

<https://www.leads4pass.com/350-901.html>

Vendor: Cisco

Exam Code: 350-901

Exam Name: Designing, Deploying, and Managing Network Automation Systems

Certification: Cisco Certified Automation Specialist - Core

Total Questions: 557 Q&A (View Details)

Updated on: Mar 17, 2026

Question 1:

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

Question 2:

DRAG DROP

Drag and drop the code from the bottom onto the box where the code is missing to configure a router that runs Cisco IOS XE by using RESTCONF. The API call is made to the management URL of the IOS XE device and on a registered port number. Errors that are caught during HTTP operation are registered. Not all options are used.

Select and Place:

```
import requests

HOST = ' [ ] .cisco.com'
PORT = [ ]
USER = 'root'
PASS = 'Cisc0123'
BASE = 'GigabitEthernet3'

url = "https://{h}:{p}/api/running/native/ip/route".format(h=HOST, p=PORT)
headers = {'content-type': 'application/vnd.yang.data+json',
          'accept': 'application/vnd.yang.data+json'}

try:
    result = [ ] (url, auth=(USER, PASS), data=data,
                  headers=headers, verify=not insecure)
except Exception:
    print(str(sys.exc_info()[0]))
    print(result.status_code, result.text)

if result.status_code == [ ] :
    print result.text

# something went wrong
print(result.status_code, result.text)
```

ios-xe-mgmt	401	requests.post
9443	201	requests.patch

Correct Answer:

```
import requests

HOST = ' [input: ios-xe-mgmt] .cisco.com'
PORT = [input: 9443]
USER = 'root'
PASS = 'Cisc0123'
BASE = 'GigabitEthernet3'

url = "https://{h}:{p}/api/running/native/ip/route".format(h=HOST, p=PORT)
headers = {'content-type': 'application/vnd.yang.data+json',
           'accept': 'application/vnd.yang.data+json'}

try:
    result = [input: requests.post] (url, auth=(USER, PASS), data=data,
                                     [input: ] headers=headers, verify=not insecure)
except Exception:
    print(str(sys.exc_info()[0]))
    print(result.status_code, result.text)

if result.status_code == [input: 201] :
    print result.text

# something went wrong
print(result.status_code, result.text)
```

[input:]	[input: 401]	[input:]
[input:]	[input:]	[input: requests.patch]

Question 3:

DRAG DROP

Drag and drop the code from the bottom onto the box where the code is missing to permit network traffic between 10.0.0.0/8 and all other networks on port 80 for a Cisco Nexus 9000 switch by using the Puppet module. Not all options are used.

Select and Place:

```

[ ] ( 'ipv4 devnet_acl 42':
ensure          => 'present',
src_addr       => [ ]
src_port       => 'any',
dst_addr       => 'any',
dst_port       => [ ]
established    => 'true',
log            => 'true',
precedence     => 'flash',
ttl            => '128',
remark         => 'DevNet Example',
action         => [ ]
proto         => 'tcp',
}

```

'permit',	'deny',	cisco_ace
'10.0.0.0/8',	cisco_nexus_ace	'eq 80',

Correct Answer:

```

cisco_ace { 'ipv4 devnet_acl 42':
  ensure => 'present',
  src_addr => '10.0.0.0/8',
  src_port => 'any',
  dst_addr => 'any',
  dst_port => 'eq 80',
  established => 'true',
  log => 'true',
  precedence => 'flash',
  ttl => '128',
  remark => 'DevNet Example',
  action => 'permit',
  proto => 'tcp',
}
'deny',
cisco_nexus_ace

```

Question 4:

Refer to the exhibit.

```

import requests
import time
import json

class Connection:
    def __init__(self, config):
        self._config = config
        self._session = None
        self._retries = 0
        self._MAX_RETRIES = 12

    def _setupSession(self):
        self._retries = 0
        if self._session is None:
            self._session = requests.Session()
        return

    def get(self, url, params=None):
        self._setupSession()
        resp = self._session.get(self._config.host + url, verify=False, params=params)
        if resp.status_code == 200:
            return json.loads(resp.content.decode('utf-8'))
            
        self._retries += 1
        exp_backoff = (2**(self._retries+3))/1000
        time.sleep(exp_backoff)
        self.get(url=url, params=params)
    return resp

```

A network engineer must integrate error handling for time-outs on network devices using the REST interface. Which line of code needs to be placed on the snippet where the code is missing to accomplish this task?

