",
    headers=headers
)
r.raise_for_status()

with open("device_statuses.json", "w") as f:
    json.dump(r.json(), f, indent=2)
```

organizations	post
X-API-Key	devices/statuses
get	organizations/statuses
x-cisco-meraki-api-key	

Answer:



Explanation:
The Meraki Dashboard API authenticates requests with the X-API-Key header. Retrieving device health data is a read operation, so the correct method is get. The required endpoint structure is the organization-level path organizations/{org_id}/devices/statuses, which returns the device status JSON that can then be archived to disk.

QUESTION NO: 25
展示を参照してください。エンジニアはCisco Meraki
APIを使用してネットワークENAUTOを消去する必要があります。
Python コードを完成させるには、コード内のボックスにどのコード
ス二ペットを追加する必要がありますか?

```
import requests

url = "https://api.meraki.com/api/v1/networks/ENAUTO/split"

payload = None

headers = {
    "Content-Type": "application/json",
    "Accept": "application/json",
    "": "",
    "X-Cisco-Meraki-API-Key": "6bec40cf957de430a6f1f2012345678a4fac9ea0"
}

[                                                                    ]

print(response.text.encode('utf8'))
```

A. response = requests.response('REMOVE', url, headers=headers, data = payload)
B. response = requests.requests('ERASE', url, headers=headers, data = payload)
C. response = requests.response('DELETE', url, headers=headers, data = payload)
D. response = requests.request('DELETE', url, headers=headers, data = payload)
Answer: D
Explanation:
To delete a Meraki network using the API, the correct HTTP method is DELETE, and the proper syntax with the requests library is:
response = requests.request('DELETE', url, headers=headers, data=payload) This line sends a DELETE request to the specified Meraki API endpoint to erase the network identified by "ENAUTO".

QUESTION NO: 26
展示を参照してください。エンジニアはCisco Meraki APIを使用してネットワークをENAUTOに分割する必要があります。Python コードを完成させるには、コード内のボックスにどのコードスニペットを追加する必要がありますか?

```
import requests

url = "https://api.meraki.com/api/v1/networks/ENAUTO/split"

payload = None

headers = {
    "Content-Type": "application/json",
    "Accept": "application/json",
    "": "",
    "X-Cisco-Meraki-API-Key": "6bec40cf957de430a6f1f2012345678a4fac9ea0"
}

[                                                                    ]

print(response.text.encode('utf8'))
```

A. レスポンス = リクエスト.リクエスト('POST', url, ヘッダー = ヘッダー, データ = ペイロード)
B. レスポンス = リクエスト.リクエスト('PATCH', url, ヘッダー = ヘッダー, データ = ペイロード)
C. レスポンス = リクエスト.リクエスト ('PUT', url, ヘッダー = ヘッダー, ペイロード = なし)
D. レスポンス = リクエスト.リクエスト ('GET', url, ヘッダー = ヘッダー, ペイロード = なし)
Answer: A
Explanation:
To perform a network split using the Cisco Meraki API, the correct method is a POST request.
The code must use:
response = requests.request('POST', url, headers=headers, data=payload) This correctly sends the API request with the appropriate headers and payload (even if None).