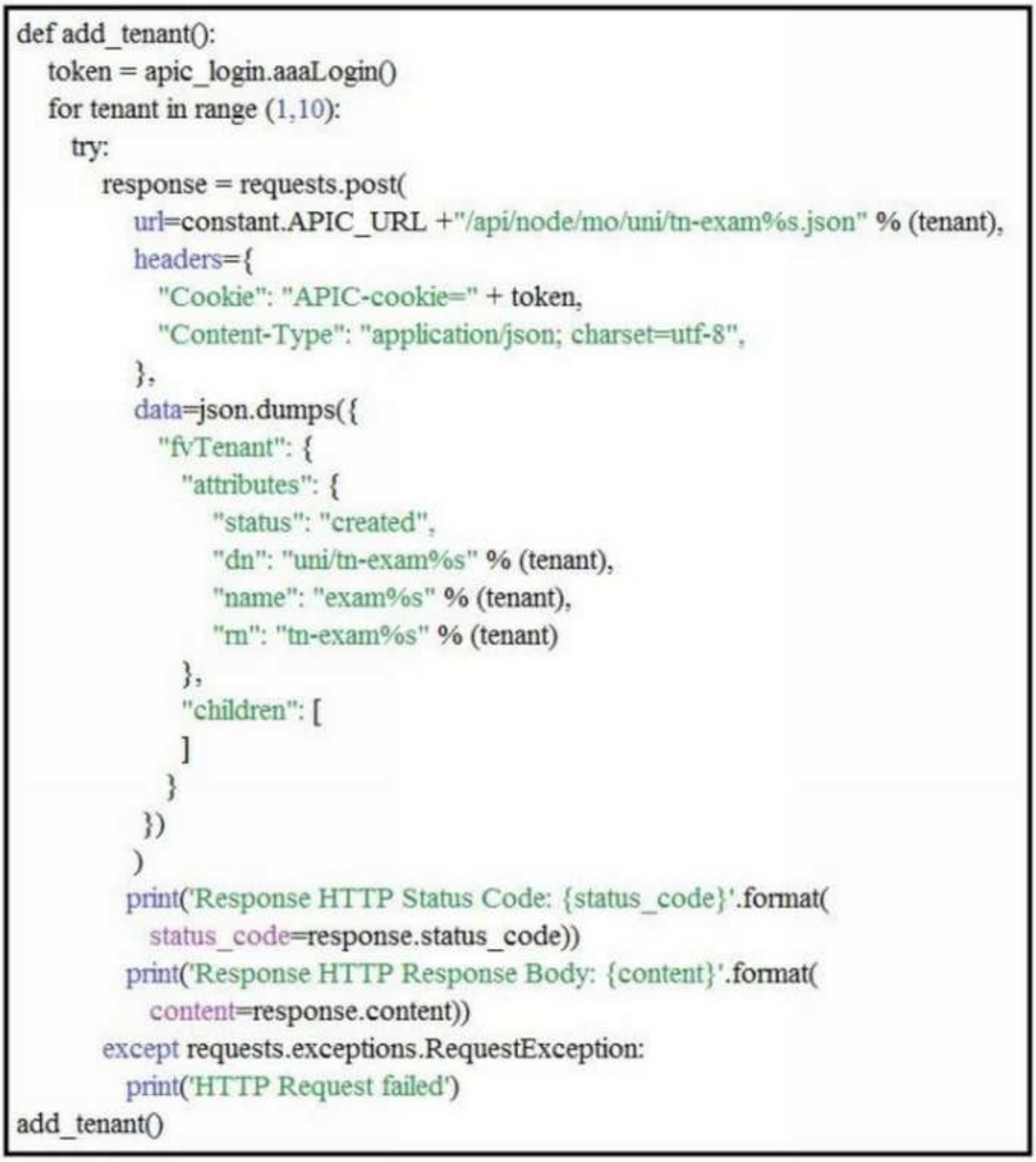
- **Vendor:** Cisco
- **Exam Code:** 300-635
- **Exam Name:** Automating and Programming Cisco Data Center Solutions (DCAUTO)
- **New Updated Questions from Braindump2go (Updated in April/2020)**

QUESTION 22
Refer to the exhibit. Assuming a new ACI instance, what is the result when this script is run?