- A. elif resp.status_code == 429 or self._retries < self._MAX_RETRIES:
- B. elif resp.status_code == 404 or self._retries < self._MAX_RETRIES:
- C. elif resp.status_code == 429 and self._retries < self._MAX_RETRIES:
- D. elif resp.status_code == 404 and self._retries < self._MAX_RETRIES:

Correct Answer: C

Question 5:

DRAG DROP

Refer to the exhibit.

Handling Requests from Webex

When one of your webhooks is triggered by an event, Webex will send an HTTP POST to the backend `targetURL` that you've specified. The body of the POST will look something like this:

```
{
  "id": "Y2lzY29zcGFyazovL3VzL1dFQkHPT0svZjRlNjA1NjA1NjYwMjY0ZmZlLWREyNREtOTQ5O...",
  "name": "New message in 'Project Unicorn' room",
  "resource": "messages",
  "event": "created",
  "filter": "roomId=Y2lzY29zcGFyazovL3VzL1JPT00vYmJjZWl0YXQtdNDNwMS0zYjU4LTkxND...",
  "orgId": "OTZlYmMyYREtM2RjYyBxMmU1LWRExNTItZmZzNDgxdWkxYzlh",
  "createdBy": "Y2lzY29zcGFyazovL3VzL1BFT1BMRS9mNWlZnE4Ny1jOGRkLTQ3MjctOGIyZl...",
  "appId": "Y2lzY29zcGFyazovL3VzL1BFT1BMRS9mNWlZnE4Ny1jOGRkLTQ3MjctOGIyZl...",
  "ownedBy": "creator",
  "status": "active",
  "actorId": "Y2lzY29zcGFyazovL3VzL1BFT1BMRS9mNWlZnE4Ny1jOGRkLTQ3MjctOGIyZl...",
  "data": {
    "id": "Y2lzY29zcGFyazovL3VzL1BFT1BMRS9mNWlZnE4Ny1jOGRkLTQ3MjctOGIyZl...",
    "roomId": "Y2lzY29zcGFyazovL3VzL1JPT00vYmJjZWl0YXQtdNDNwMS0zYjU4LTkxNDctZjE...",
    "personId": "Y2lzY29zcGFyazovL3VzL1BFT1BMRS9mNWlZnE4Ny1jOGRkLTQ3MjctOGIyZl...",
    "personEmail": "y@example.com",
    "created": "2015-10-18T14:26:16.000Z"
  }
}
```

The first few properties shown above are called the "envelope." They identify all webhooks that are sent. This envelope contains the following information:

Parameter	Explanation
<code>id</code>	The webhook ID. This is the same ID returned when you created the webhook and is what you would use to view the webhook configuration or delete the webhook.
<code>id</code>	The webhook ID. This is the same ID returned when you created the webhook and is what you would use to view the webhook configuration or delete the webhook.

Get Message Details

Shows details for a message, by message ID. Specify the message ID in the `messageId` parameter in the URI.

GET `/v1/messages/{messageId}`

URI Parameters

messageId
string Required
The unique identifier for the message.

Response Properties

id
string
The unique identifier for the message.

parentId
string
The unique identifier for the parent message.

roomId
string
The room ID of the message.

Drag and drop the code from the bottom onto the box where the code is missing to complete the API request. An engineer is using this API request to implement Chat-Ops to generate notifications in a Webex space by using webhooks. Not all options are used.

Select and Place:

```

def process_incoming_message(  ):
    # Get the webhook data
    webhook_data = inbound_webhook_request.json

    # Determine the Teams Room to send reply to
    room_id = 

    # Get the details about the message that was sent.
    message_id = 
    message = teams.messages.get(message_id)

    # Verify message isn't from bot
    if  in teams.people.me().id:
        return ""

```

Correct Answer:

```
def process_incoming_message(  ):
    # Get the webhook data
    webhook_data = inbound_webhook_request.json

    # Determine the Teams Room to send reply to
    room_id = 
    # Get the details about the message that was sent.
    message_id = 
    message = teams.messages.get(message_id)

    # Verify message isn't from bot
    if  in teams.people.me().id:
        return ""
```

Question 6:

Which load balancing algorithm balances load based on the active sessions of a node?

