



350-901^{Q&As}

Developing Applications Using Cisco Core Platforms and APIs
(DEVCOR)

Pass Cisco 350-901 Exam with 100% Guarantee

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

<https://www.passapply.com/350-901.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

On a Cisco Catalyst 9300 Series Switch, the guest shell is being used to create a service within a container. Which change is needed to allow the service to have external access?

- A. Apply ip nat overload on VirtualPortGroup0.
- B. Apply ip nat inside on Interface VirtualPortGroup0.
- C. Apply ip nat outside on Interface VirtualPortGroup0.
- D. Apply ip nat inside on Interface GigabitEthernet1.

Correct Answer: B

QUESTION 2

Refer to the exhibit.

```
{
  "errors": [
    "API rate limit exceeded for organization"
  ]
}
```

An architect wrote an application to collect device information from the Cisco Meraki Dashboard API. Every time a network change occurs the application collects information and records new endpoint MAC addresses. The application stopped working after the locations and network equipment of a competitor were acquired. Which application approach must be applied to reduce latency and rate limiting?

- A. MOS scoring system before collecting information
- B. leaky faucet algorithm for fault categorizing
- C. error handling to check for reachability first
- D. webhooks to trigger updates

Correct Answer: C

QUESTION 3

DRAG DROP

Drag and drop the code from the bottom onto the box where the code is missing on the Ansible task to enable a VLAN on a Meraki MX Device, Not all options are used

Select and Place:



Answer Area

```
- name: Create combined network
meraki_network:
  auth_key: "{{ meraki_api_key }}"
  net_name: "{{ item }}"
  org_id: "{{ meraki_org_id }}"
  type:
    -switch
    -wireless
    -appliance
  timezone: Europe/London
  tags: staging, uk
  loop: "{{ network_ids }}"
  delegate_to: localhost
  register: result

- name: Enable VLAN support on MX
uri:
  url: "https://api.meraki.com/api/v0/networks/{{ item.data.id }}/vlansEnabledState"
  return_content: yes
  headers:
    X-Cisco-Meraki-API-Key: "{{ meraki_api_key }}"
  body:
    enabled: true
    follow_redirects: all
    status_code: 200
    body_format: json
  delegate_to: localhost
```

loop: "{{ result.results }}"	method: PUT
when: "{{ result.results }}"	method: PATCH
body: application/json	

Correct Answer:



Answer Area

```
- name: Create combined network
meraki_network:
  auth_key: "{{ meraki_api_key }}"
  net_name: "{{ item }}"
  org_id: "{{ meraki_org_id }}"
  type:
    -switch
    -wireless
    -appliance
  timezone: Europe/London
  tags: staging, uk
  loop: "{{ network_ids }}"
  delegate_to: localhost
  register: result

- name: Enable VLAN support on MX
uri:
  url: "https://api.meraki.com/api/v0/networks/{{ item.data.id }}/vlansEnabledState"
  return_content: yes
  headers:
    X-Cisco-Meraki-API-Key: "{{ meraki_api_key }}"
  body:
    enabled: true
    follow_redirects: all
    status_code: 200
    body_format: json
  method: PUT
delegate_to: localhost
loop: "{{ result.results }}"
```

```
when: "{{ result.results }}"
```

```
method: PATCH
```

```
body: application/json
```

Target Place 1: method : put

Target Place 2: loop : {{request.results}}

QUESTION 4

Refer to the exhibit.



```
response = requests.post(url)

backoff = 5
time.sleep(int(backoff))
response = requests.post(url)
while response.status_code != 200 and backoff < 80:
    backoff *= 2
    time.sleep(int(backoff))
    response = requests.post(url)
else:
    continue
```

An engineer needs to implement REST API error handling when a timeout or rate limit scenario is present. Which code snippet must be placed into the blank in the code to complete the API request?

- A.

```
if response.status_code == 429:
    wait = response.headers.get('Retry-After', 99)
    print(f'-> got {response.status_code} from {url}. retrying after {wait}s')
    time.sleep(int(wait))
    response = requests.post(url)
elif response.status_code == 408:
```
- B.

```
if response.status_code == 401:
    wait = response.headers.get('Retry-After', 99)
    print(f'-> got {response.status_code} from {url}. retrying after {wait}s')
    time.sleep(int(wait))
    response = requests.post(url)
elif response.status_code == 408:
```
- C.

```
if response.status_code == 408:
    wait = response.headers.get('Retry-After', 99)
    print(f'-> got {response.status_code} from {url}. retrying after {wait}s')
    time.sleep(int(wait))
    response = requests.post(url)
elif response.status_code == 429:
```
- D.

```
if response.status_code == 429:
    wait = response.headers.get('Retry-After', 99)
    print(f'-> got {response.status_code} from {url}. retrying after {wait}s')
    time.sleep(int(wait))
    response = requests.post(url)
elif response.status_code == 401:
```

