- **Vendor:** Cisco
- **Exam Code:** 300-635
- **Exam Name:** Automating and Programming Cisco Data Center Solutions (DCAUTO)
- **New Updated Questions from Braindump2go (Updated in April/2020)**

QUESTION 1
What is the network bootstrap program used by Cisco NX-OS iPXE?

A. NETBOOT
B. NX-OS iPXE
C. iPXE-POAP
D. Mini-OS

Answer: A

QUESTION 2
What is a description of a Cisco UCS Director script module?

A. function to convert internal workflow tasks into Python scripts
B. place to store custom workflow scripts, jars, and custom lists of values for use in custom workflow tasks
C. place to store external scripts that are not related to Cisco UCS Director
D. place to store imported scripts, Bash, and custom Python code for use in custom workflow tasks

Answer: B

QUESTION 3
Which procedure accesses the REST API browser within Cisco UCS Director?

A. Send an HTTP GET request to https://[UCS Director IP]/api/get_resources/.
B. Log in as the user REST/user to access the REST API interface.
C. Enable the Developer menu. Select Orchestration in the UI, then select the REST API browser tab.
D. Select the API browser from the Cisco UCS Director End User Portal catalog of services.

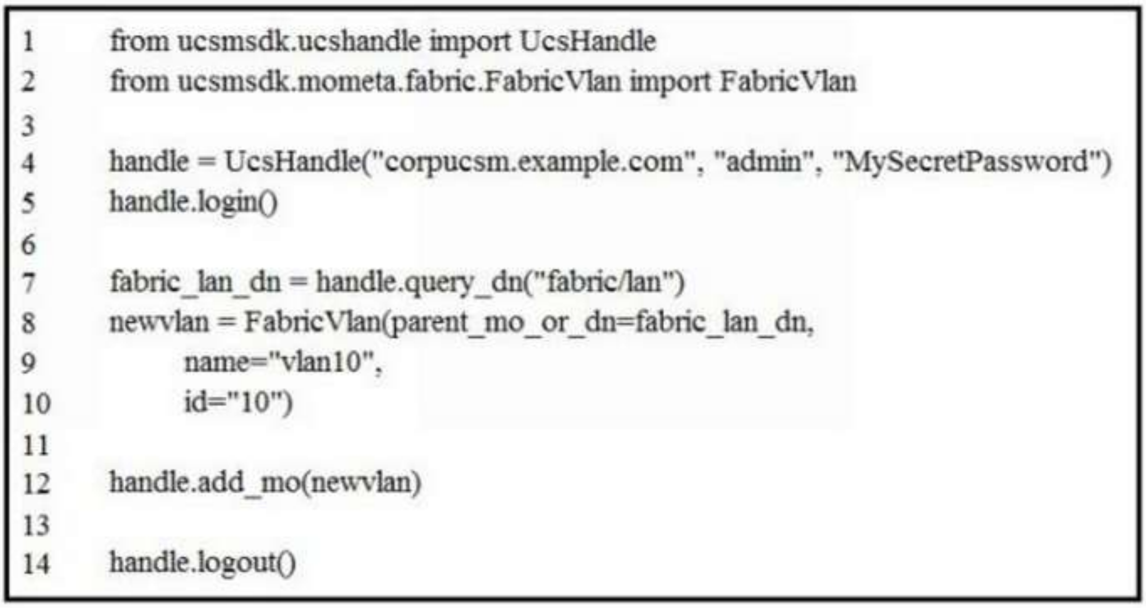
Answer: C

QUESTION 4
Which step must be taken to enable the REST API browser within Cisco UCS Director?

A. Edit the user profile and enable developer options.
B. Raise a case with TAC.
C. The REST API browser is automatically enabled in Cisco UCS Director when a Power User is created.
D. Log in as the user "REST".