- A. weighted round-robin
- B. IP source affinity
- C. least connections
- D. sticky session

Correct Answer: C

Question 7:

A web application is being developed to provide online sales to a retailer. The customers will need to use their username and passwords to login into their profile and complete their order. For this reason, the application must store user passwords.

Which approach ensures that an attacker will need to crack the passwords one at a time?

- A. Apply the peppering technique
- B. Store the passwords by using asymmetric encryption
- C. Apply the salting technique
- D. Store the passwords by using symmetric encryption

Correct Answer: C

Question 8:

Refer to the exhibit.

```
apiVersion: v1
clusters:
- cluster:
  certificate-authority: fake-ca-file
  server: https://1.2.3.4
  name: development
- cluster:
  insecure-skip-tls-verify: true
  server: https://5.6.7.8
  name: scratch
contexts:
- context:
  cluster: development
  namespace: frontend
  user: developer
  name: dev-frontend
- context:
  cluster: development
  namespace: storage
  user: developer
  name: dev-storage
- context:
  cluster: scratch
  namespace: default
  user: experimenter
  name: exp-scratch
current-context: ""
kind: Config
preferences: {}
users:
- name: developer
  user:
    client-certificate: fake-cert-file
    client-key: fake-key-file
- name: experimenter
  user:
    password: some-password
    username: exp
```

A kubeconfig file to manage access to clusters is provided. How many clusters are defined and which of them are accessed using username/password authentication versus certificate?

- A. two clusters; scratch
- B. three clusters; scratch
- C. three clusters; development
- D. two clusters; development

Correct Answer: A

Question 9:

DRAG DROP

Refer to the exhibit. A developer is creating a Python script by using Cisco DNA Center APIs. Drag and drop the code from the bottom onto the box where the code is missing in the Python script to retrieve and display wireless health information for each site. Not all options are used.

Operation Id: `getSiteHealth`

Description: *Returns Overall Health information for all sites*

```
GET /dna/intent/api/v1/site-health
```

Responses

Status: 200

The request was successful. The result is contained in the response body.

Schema Definition	Example Body
-------------------	--------------

- GetSiteHealthResponse
 - response: array[]
 - accessGoodCount: string
 - accessTotalCount: string
 - clientHealthWired: string
 - clientHealthWireless: object
 - clientIssueCount: object
 - clientNumberOfIssues: object
 - latitude: object
 - longitude: object
 - networkHealthAverage: object
 - networkHealthOthers: object
 - networkHealthWireless: object
 - networkNumberOfIssues: object
 - numberOfWirelessClients: object
 - wirelessGoodClients: object

Select and Place:

```

import requests

URL = 'https://cisco.dnatest.com:443/dna/intent/api/v1/site-health'
ACCESS_TOKEN = 'ABCD1234'

headers =
{'X-Auth-Token': [REDACTED]}
'Content-type': 'application/json;charset=utf-8')

response = requests.get(URL, params=params_data, headers=headers)

[REDACTED]

sites_response = response.json()['response']
for site in sites_response:
    [REDACTED]
else:
    print([REDACTED],
response.text)

```

`response.status_code`

`ACCESS_TOKEN`

`print(site['siteName'][0],
site['networkHealthWireless'])`

`if response.status_code == 200:`

`response.error`

`while response.code == 200:`

`print('{}{}'.format(site['siteName'],
site['networkHealthWireless']))`

Correct Answer:

```

import requests

URL = 'https://cisco.dnatest.com:443/dna/intent/api/vl/site-health'
ACCESS_TOKEN = 'ABCD1234'

headers =
{'X-Auth-Token': ACCESS_TOKEN
'Content-type': 'application/json;charset=utf-8'}

response = requests.get(URL, params=params_data, headers=headers)

if response.status_code == 200:
    sites_response = response.json ['response']
    for site in sites_response:
        print('{}{}'.format(site['siteName'],
        site['networkHealthWireless']))
else:
    print(response.status_code
response.text)

```

```
print(site['siteName'][0]
['networkHealthWireless'])
```

```
response.error
```

```
while response.code == 200:
```

Question 10:

A developer deploys a SQLite database in a Docker container. Single-use secret keys are generated each time a user accesses the database. The keys expire after 24 hours. Where should the keys be stored?

