

➤ **Vendor: Cisco**

➤ **Exam Code: 300-635**

➤ **Exam Name: Automating and Programming Cisco Data Center Solutions (DCAUTO)**

➤ **New Updated Questions from [Braindump2go](#) (Updated in [Sep/2020](#))**

[Visit Braindump2go and Download Full Version 300-635 Exam Dumps](#)

**QUESTION 56**

Which two Cisco UCS components require drivers? (Choose two )

- A. FNIC
- B. BIOS
- C. CIMC
- D. board controller
- E. ENIC

**Answer: AE**

**QUESTION 57**

Which two HTTP methods are supported by the Cisco Nexus REST API? (Choose two )

- A. PUT
- B. POST
- C. DELETE
- D. UPDATE
- E. CONNECT

**Answer: BC**

**QUESTION 58**

Which action allows Docker daemon persistence during switchover on the Cisco Nexus 9500 Series Switches running Cisco NX-OS?

- A. Change the Docker configuration to include the live restore option.
- B. Copy the dockerpart file manually to the standby supervisor after performing the switchover.
- C. Copy the dockerpart file manually to the standby supervisor before performing the switchover.
- D. The system takes automatic action.

**Answer: C**

**QUESTION 59**

Using the NX-API CLI JSON-RPC interface, which two Python data structure and requests call create an SVI? (Choose two.)

- A. 

```
requests.post(url, data=json.dumps(payload), headers={'content-type': 'application/json-rpc'}, auth=(username, password))
```
- B. 

```
requests.post(url, data=json.dumps(payload), headers={'content-type': 'application/json'}, auth=(username, password))
```
- C. 

```
payload = {
    "jsonrpc": "2.0", "method": "cli_conf",
    "params": {
        "command": "conf t; interface vlan " + id,
        "version": 1},
    "id": 1
}
```
- D. 

```
payload = [
    {
        "jsonrpc": "2.0", "method": "cli",
        "params": {"cmd": "conf t", "version": 1},
        "id": 1
    },
    {
        "jsonrpc": "2.0", "method": "cli",
        "params": {"cmd": "interface vlan " + id, "version": 1},
        "id": 2
    }
]
```
- E. 

```
payload = {
    "jsonrpc": "2.0", "method": "cli_conf",
    "params": {"cmd": "interface vlan " + id, "version": 1},
    "id": 1
}
```

**Answer:** AD

#### QUESTION 60

Refer to the exhibit. The exhibit shows a Cisco NX-OS switch configuration, an Ansible playbook, and the output of running this playbook. The playbook failed due to error "msg" 'Request failed <urlopen error [Errno 61] Connection refused>', 'status' -1, "url" "http://192.168.251.129:80/ins".

Switch configuration	Ansible playbook
<pre>!Command: show running-config ! feature hsrp ! ip access-list allow_http_traffic  10 permit tcp any any eq www ! vrf context management  ip route 0.0.0.0/0 192.168.151.2 ! interface mgmt0  ip address 192.168.251.129 255.255.255.0  vrf member management</pre>	<pre>--- - name: Vlan Provisioning   hosts: nxos   gather_facts: no    vars:     nxos_provider:       username: "{{ un }}"       password: "{{ pwd }}"       transport: nxapi       host: "{{ inventory_hostname }}"    tasks:      - name: CREATE VLANS AND ASSIGN A NAME, USING VLAN_ID       nxos_vlan:         vlan_id: "{{ item.vlan_id }}"         name: "{{ item.name }}"         provider: "{{ nxos_provider }}"       with_items:         - vlan_id: 2           name: Native         - vlan_id: 15           name: Web         - vlan_id: 20           name: App         - vlan_id: 30           name: DB</pre>
<p><b>Playbook output</b></p> <pre>\$ ansible-playbook playbook.yml  PLAY [Vlan Provisioning] ***** *****  TASK [CREATE VLANS AND ASSIGN A NAME, USING VLAD_ID]***** ***** failed: [192.168.252.129] (item={'vlan_id': 2, 'name': 'Native'}) =&gt; {"ansible_facts": {'discovered_interpreter_python': "/usr/bin/python"}, "ansible_loop_var": "item", 'changed': false, 'item': 'name': "Native", "vlan_id": 2}, "msg": "Request failed: &lt;urlopen error [Errno 61] Connection 'refused'&gt;" "status": -1, "url": "http://192.168.251.129:80/ins"}</pre>	

Which Cisco NX-OS configuration command resolves this failure?

- A. feature nxapi
- B. http-server enabled
- C. interface mgmt0; ip access-group allow\_http\_traffic in
- D. feature http

**Answer: A**