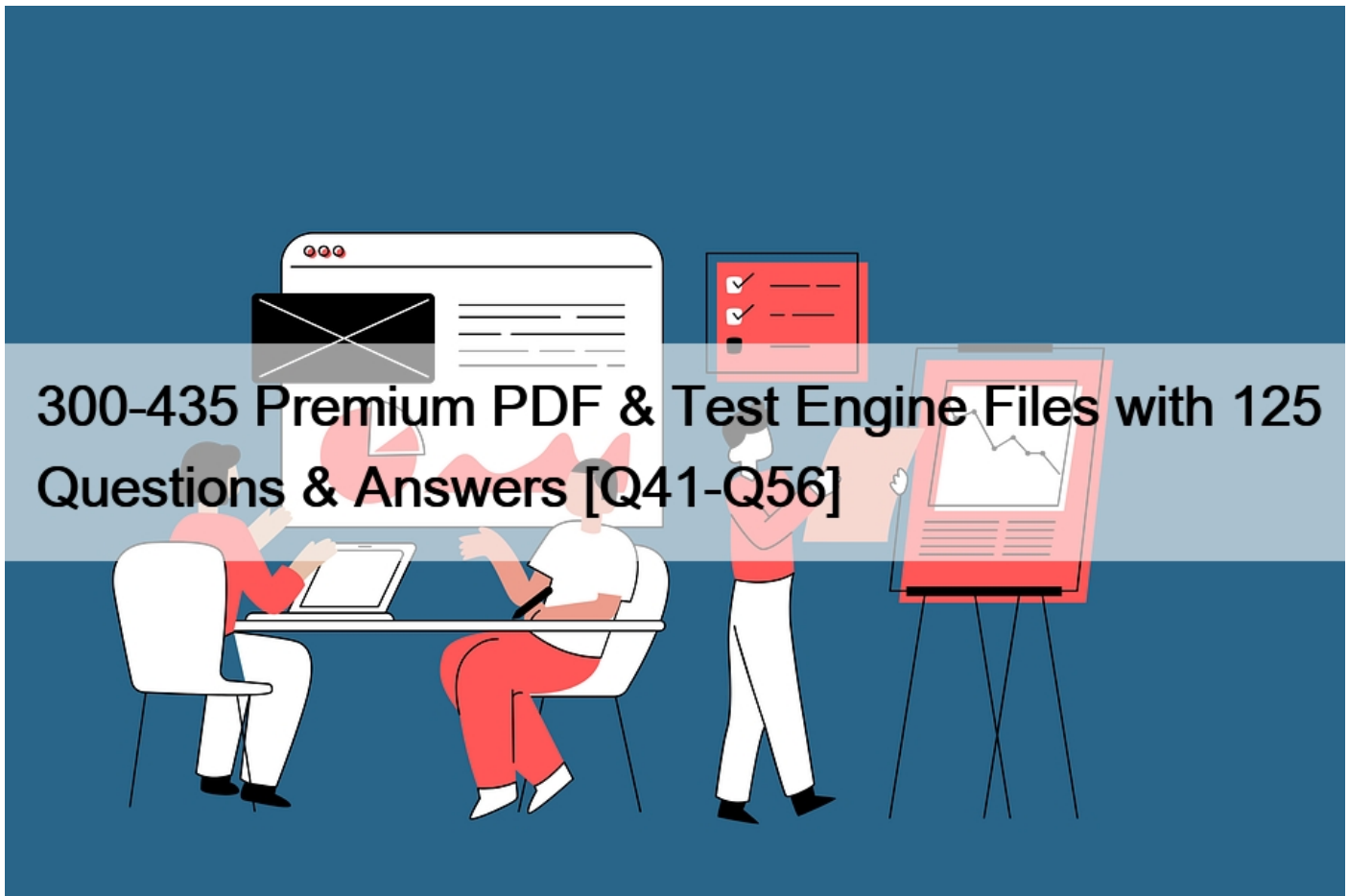


## 300-435 Premium PDF & Test Engine Files with 125 Questions & Answers [Q41-Q56]



**300-435 Premium PDF & Test Engine Files with 125 Questions & Answers Get 100% Real 300-435 Exam Questions, Accurate & Verified Answers As Seen in the Real Exam!**

Cisco 300-435 (Automating and Programming Cisco Enterprise Solutions) Certification Exam is an excellent opportunity for professionals to further their careers in the field of network automation and programming. It is a comprehensive exam that tests candidates' knowledge and skills in various areas related to automating and programming enterprise solutions using Cisco technologies. With the right preparation and experience, candidates can successfully pass the exam and obtain this valuable certification.