Answer: A

QUESTION 5
Refer to the exhibit. Which change allows the code to configure VLAN 10 in the Cisco UCS?



```
1       from ucsmsdk.ucshandle import UcsHandle
2       from ucsmsdk.mometa.fabric.FabricVlan import FabricVlan
3
4       handle = UcsHandle("corpucsm.example.com", "admin", "MySecretPassword")
5       handle.login()
6
7       fabric_lan_dn = handle.query_dn("fabric/lan")
8       newvlan = FabricVlan(parent_mo_or_dn=fabric_lan_dn,
9               name="vlan10",
10              id="10")
11
12      handle.add_mo(newvlan)
13
14      handle.logout()
```

A. Lines 8 and 9 should have a line continuation \ at the end.
B. Line 13 should include "handle.commit()".
C. Line 4 should include transport 443 option.
D. Line 3 should add an import for query_dn.

Answer: C

QUESTION 6
Which programming language are the Cisco UCS Director custom workflow tasks written in?

A. C++
B. Python
C. Java
D. Cloupia Script

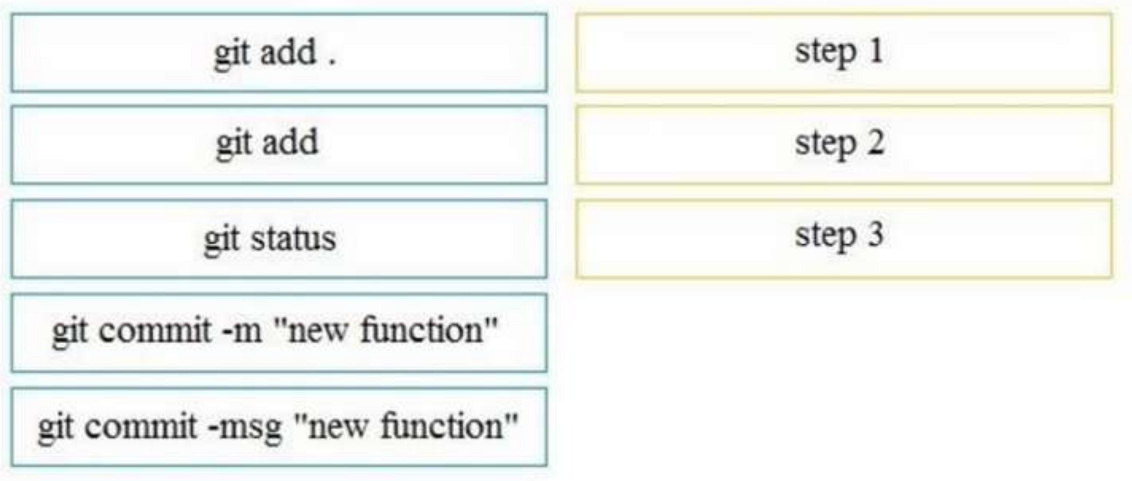
Answer: C

QUESTION 7
Which authentication method is used when the REST API of the Cisco UCS Director is accessed?

A. Bearer ((Bearer Token))
B. HTTP Basic Auth
C. RestAuth: ((User's Auth Token))
D. X-Cloupia-Request-Key: ((User's Auth Token))

Answer: B

QUESTION 8
Drag and Drop Question
A file named myfunc.py has been edited. Drag and drop the steps from the left that ensure that this file is committed to the local Git repository and verify its status into the correct order on the right. Not all options are used.



git add .

git add

git status

git commit -m "new function"

