

➤ **Vendor:** Cisco

➤ **Exam Code:** 200-901

➤ **Exam Name:** Developing Applications and Automating Workflows using Cisco Core Platforms (DEVASC)

➤ **New Updated Questions from [Braindump2go](#) (Updated in [February/2022](#))**

[Visit Braindump2go and Download Full Version 200-901 Exam Dumps](#)

QUESTION 246

Refer to the exhibit. A network engineer wants to automate the port enable/disable process on specific Cisco switches. The engineer creates a script to send a request through RESTCONF and uses ietf as the YANG model and JSON as payload. Which command enables an interface named Loopback1?

```
1 def enable_function(if_name, if_status, if_type):
2     headers = {'Accept': 'application/yang-data+json',
3               'Content-Type': 'application/yang-data+json'}
4     payload = {
5         "ietf-interfaces:interface": {
6             "name": if_name,
7             "enabled": if_status,
8             "type": if_type,
9         }
10    }
11    base_url = 'https://192.168.1.1:8443'
12    restconf_url = '/restconf/data/ietf-interfaces:interfaces/interface'
13
14    res = requests.put(f'{base_url}{restconf_url}={if_name}',
15                      headers=headers, json=payload,
16                      auth=('cisco', 'secret'), verify=False)
```

- A. enable_function(Loopback1, true, 'iana-if-type:softwareLoopback')
- B. enable_function('iana-if-type:softwareLoopback', Loopback1, true,)
- C. def enable_function('iana-if-type:softwareLoopback', Loopback1, false,)
- D. def enable_function(Loopback1, true, 'iana-if-type:softwareLoopback')

Answer: D

QUESTION 247

An engineer must configure Cisco Nexus devices and wants to automate this workflow. The engineer enables the Cisco NX-OS REST API to configure the devices by using an Ansible playbook. Before running the code, which resource must be used to verify that the code works correctly?

- A. Cisco NX-OS SDK
- B. Cisco Learning Labs
- C. Cisco Code Exchange
- D. Cisco Modeling Labs

Answer: A

QUESTION 248

A new application is being developed with specific platform requirements. The platform and application deployment and the ongoing platform management must be fully automated. The application includes a large database component that has significant disk I/O requirements. Which application deployment type meets the requirements?

- A. Python virtual environment
- B. virtual machine
- C. bare metal
- D. container

Answer: B

QUESTION 249

Refer to the exhibit. A developer must check packages on web load balancers that use nginx packages. The developer updates the servers as required in Ansible inventory and prepares an Ansible playbook to automate this workflow on related servers. Which process is being automated by the Ansible playbook?

```
1 - hosts: servers
2   tasks:
3
4   - name: task1
5     apt:
6       name: apache2
7       state: absent
8
9   - name: task2
10    apt:
11      name: nginx
12      state: present
13
14   - name: task3
15    apt:
16      name: "*"
17      state: latest
```

- A. verifying that nginx is installed only if apache2 is unavailable on the system, and then updating all the other packages
- B. verifying, without installing, that apache2 is unavailable and that nginx is installed, and then updating the repository to the latest version
- C. uninstalling the apache2 server, installing the nginx server, and then updating all the packages to the latest version
- D. installing nginx, uninstalling apache2, and then verifying that the package repository is at the latest version

Answer: B

QUESTION 250

Refer to the exhibit. An engineer must check the admin rights of users on a database regularly and prepares the Python script to automate the process. The script connects to the database and runs a query. What is a security issue about the secrets in the code that relates to secret protection?

```
user_manager.py
1 import psycopg2
2 import json, sys
3
4 # Load config file
5 config = None
6 with open('config.json') as config_file:
7     config = json.load(config_file)
8 if not config:
9     print("Error loading configuration file.")
10    sys.exit(1)
11
12 connection = psycopg2.connect(
13     database="users",
14     host=config["DB_USER"], user=config["DB_USER"], password=config["DB_PWD"]
15 )
16 connection.set_session()
17
18 def has_admin_role(username):
19     with connection.cursor() as cursor:
20         query = 'SELECT admin_role FROM users WHERE username = {0}'
21         cursor.execute(query.format(username))
22         result = cursor.fetchone()
23     return result
24
25 has_admin_role("ops.shared")

config.json
1 {
2     "DB_USER": "pguser",
3     "DB_PWD": "@123Qwer%4",
4     "DB_HOST": "devnetdb.example.com"
5 }
```