Cisco 300-435 exam is one of the highly sought-after certification exams that validate the candidates' skills in automating and programming Cisco enterprise solutions. 300-435 exam is a part of the CCNP Enterprise certification track and is designed to test the candidates' proficiency in using Cisco technologies to automate, program, and perform network configurations.

### NEW QUESTION 41

What does Cisco DNA Center use to manage third-party devices?

- \* command runners
- \* multivendor SDK
- \* templates
- \* device packages

### NEW QUESTION 42

A Cisco DNA Center script must be written to retrieve a list of interfaces on a switch. Drag and drop the API calls that are needed to return the list of interfaces using the Command Running APIs from the left into the correct sequence on the right.

Get task by ID.	run 1
Get file by ID.	run 2
Run read-only commands on devices.	run 3
Get device list.	run 4

Get task by ID.	Run read-only commands on devices.
Get file by ID.	Get device list.
Run read-only commands on devices.	Get file by ID.
Get device list.	Get task by ID.

Run read-only commands on devices.
Get device list.
Get file by ID.
Get task by ID.

### NEW QUESTION 43

Which statement describe the difference between OpenConfig and native YANG data models?

- \* Native models are designed to be independent of the underlying platform and are developed by vendors and standards bodies, such as the IETF.
- \* Native models are developed by individual developers and designed to apply configurations on platforms.
- \* OpenConfig models are developed by vendors and designed to integrate to features or configurations that are relevant only to that platform.
- \* Native models are developed by vendors and designed to integrate to features or configurations that are relevant only to that platform.

The difference between OpenConfig and native YANG data models lies in their development and purpose:

\* D: Native models are developed by vendors and are specific to their platforms. They are designed to integrate features or configurations that are relevant only to that vendor's platform. Native models, being vendor-specific, provide the most detailed and granular control over the vendor's devices.

OpenConfig models, on the other hand, are developed by a collaborative group of network operators and are intended to be vendor-neutral. They aim to provide a more standardized model that can be used across different vendors' equipment, focusing on the needs of the operators rather than the specific capabilities of the devices.

References:

- \* Cisco Blogs on YANG models1
- \* CBT Nuggets on Native YANG models2
- \* Introduction to YANG & LTRSDN-2260

### NEW QUESTION 44

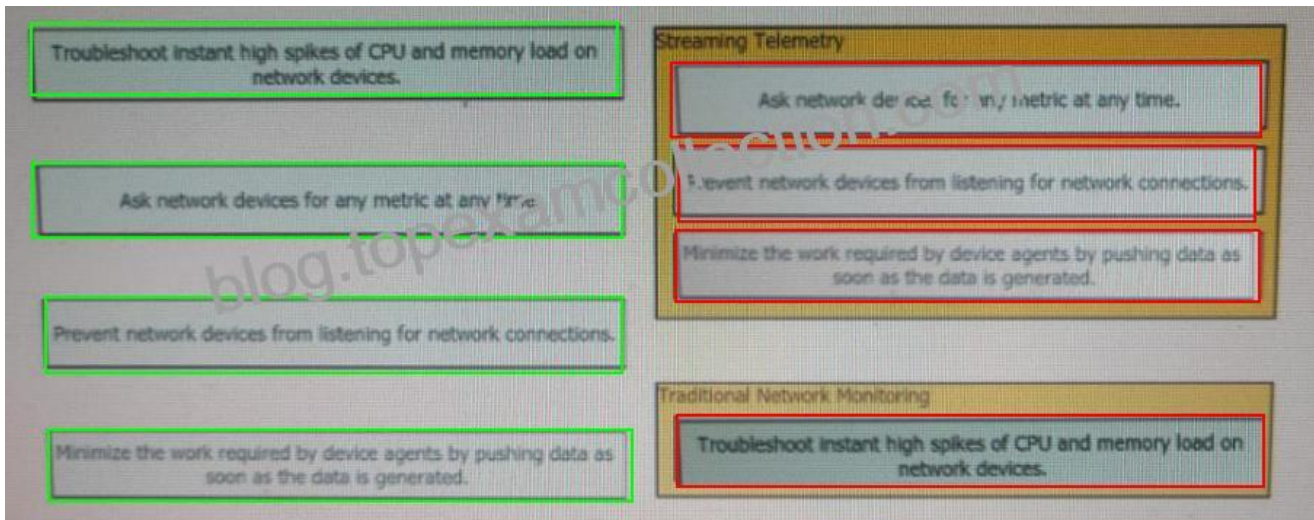
Drag and drop the characteristic from the left onto the monitoring type described on the right

The interface shows four characteristics on the left and two monitoring types on the right. The characteristics are:

- Troubleshoot instant high spikes of CPU and memory load on network devices.
- Ask network devices for any metric at any time.
- Prevent network devices from listening for network connections.
- Minimize the work required by device agents by pushing data as soon as the data is generated.