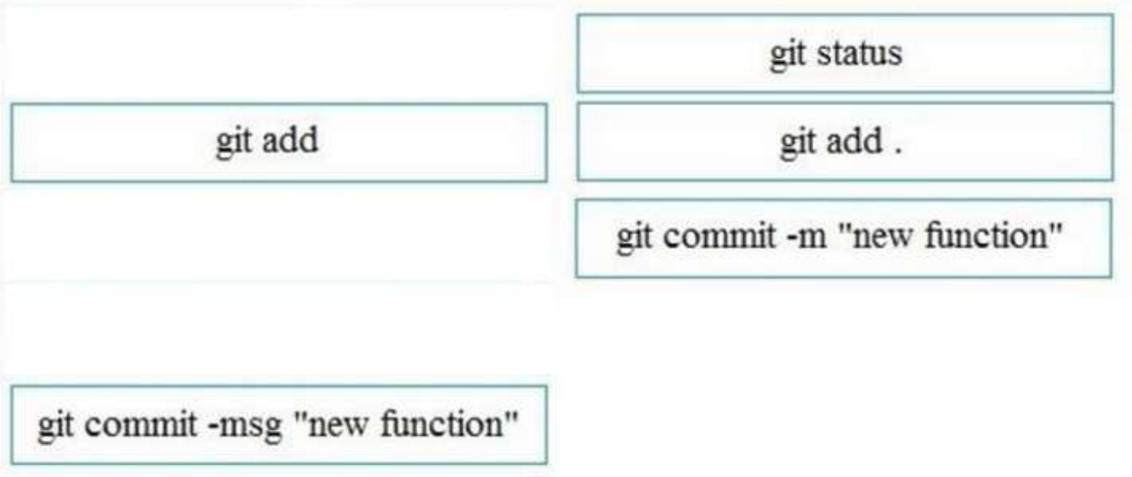
git commit -msg "new function"

step 1

step 2

step 3

Answer:



git add

git commit -msg "new function"

git status

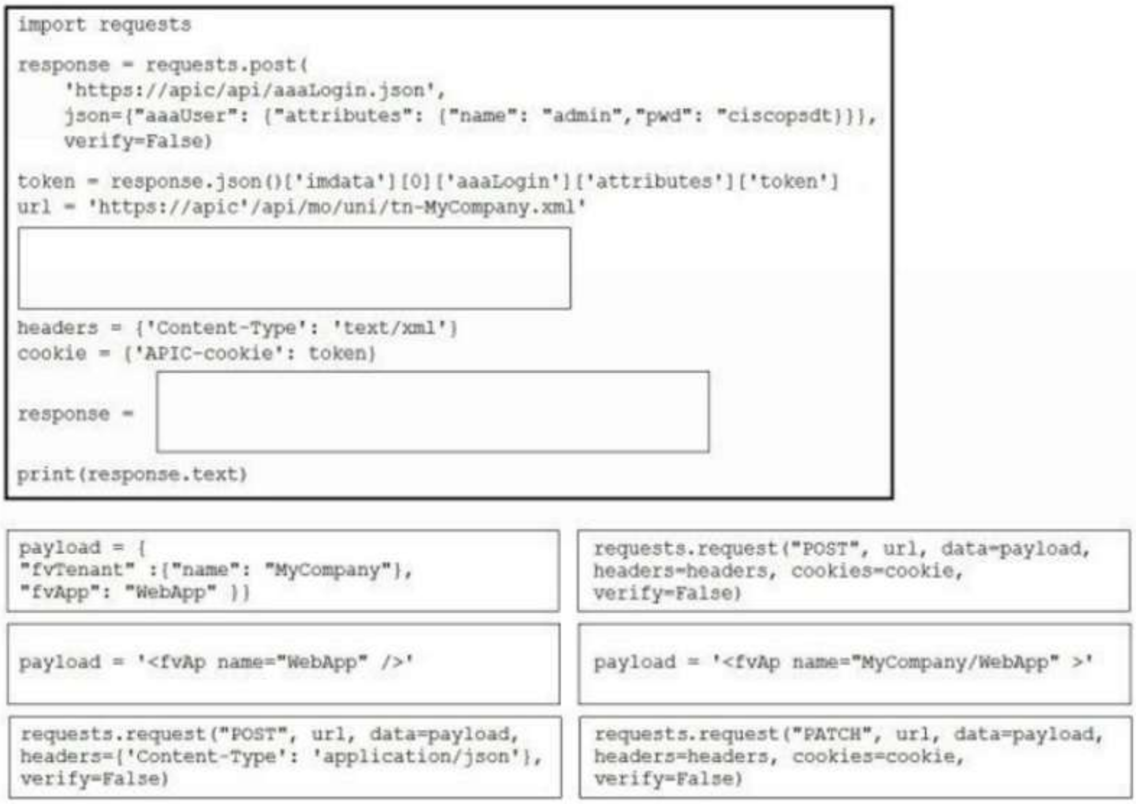
git add .