- A. Outside of the Docker container in the source code of applications that connect to the SQLite database.
- B. In a separate file inside the Docker container that runs the SQLite database.
- C. In an encrypted database table within the SQLite database.
- D. In a separate storage volume within the Docker container.

Correct Answer: D

Question 11:

FILL BLANK

A local Docker image has an image ID of 386231131. Fill in the blanks to complete the command in order to tag the image into the "cisco" repository with "version1 0\".

\$ docker tag

Correct Answer: See explanation below

```
$docker tag 386231131 cisco/386231131:version1.0
```

Question 12:

Which Puppet manifest needs to be used to configure an interface GigabitEthernet 0/1 on a Cisco IOS switch?

A.

```

ios_interface {
    name                => 'GigabitEthernet0/1',
    link_status         => false,
    logging_event       => [
        'spanning-tree',
        'subif-link-status'
    ],
    logging_event_link_status => false,
    ip_dhcp_snooping_trust => true,
    ip_dhcp_snooping_limit => 1500,
}

```

B.

```

ios_interface {
    'GigabitEthernet0/1' => {
        link_status         => false,
        logging_event       => [
            'spanning-tree',
            'subif-link-status'
        ],
        logging_event_link_status => false,
        ip_dhcp_snooping_trust => true,
        ip_dhcp_snooping_limit => 1500,
    }
}

```

C.

```

ios_interface {
    id                => 'GigabitEthernet0/1',
    link_status         => false,
    logging_event       => [
        'spanning-tree',
        'subif-link-status'
    ],
    logging-event_link_status => false,
    ip_dhcp_snooping_trust => true,
    ip_dhcp_snooping_limit => 1500,
}

```

D.

```

ios_interface { 'GigabitEthernet0/1':
    link_status         => false,
    logging_event       => [
        'spanning-tree',
        'subif-link-status'
    ],
    logging_event_link_status => false,
    ip_dhcp_snooping_trust => true,
    ip_dhcp_snooping_limit => 1500,
}

```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Correct Answer: D

https://github.com/puppetlabs/cisco_ios/blob/main/examples/network_interface.pp

Question 13:

Which data encoding format uses gRPC by default for serializing structured sets of information?