The monitoring types are:

- Streaming Telemetry (with three empty slots)
- Traditional Network Monitoring (with one empty slot)



Reference:

[https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/telemetry/70x/b-telemetry-cg-ncs5500-70x/b-telemetry-cg-ncs5500-70x\\_chapter\\_010.html](https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/telemetry/70x/b-telemetry-cg-ncs5500-70x/b-telemetry-cg-ncs5500-70x_chapter_010.html)

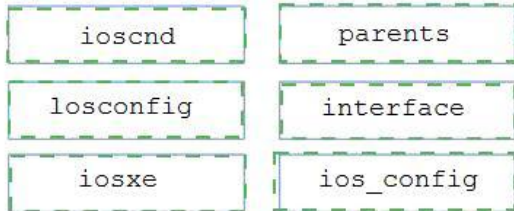
#### NEW QUESTION 45

Drag and drop the code from the bottom onto the box where the code is missing in the Ansible playbook to apply the configuration to an interface on a Cisco IOS XE device. Not all options are used.

```
- name: configure interface settings
  [ ] :
  lines:
    - ip address 172.31.1.1 255.255.255.0
    - no shutdown
  [ ] : interface GigabitEthernet1/0
```

- |           |            |
|-----------|------------|
| ioscmd    | parents    |
| iosconfig | interface  |
| iosxe     | ios_config |

```
- name: configure interface settings
  ios_config :
    lines:
      - ip address 172.31.1.1 255.255.255.0
      - no shutdown
  parents : interface GigabitEthernet1/0
```



Explanation:

Graphical user interface, text, application Description automatically generated

```
- name: configure interface settings
  ios_config :
    lines:
      - ip address 172.31.1.1 255.255.255.0
      - no shutdown
  parents : interface GigabitEthernet1/0
```

#### NEW QUESTION 46

Fill in the blanks to complete this API request against the Cisco SD-WAN vManage Statistics API, which specified a deviceId of 260faff9-2d31-4312-cf96-143b46db0211, a local-color of biz-internet, and a remote-color of gold.

<https://vmanage-ip-address:8443/dataservice/device/app-route/statistics?>  260faff9-2d31-4312-cf96-143b46db0211  biz-internet  gold

deviceId= , &local-color= , &remote-color=

Reference:

[https://sdwan-docs.cisco.com/Product\\_Documentation/Command\\_Reference/Command\\_Reference/vManage\\_REST\\_APIs/Real-Time\\_Monitoring\\_APIs/Application-Aware\\_Routing#Statistics](https://sdwan-docs.cisco.com/Product_Documentation/Command_Reference/Command_Reference/vManage_REST_APIs/Real-Time_Monitoring_APIs/Application-Aware_Routing#Statistics)

### NEW QUESTION 47

Which two API calls are used to trigger a device configuration sync in Cisco DNA Center? (Choose two.)

- \* PUT /dna/intent/api/v1/network-device
- \* PUT /dna/intent/api/v1/network-device/sync-all
- \* PUT /dna/intent/api/v1/network-device/{networkDeviceId}/sync
- \* PUT /dna/intent/api/v1/network-device/sync
- \* POST /dna/intent/api/v1/network-device/{networkDeviceId}/sync

### NEW QUESTION 48

FILL BLANK

Fill in the blank to complete the statement.

\_\_\_\_\_ is a solution for automating the configuration of a device when it is first powered on, using DHCP and TFTP.

Zerotouchprovisioning

Reference:

[https://www.cisco.com/c/en/us/td/docs/iosxml/ios/prog/configuration/169/b\\_169\\_programmability\\_cg/zero\\_touch\\_provisioning.html](https://www.cisco.com/c/en/us/td/docs/iosxml/ios/prog/configuration/169/b_169_programmability_cg/zero_touch_provisioning.html)

### NEW QUESTION 49

```
return_val=
{
  "alertId": "643451796765672516",
  "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 123456789",
  "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?

- \* `with return_val:`  
    `print("The Switch: "+deviceName+ ",`  
    `went down at: "+occurredAt)`

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

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

Section: Network Programmability Foundation

### NEW QUESTION 50

Refer to the exhibit.