git commit -m "new function"

QUESTION 9

Drag and Drop Question

Drag and drop the correct code snippets into the Python code to create a new application profile "WebApp" using the ACI REST API. Not all options are used.



```
import requests

response = requests.post(
    'https://apic/api/aaaLogin.json',
    json={"aaaUser": {"attributes": {"name": "admin","pwd": "ciscopsdt"}}},
    verify=False)

token = response.json()['imdata'][0]['aaaLogin']['attributes']['token']
url = 'https://apic'/api/mo/uni/tn-MyCompany.xml'

headers = {'Content-Type': 'text/xml'}
cookie = {'APIC-cookie': token}

response = 

print(response.text)
```

```
payload = {
"fvTenant" :{"name": "MyCompany"},
"fvApp": "WebApp" }}
```

```
requests.request("POST", url, data=payload,
headers=headers, cookies=cookie,
verify=False)
```

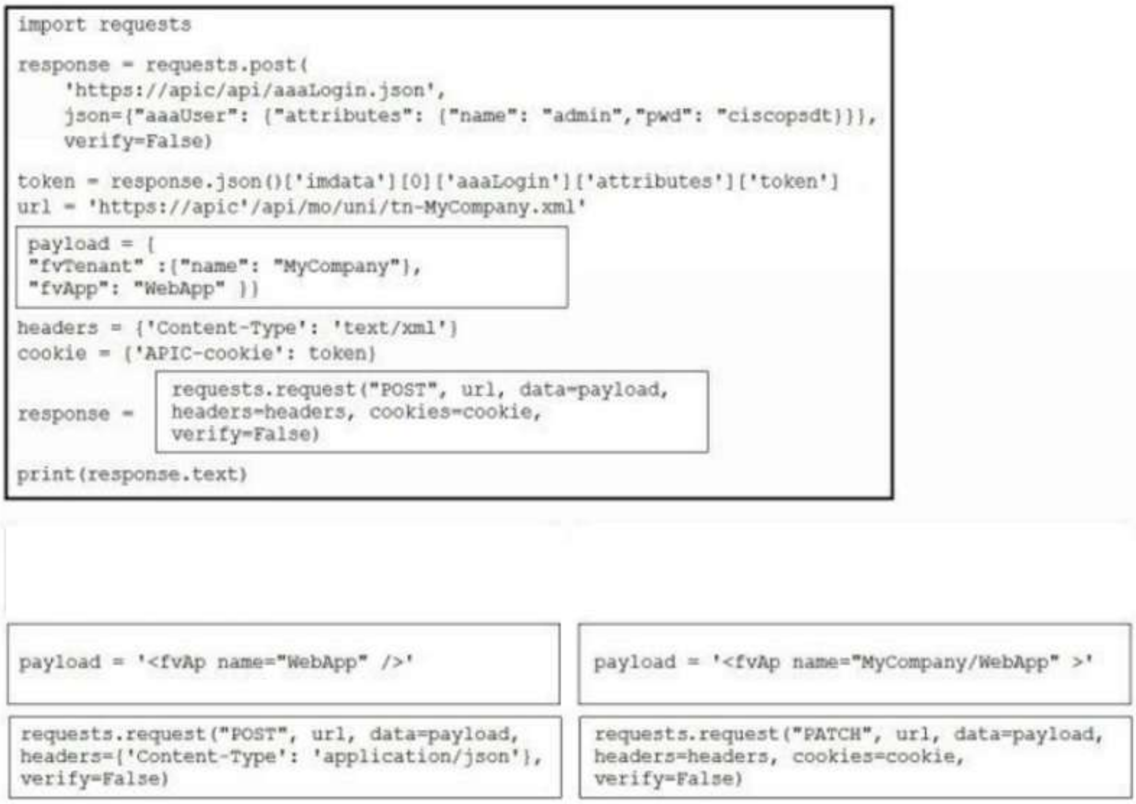
```
payload = '<fvAp name="WebApp" />'
```

```
payload = '<fvAp name="MyCompany/WebApp" >'
```

```
requests.request("POST", url, data=payload,
headers={'Content-Type': 'application/json'},
verify=False)
```

```
requests.request("PATCH", url, data=payload,
headers=headers, cookies=cookie,
verify=False)
```