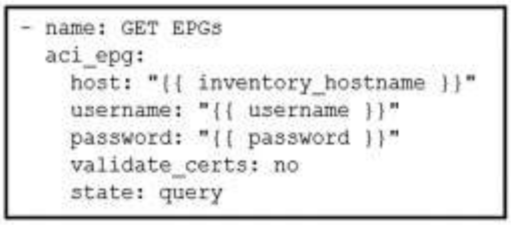
```
def add_tenant():
    token = apic_login.aaaLogin()
    for tenant in range (1,10):
      try:
          response = requests.post(
            url=constant.APIC_URL +"/api/node/mo/uni/tn-exam%s.json" % (tenant),
            headers={
              "Cookie": "APIC-cookie=" + token,
              "Content-Type": "application/json; charset=utf-8",
            },
            data=json.dumps({
              "fvTenant": {
                "attributes": {
                    "status": "created",
                    "dn": "uni/tn-exam%s" % (tenant),
                    "name": "exam%s" % (tenant),
                    "rn": "tn-exam%s" % (tenant)
                },
                "children": [
                ]
              }
            })
          )
          print('Response HTTP Status Code: {status_code}'.format(
            status_code=response.status_code))
          print('Response HTTP Response Body: {content}'.format(
            content=response.content))
      except requests.exceptions.RequestException:
          print('HTTP Request failed')
add_tenant()
```

A. Ten objects are created and subsequently deleted.
B. Nine objects are created.
C. An exception is thrown.
D. Ten objects are created.

Answer: D

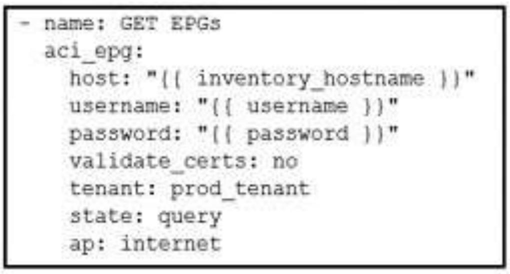
QUESTION 23
Which Ansible playbook fragment returns the fewest queried ACI endpoint groups?

A.



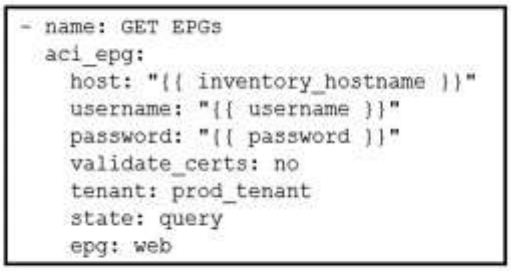
```
- name: GET EPGs
  aci_epg:
    host: "{{ inventory_hostname }}"
    username: "{{ username }}"
    password: "{{ password }}"
    validate_certs: no
    state: query
```

B.



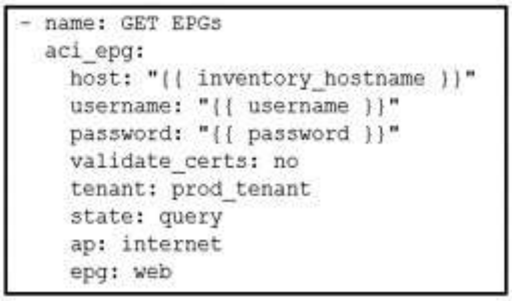
```
- name: GET EPGs
  aci_epg:
    host: "{{ inventory_hostname }}"
    username: "{{ username }}"
    password: "{{ password }}"
    validate_certs: no
    tenant: prod_tenant
    state: query
    ap: internet
```

C.



```
- name: GET EPGs
  aci_epg:
    host: "{{ inventory_hostname }}"
    username: "{{ username }}"
    password: "{{ password }}"
    validate_certs: no
    tenant: prod_tenant
    state: query
    epg: web
```

D.



```
- name: GET EPGs
  aci_epg:
    host: "{{ inventory_hostname }}"
    username: "{{ username }}"
    password: "{{ password }}"
    validate_certs: no
    tenant: prod_tenant
    state: query
    ap: internet
    epg: web
```