```
{  
  "ietf-interfaces:interfaces": {  
    "interface": [  
      {  
        "name": "GigabitEthernet1",  
        "description": "MANAGEMENT INTERFACE",  
        "type": "iana-if-type:ethernetCsmacd",  
        "enabled": true,  
        "ietf-ip:ipv4": {  
          "address": [  
            {  
              "ip": "10.10.20.48",  
              "netmask": "255.255.255.0"  
            }  
          ]  
        },  
        "ietf-ip:ipv6": {}  
      }  
    ]  
  }  
}
```

A RESTCONF GET request is sent to a Cisco IOS XE device. A portion of the response is shown in the exhibit.

Which module name corresponds to the YANG model referenced in the request?

- \* ietf-interfaces:ietf-ipv4
- \* iana-if-type:ethernetCsmacd
- \* ietf-interfaces:interfaces
- \* ietf-interfaces

### NEW QUESTION 51

Refer to the exhibit.

```
module: ietf-ip
augment /if:interfaces/if:interface:
  +--rw ipv4!
  |   +--rw enabled?      boolean
  |   +--rw forwarding?  boolean
  |   +--rw mtu?          uint16
  |   +--rw address* [ip]
  |   |   +--rw ip          inet:ipv4-address-no-zone
  |   |   |   +--rw (subnet)
  |   |   |   |   +---:(prefix-length)
  |   |   |   |   |   +--rw prefix-length?      uint8
  |   |   |   |   |   +---:(netmask)
  |   |   |   |   |   +--rw netmask?          yang:dotted-guad (ipv4-non-contiguous-netmasks)?
  |   |   |   |   |   +--ro origin?          ip-address-origin
  |   |   +--rw neighbor* [ip]
  |   |   |   +--rw ip          inet:ipv4-address-no-zone
  |   |   |   +--rw link-layer-address  yang:phys-address
```

Which NETCONF statement type is represented by `+--rw address* [ip]`?

- \* list
- \* leaf-list
- \* container
- \* submodule

Symbols after data node names: `?&#8221;` means an optional node, `!&#8221;` means a presence container, and `*&#8221;` denotes a list and leaf-list.

## NEW QUESTION 52

```
GET: https://dnacsrvc/api/v1/network-device
{
  "response": [
    {
      "type": "Cisco Catalyst 9300 switch",
      "errorCode": null,
      "family": "Switches and Routers",
      "location": "DC1",
      "role": "ACCESS",
      "macAddress": "a1:2b:30:40:41:50",
      "hostname": "cat_9k_1",
      "serialNumber": "FCW2136LOAK",
      "softwareVersion": "16.6.1",
      "locationName": null,
      "upTime": "13 days, 18:30:33.81",
      "softwareType": "IOS-XE",
      "collectionStatus": "Managed",
      "managementIpAddress": "10.10.22.66",
      "platformId": "C9300-24UX",
      "reachabilityStatus": "Reachable",
      "series": "Cisco Catalyst 9300 Series Switches",
      "snmpContact": "",
      "snmpLocation": ""
    }
  ]
}
```

Refer to the exhibit. A GET request is issued to the Cisco DNA Center REST API. Drag and drop the GET request URL subpaths



from the left onto the objectives on the right. Not all options are used.

- /api/v1/network-device?softwareType=IOS-XE&softwareVersion=16.4.2
- /api/v1/network-device?location=DC2
- /api/v1/network-device?(softwareType=IOS-XE) AND (softwareVersion=16.4.2)
- /api/v1/network-device?family=Switches and Hubs
- /api/v1/network-device?ipAddress=10.222.10.35
- /api/v1/network-device?snmpLocation=DC2
- /api/v1/network-device?managementIpAddress=10.222.10.35
- /api/v1/network-device?family=cat\_9k\_1

- List devices that are configured by using SNMP to be in the DC2 location
- List device types
- List the device that has an IP address of 10.222.10.35
- Display Cisco IOS XE devices that have IOS version 16.4.2

- /api/v1/network-device?softwareType=IOS-XE&softwareVersion=16.4.2
- /api/v1/network-device?location=DC2
- /api/v1/network-device?(softwareType=IOS-XE) AND (softwareVersion=16.4.2)
- /api/v1/network-device?family=Switches and Hubs
- /api/v1/network-device?ipAddress=10.222.10.35
- /api/v1/network-device?snmpLocation=DC2
- /api/v1/network-device?managementIpAddress=10.222.10.35
- /api/v1/network-device?family=cat\_9k\_1

- /api/v1/network-device?location=DC2
- /api/v1/network-device?managementIpAddress=10.222.10.35
- /api/v1/network-device?ipAddress=10.222.10.35
- /api/v1/network-device?(softwareType=IOS-XE) AND (softwareVersion=16.4.2)

Reference:

[https://meraki.cisco.com/lib/pdf/meraki\\_whitepaper\\_captive\\_portal.pdf](https://meraki.cisco.com/lib/pdf/meraki_whitepaper_captive_portal.pdf)

### NEW QUESTION 53

What is primary purpose of using the Cisco SD-WAN vManage Certificate Management API?

- \* to securely deploy vManage
- \* to report an issue to Cisco TAC
- \* to install signed certificates
- \* to contact Enterprise Certificate Authority

Reference:

[https://sdwandocs.cisco.com/Product\\_Documentation/vManage\\_Help/Release\\_17.1/Configuration/Certificates](https://sdwandocs.cisco.com/Product_Documentation/vManage_Help/Release_17.1/Configuration/Certificates)

[https://sdwandocs.cisco.com/Product\\_Documentation/vManage\\_Help/Release\\_17.1/Configuration/Certificates](https://sdwandocs.cisco.com/Product_Documentation/vManage_Help/Release_17.1/Configuration/Certificates)

#### NEW QUESTION 54

Refer to the exhibit.