- A. They must be stored in configuration files if there is a possibility of leakage.
- B. They must be Base64-encoded if stored in the user database.
- C. They must be encrypted if stored in the user database.
- D. They must be stored in configuration files if no authentication will be used.

Answer: D

QUESTION 251

Refer to the exhibit. A network engineer must collect information from the network using Cisco DNA Center APIs. The script must query for the devices with 'platformId' 'C9540-x' and then send commands to be executed. Which process is automated by the Python script?

[200-901 Exam Dumps](#) [200-901 Exam Questions](#) [200-901 PDF Dumps](#) [200-901 VCE Dumps](#)

<https://www.braindump2go.com/200-901.html>

```

1 import requests
2 import json
3
4 BASE_URL = "https://<IP Address>"
5 FIRST_URL = "/dna/intent/api/v1/network-device"
6 SECOND_URL = "/dna/intent/api/v1/network-device-poller/cli/read-request"
7 THIRD_URL = "/dna/intent/api/v1/task/{task_id}"
8 FOURTH_URL = "/dna/intent/api/v1/file/{file_id}"
9
10 headers = {"X-Auth-Token": "TOKEN", "Content-Type": "application/json"}
11
12 params = {"platformId": "C9500-40X"}
13 response = requests.get(BASE_URL + FIRST_URL, headers=headers, params=params)
14 devices = []
15 for device in response.json()["response"]:
16     devices.append(device["id"])
17
18     payload = {
19         "commands": ["show version", "show ip int brief"],
20         "deviceUuids": devices,
21         "timeout": 0,
22     }
23 response = requests.post(BASE_URL + SECOND_URL, data=json.dumps(payload),
24     . headers=headers)
25 task_id = response.json()["response"]["taskId"]
26 response = requests.get(BASE_URL + THIRD_URL.format(task_id=task_id),
27     . headers=headers)
28 progress_json = json.loads(response.json()["response"]["progress"])
29 file_id = progress_json("fileId")
30 response = requests.get(BASE_URL + FOURTH_URL.format(file_id=file_id),
31     . headers=headers)
32 file_json = response.json()
33 for cmd in file_json:
34     print(file_json[cmd]["commandResponses"]["SUCCESS"]["show ip int brief"])

```

- A. results gathering of the **show version** and **show ip int brief** commands
- B. output of **show ip int brief** command on devices of type C9500-40X
- C. execution information of the **show version** and **show ip int brief** commands
- D. list of executed commands on C9500-40X and the information of **show ip int brief** command

Answer: B

QUESTION 252

Refer to the exhibit. Which two files are being edited in the unified diff? (Choose two.)

```

diff --git a/ciscoaxl/axl.py b/ciscoaxl/axl.py
index bc7727e..d66ef4d 100644
--- a/ciscoaxl/axl.py
+++ b/ciscoaxl/axl.py
@@ -99,7 +99,7 @@ class axl(object):
     :return: result dictionary
     """
     try:
-         return self.client.executeSQLUpdate(queru)
+         return self.client.executeSQLUpdate(queru) ['return']
     except Fault as e:
         return e

diff --git a/setup.py b/setup.py
index df79ef4..1470b05 100644
--- a/setup.py
+++ b/setup.py
@@ -5,7 +5,7 @@ with open("README.md", "r") as fh:

     setup(
         name="ciscoaxl",
-         version="0.14",
+         version="0.141",
         author="User 1",
         author_email="user1@example.com",
         description="Cisco CUCM AXL Library. Simple to use.",

```

- A. README.md
- B. setup.py
- C. axl.py
- D. ciscoaxl.py
- E. index.py

Answer: BC