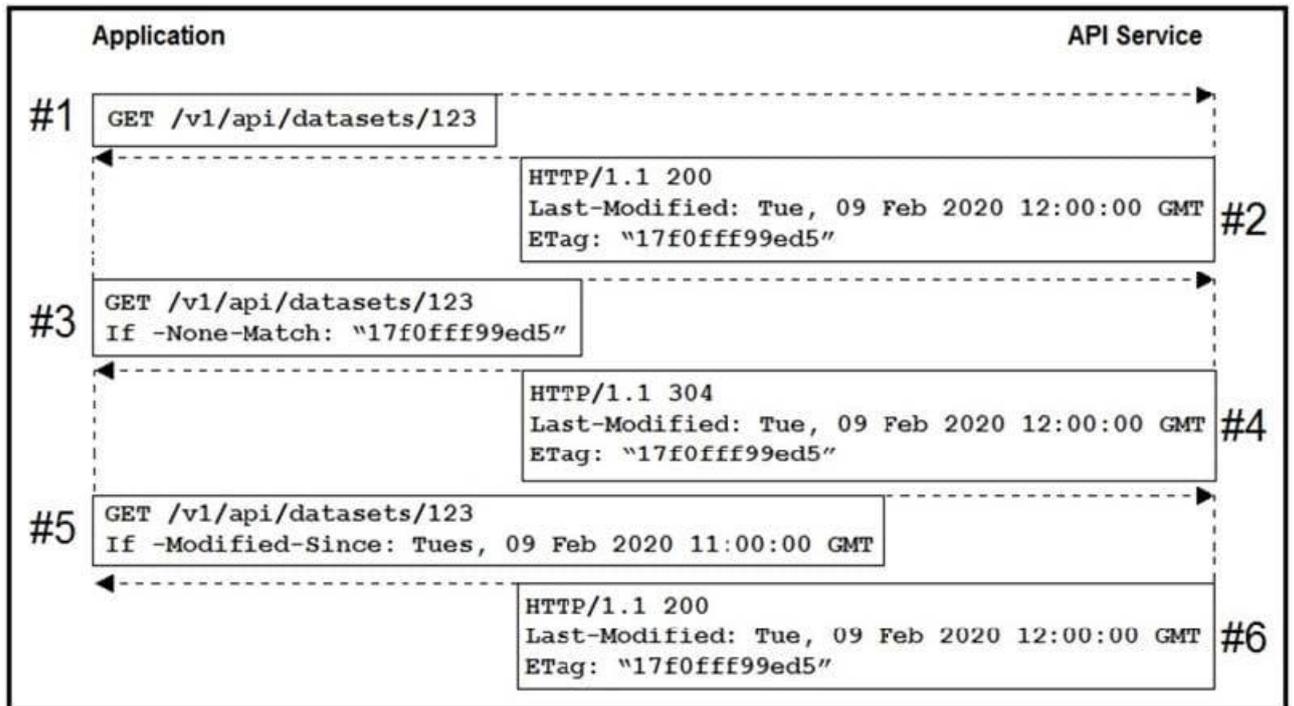
- A. JSON
- B. YAML
- C. Protobuf
- D. XML

Correct Answer: C

Explanation: The data encoding format used by gRPC by default is Protobuf.

Question 14:

Refer to the exhibit.



An application uses an API to periodically sync a large data set. Based on the HTTP message sequence provided, which statements are true about the caching behavior seen in the scenario? (Choose two.)

- A. The full dataset was transmitted to the client twice.
- B. The dataset changed sometime between message #4 and #5.
- C. A partial dataset was transmitted to the client in message #4.
- D. The dataset did not change during the scenario.
- E. Messages #3 and #5 are equivalent.

Correct Answer: AD

Question 15:
DRAG DROP

```

import request
import json
import sys

token = ""

def get_dnac_devices():
    <item 1>:
        url = "https://sandboxnac.cisco.com/dna/intent/api/v1/network-device"

        print(token)
        payload = {}
        headers = {
            'Content-Type': 'application/json',
            'Accept': 'application/json',
            'x-auth-token': token
        }

        response = requests.request("GET", url, headers=headers, data = payload)
        response.raise_for_status()
        return response.text

    <item 2>:
        print(e)
        if str(<item 3>) in str(e):
            create_dnac_token()

def create_dnac_token():
    try:
        url = "https://sandboxnac.cisco.com/dna/system/api/v1/auth/token"

        payload = {}
        headers = {
            <item 4>: 'Basic ZGV2bmV0dXNlcjpaXNjbzEyMyE=',
            'Content-Type': 'application/json'
        }

        response = requests.request("POST", url, headers=headers, data = payload)
        response.raise_for_status()
        return response.json()["Token"]
    except Exception as e:
        print(e)
        if str(<item 5>) in str(e):
            sys.exit("DNAC Service is not reachable")

if __name__ == "__main__":
    token = create_dnac_token()
    print(get_dnac_devices())

```

Refer to the exhibit. Drag and drop the code snippets from the left onto the item numbers on the right that match the missing sections in the exhibit to complete the script to implement control flow.

Select and Place:

Answer Area

- except Exception as e
- try
- Authorization
- request.status_codes.codes.SERVER_ERROR
- request.status_codes.codes.UNAUTHORIZED

- <item 1>
- <item 2>
- <item 3>
- <item 4>
- <item 5>

Correct Answer:

Answer Area

- try
- except Exception as e
- request.status_codes.codes.UNAUTHORIZED
- Authorization
- request.status_codes.codes.SERVER_ERROR