```
neighbors = ['s1', 's2', 's3']  
switch = {'hostname': 'nexus', 'os': '7.0.3', 'neighbors': neighbors}  
print(switch['neighbors'][1])
```

What is the result when running the Python scripts?

- \* s1
- \* s2
- \* s1, s2, s3
- \* s3

```
1 neighbors = ['s1', 's2', 's3']  
2 switch = {'hostname': 'nexus', 'os': '7.0.3', 'neighbors': neighbors}  
3 print(switch['neighbors'][1])
```

Execute Mode, Version, Inputs & Arguments

3.7.4  Int

CommandLine Arguments

Result

CPU Time: 0.02 sec(s), Memory: 7604 kilobyte(s)

s2

## NEW QUESTION 55

What is the purpose of using the Cisco SD-WAN vManage Certificate Management API?

- \* To securely deploy vManage
- \* To contact Enterprise certificate Authority
- \* To install signed certificates
- \* To report an issue to cisco TAC

The purpose of using the Cisco SD-WAN vManage Certificate Management API is to install signed certificates. This API facilitates secure communication within the SD-WAN infrastructure by allowing administrators to manage and deploy certificates that verify the identity of devices and services. References: ( Automating Cisco Enterprise Solutions Official Cert Guide )

## NEW QUESTION 56

A new project called `device_status`; must be stored in a central Git repository called `device_status`; with the first file named `device_status.py`;. The Git repository is created using the account `python_programmer`.

Which set of commands inserts the project into Git?

- A. `git init`  
`git add device_status.py`  
`git commit -m "Initial Revision"`  
`git remote add origin \`  
`https://git.cisco.com/python_programmer/device_status.git`  
`git push -u origin master`
- B. `git init`  
`git remote add origin \`  
`https://git.cisco.com/python_programmer/device_status.git`  
`git add device_status.py`  
`git pull`
- C. `git init`  
`git remote add origin \`  
`https://git.cisco.com/python_programmer/device_status.git`  
`git add device_status.py`  
`git commit -m "Initial Revision"`  
`git pull -u origin master`
- D. `git init`  
`git add device_status.py`  
`git remote add python_programmer/device_status`  
`git push`

- \* Option A
- \* Option B
- \* Option C
- \* Option D

To insert the project into Git using the account `python_programmer`, one would need to initialize the local directory as a Git repository, add files to it, commit those files with an appropriate message, add a remote repository URL pointing to where the repository should be pushed on GitHub under `python_programmer`'s account, and finally push the changes up to that remote repository. Option A follows this process correctly:

`git init`; initializes the local directory as a Git repository;  
`git add device_status.py`; stages changes;  
`git commit -m "Initial Revision"`; commits staged changes with message;  
`git remote add origin https://git.cisco.com/python_programmer/device_status.git`; adds new remote repo;  
`git push -u origin master`; pushes changes to the remote repository.

master&#8217; pushes committed changes up to GitHub. References: = Automating Cisco Enterprise Solutions Official Cert Guide

**300-435 Premium Files Practice Valid Exam Dumps Question:** <https://www.topexamcollection.com/300-435-vce-collection.html>

]