A. Option A

B. Option B



C. Option C

D. Option D

Correct Answer: A

QUESTION 5

DRAG DROP

Refer to the exhibit above and click on the resource tabs in the top left corner to view resources to help with this question. Drag and drop the correct code snippets from the left onto the item numbers on the right that match the missing sections in the exhibit to complete the cURL script that will use RESTCONF to update an interface on a CISCO IOS XE device.

```
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-quad {ipv4-non-
contiguous-netmasks}?
        +--rw neighbor* [ip]
          +--rw ip              inet:ipv4-address-no-zone
          +--rw link-layer-address  yang:phys-address
    +--rw ipv6!
      +--rw enabled?          boolean
      +--rw forwarding?      boolean
      +--rw mtu?              uint32
      +--rw address* [ip]
        +--rw ip              inet:ipv6-address-no-zone
        +--rw prefix-length    uint8
      +--rw neighbor* [ip]
        +--rw ip              inet:ipv6-address-no-zone
        +--rw link-layer-address  yang:phys-address
      +--rw dup-addr-detect-transmits?  uint32
      +--rw autoconf
        +--rw create-global-addresses?  boolean
        +--rw create-temporary-addresses?  boolean {ipv6-
privacy-autoconf}?
        +--rw temporary-valid-lifetime?  uint32 {ipv6-privacy-
autoconf}?
```



```
module: ietf-interfaces
+--rw interfaces
|   +--rw interface* [name]
|   |   +--rw name                string
|   |   +--rw description?       string
|   |   +--rw type               identityref
|   |   +--rw enabled?           boolean
|   |   +--rw link-up-down-trap-enable? enumeration {if-mib}?
+--ro interfaces-state
|   +--ro interface* [name]
|   |   +--ro name                string
|   |   +--ro type               identityref
|   |   +--ro admin-status       enumeration {if-mib}?
|   |   +--ro oper-status        enumeration
|   |   +--ro last-change?       yang:date-and-time
|   |   +--ro if-index           int32 {if-mib}?
|   |   +--ro phys-address?      yang:phys-address
|   |   +--ro higher-layer-if*   interface-state-ref
|   |   +--ro lower-layer-if*   interface-state-ref
|   |   +--ro speed?             yang:gauge64
|   |   +--ro statistics
|   |   |   +--ro discontinuity-time yang:date-and-time
|   |   |   +--ro in-octets?         yang:counter64
|   |   |   +--ro in-unicast-pkts?   yang:counter64
|   |   |   +--ro in-broadcast-pkts? yang:counter64
|   |   |   +--ro in-multicast-pkts? yang:counter64
|   |   |   +--ro in-discards?      yang:counter32
|   |   |   +--ro in-errors?        yang:counter32
|   |   |   +--ro in-unknown-protos? yang:counter32
|   |   |   +--ro out-octets?       yang:counter64
|   |   |   +--ro out-unicast-pkts? yang:counter64
|   |   |   +--ro out-broadcast-pkts? yang:counter64
|   |   |   +--ro out-multicast-pkts? yang:counter64
|   |   |   +--ro out-discards?     yang:counter32
|   |   |   +--ro out-errors?       yang:counter32
```

```
curl --location --request PUT 'https://ios-xe-
mgmt.cisco.com:9443/restconf/data/<item 1>/<item 2>=GigabitEthernet2' \
--header 'Authorization: <item 3>' \
--header 'Accept: <item 4>' \
--header 'Content-Type: application/yang-data+json' \
--data-raw '{
  "ietf-interfaces:interface": {
    "<item 5>": "GigabitEthernet2",
    "description": "Configured by RESTCONF",
    "<item 6>": "iana-if-type:ethernetCsmacd",
    "enabled": true,
    "<item 7>": {
      "address": [
        {
          "<item 8>": "10.255.255.1",
          "<item 9>": "255.255.255.0"
        }
      ]
    }
  }
}'
```



Select and Place:

ietf-interface:interface	<item 1>
interface	<item 2>
Basic cm9vdDpjaXNjbzEyMw==	<item 3>
application/yang-data+json	<item 4>
name	<item 5>
type	<item 6>
left-ip:ipv4	<item 7>
ip	<item 8>
netmask	<item 9>

Correct Answer:



	ietf-interface:interface
	interface
	Basic cm9vdDpjaXNjbzEyMw==
	application/yang-data+json
	name
	type
	left-ip:ipv4
	ip
	netmask

[Latest 350-901 Dumps](#)

[350-901 Study Guide](#)

[350-901 Exam Questions](#)