Answer:



```
import requests

response = requests.post(
    'https://apic/api/aaaLogin.json',
    json={"aaaUser": {"attributes": {"name": "admin","pwd": "ciscopsdt"}}},
    verify=False)

token = response.json()['imdata'][0]['aaaLogin']['attributes']['token']
url = 'https://apic'/api/mo/uni/tn-MyCompany.xml'
 payload = {
 "fvTenant" :{"name": "MyCompany"},
 "fvApp": "WebApp" }}
headers = {'Content-Type': 'text/xml'}
cookie = {'APIC-cookie': token}

response = requests.request("POST", url, data=payload,
           headers=headers, cookies=cookie,
           verify=False)

print(response.text)
```

```
payload = '<fvAp name="WebApp" />'
```

```
payload = '<fvAp name="MyCompany/WebApp" >'
```

```
requests.request("POST", url, data=payload,
headers={'Content-Type': 'application/json'},
verify=False)
```

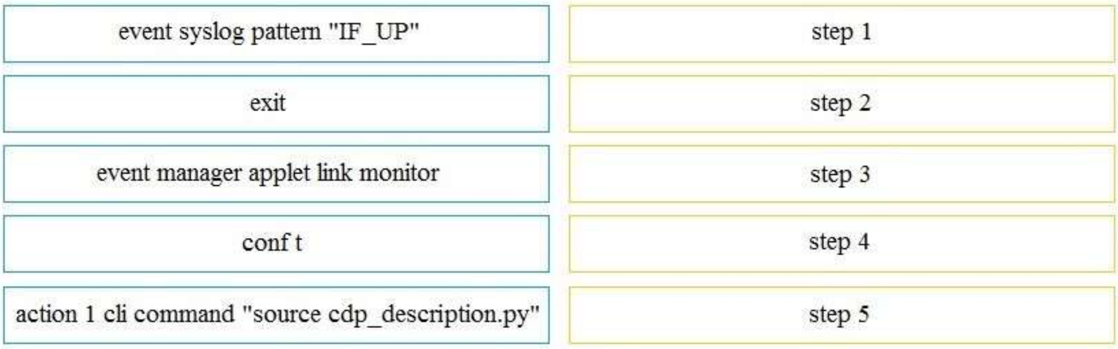
```
requests.request("PATCH", url, data=payload,
headers=headers, cookies=cookie,
verify=False)
```

QUESTION 10

Drag and Drop Question

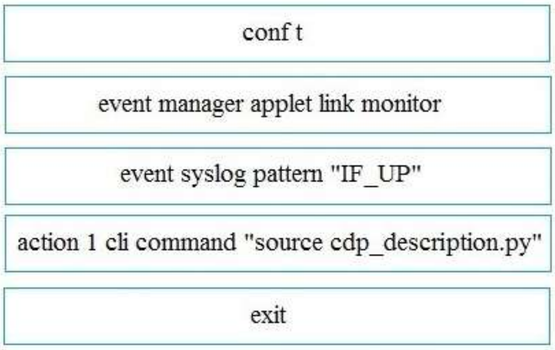
After a Cisco Nexus switch interface is enabled and in the up state, an engineer must automate the configuration of the interface descriptions using the EEM Python Module.

Drag and drop the steps the engineer must take from the left into the correct order on the right.



event syslog pattern "IF_UP"	step 1
exit	step 2
event manager applet link monitor	step 3
conf t	step 4
action 1 cli command "source cdp_description.py"	step 5

Answer:



conf t

event manager applet link monitor

event syslog pattern "IF_UP"

action 1 cli command "source cdp_description.py"

exit

Explanation:
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/datamodels/configuration/xe-16/data-models-xe-16-book/python-scripting-for-eem.html