

300-635 Demo

Question: 1

Which two benefits of using network configuration tools such as Ansible and Puppet to automate data center platforms are valid? (Choose two)

- A. consistency of systems configuration
- B. automation of repetitive tasks
- C. ability to create device and interface groups
- D. ability to add VLANs and routes per device
- E. removal of network protocols such as Spanning Tree

Answer: AB

Question: 2

DRAG DROP

Drag and drop the code to complete an Ansible playbook that creates a new tenant. Not all options are used.

```
- name: Add a new tenant
[ ]
  host: apic
  username: admin
  password: SomeSecretPassword
  [ ]
  description: MyCompany tenant
  [ ]
```

```
Tenant_name: MyCompany
```

```
state: absent
```

```
state: query
```

```
tenant: MyCompany
```

```
aci_tenant:
```

```
state: present
```

```
state: create
```

```
aci_tenant_name:
```

Answer:

```

- name: Add a new tenant
  aci_tenant:
    host: apic
    username: admin
    password: SomeSecretPassword
    tenant: MyCompany
    description: MyCompany tenant
    state: present
    
```

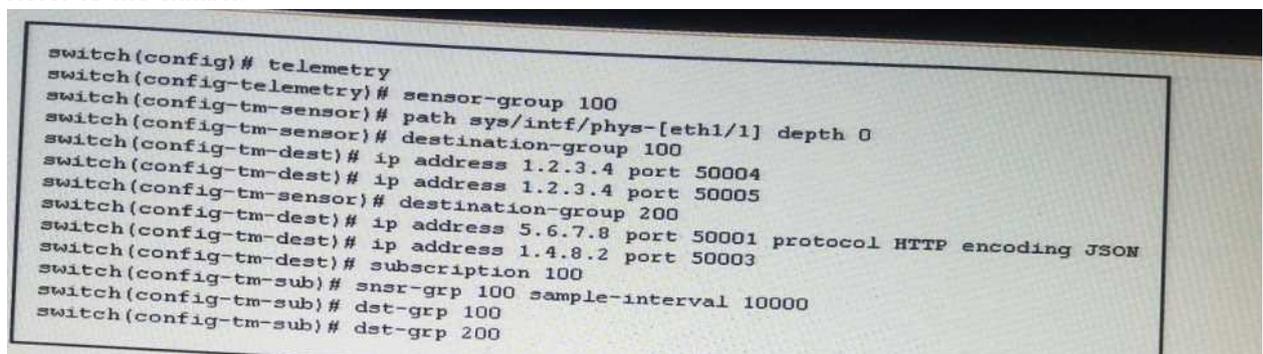
Tenant_name: MyCompany	state: absent
state: query	tenant: MyCompany
aci_tenant:	state: present
state: create	aci_tenant_name:

aci_tenant
tenant : MyCompany
state: present

Reference: https://docs.ansible.com/ansible/latest/scenario_guides/guide_aci.html

Question: 3

Refer to the exhibit:



Refer to the exhibit, Where and how often does the subscription stream data for Ethernet port 1/1?

- A. to four different destinations every 10000 microseconds
- B. to four different destinations every 100 milliseconds

- C. to four different destinations every 10 seconds
- D. to four different destinations every 10000 seconds

Answer: C

Reference:

https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus3000/sw/programmability/7_x/b_Cisco_Nexus_3000_Series_NX-OS_Programmability_Guide_7x/b_Cisco_Nexus_3000_Series_NX-OS_Programmability_Guide_7x_chapter_011101.pdf

Question: 4

Refer to the exhibit

```
mo_dir = cobra.mit.access.MoDirectory(cobra.mit.session.LoginSession(apic_url, username, password))
mo_dir.login()
cq = cobra.mit.access.ClassQuery('fvCEP')
cq.subtree = 'full'
objlist = mo_dir.query(cq)
for mo in objlist:
    print "MAC: " + mo.mac + "|" + "IP: " + mo.ip
```

Which action does the execution of this ACI Cobra Python code perform?

- A. It prints all LLDP neighbor MAC and IP addresses
- B. It prints all Cisco Discovery Protocol neighbor MAC and IP addresses
- C. It prints all endpoint MAC and IP addresses
- D. It prints all APIC MAC and IP addresses

Answer: C

Reference: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/Operating_ACI/guide/b_Cisco_Operating_ACI/b_Cisco_Operating_ACI_appendix_011.html

Question: 5

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
- 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

Reference: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/ucs-director/orchestration-guide/6-0/b_UCS_Director_Orchestration_Guide_6_0/b_UCS_Director_Orchestration_Guide_6_0_chapter_0

[1010.html](#)

Question: 6

Refer to the exhibit:

```

Switch configuration
!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

Ansible playbook
---
- 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

Playbook output
$ ansible-playbook playbook.yml

PLAY [Vlan Provisioning] *****
*****
TASK [CREATE VLANS AND ASSIGN A NAME, USING VLAN_ID] *****
*****
failed: [192.168.251.129] (item={'vlan_id': 2, 'name': 'Native'}) => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"}, "ansible_loop_var": "item", "changed": false, "item": {"vlan_id": 2, "name": "Native"}, "msg": "Request failed: <urlopen error [Errno 61] Connection refused>", "status": -1, "url": "http://192.168.251.129:80/ins"}

```

The exhibit shows a Cisco NX-OS switch configuration and 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>" 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: C

Reference:

https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/configuration/nxos/413/b_Copy_of_b_Cisco_Nexus_5000_Series_NX-OS_Software_Configuration_Guide/Copy_of_b_Cisco_Nexus_5000_Series_NX-OS_Software_Configuration_Guide_chapter22.pdf

Question: 7

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/](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
- D. Select the API browser from the Cisco UCS Director End User Portal catalog of services

Answer: C

Reference: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/ucs-director/rest-api-getting-started-guide/6-5/cisco-ucs-director-REST-API-getting-started-65.html#task_CE85B54B1DB64855BB3BECCD24C31F5B