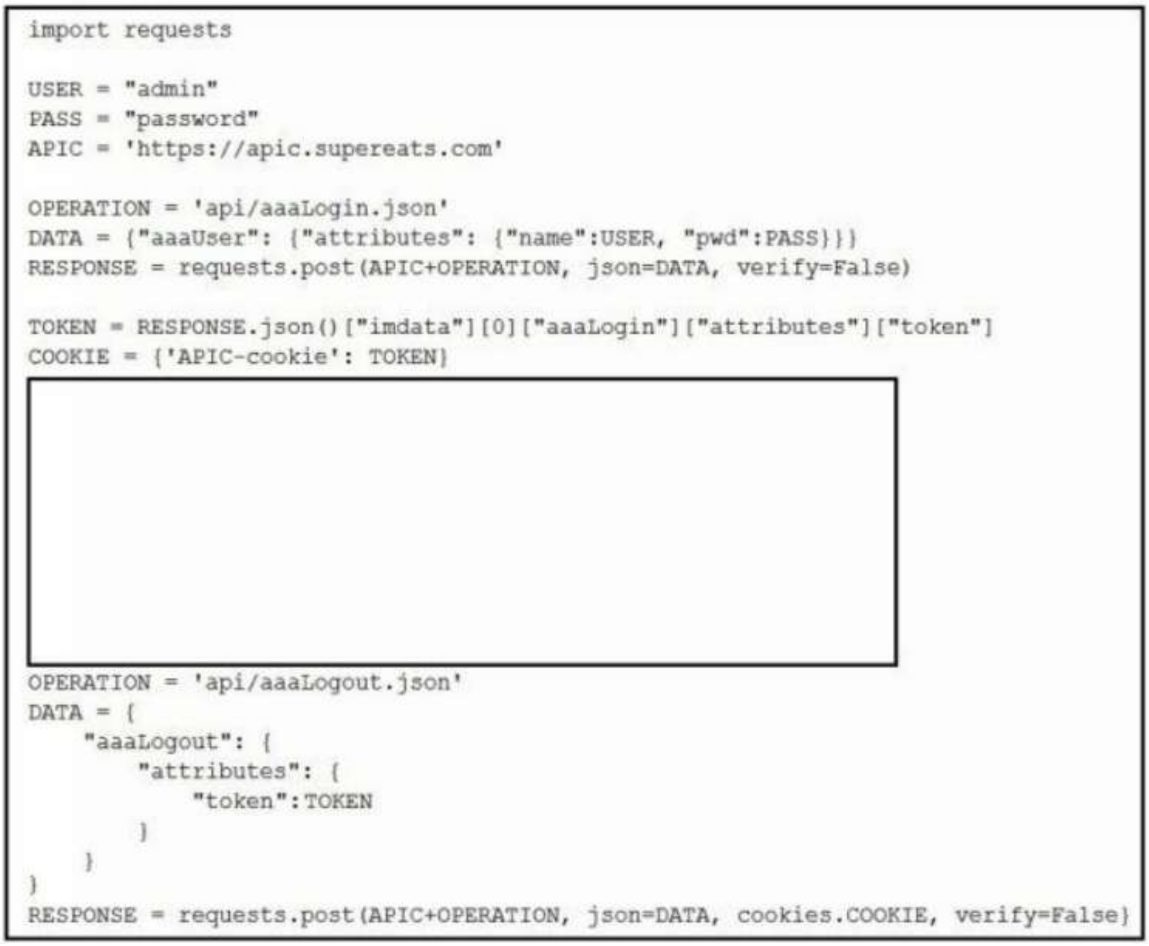
Answer: D

QUESTION 24
How is an ACI class name composed?

A. :Method
B. ClassName:Method
C. Package:ClassName
D. MitName:Method

Answer: C

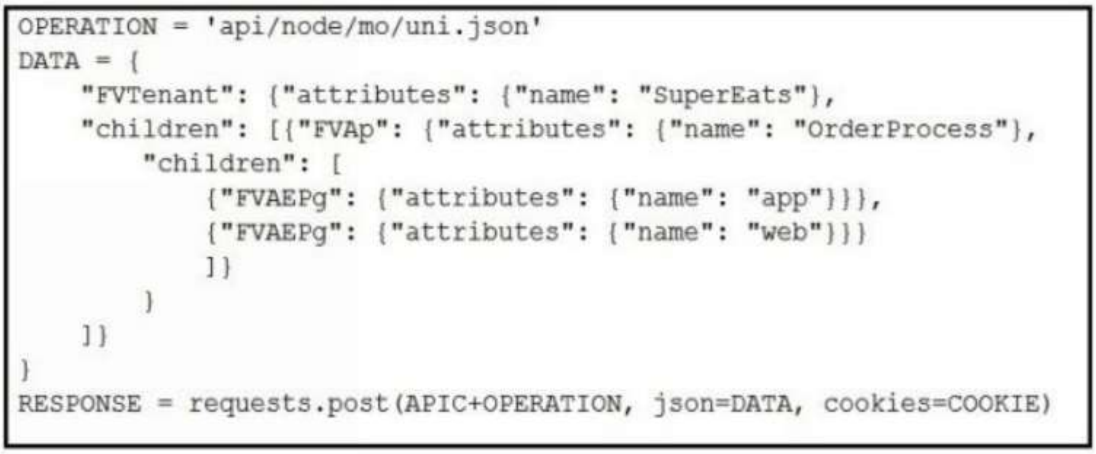
QUESTION 25
Which Python snippets create an application policy named OrderProcess that contains two application endpoint groups under Tenant SuperEats using direct calls to the ACI REST API? Assume that authentication and library imports are correct.



```
import requests

USER = "admin"
PASS = "password"
APIC = 'https://apic.supereats.com'

OPERATION = 'api/aaaLogin.json'
DATA = {"aaaUser": {"attributes": {"name":USER, "pwd":PASS}}}
RESPONSE = requests.post(APIC+OPERATION, json=DATA, verify=False)

TOKEN = RESPONSE.json()["imdata"][0]["aaaLogin"]["attributes"]["token"]
COOKIE = {'APIC-cookie': TOKEN}
```

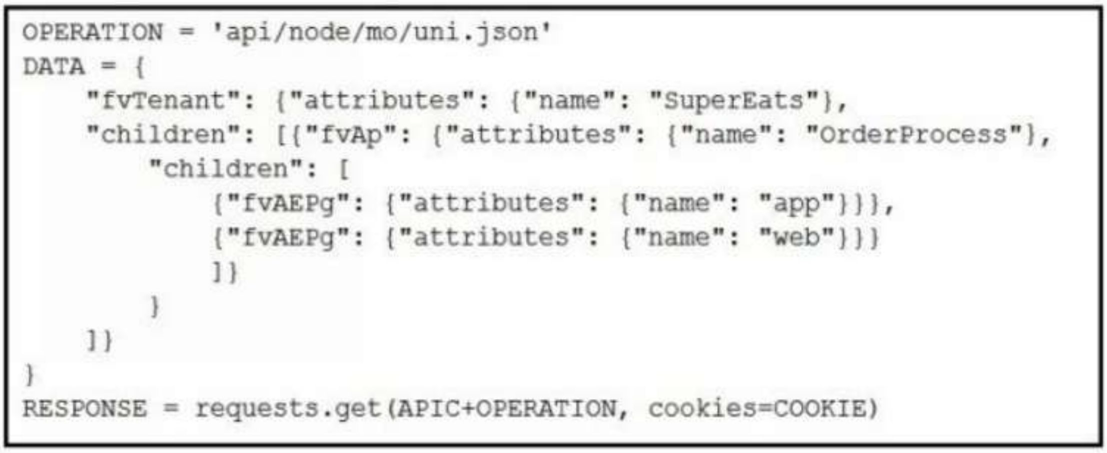
```
OPERATION = 'api/aaaLogout.json'
DATA = {
    "aaaLogout": {
        "attributes": {
            "token":TOKEN
        }
    }
}
RESPONSE = requests.post(APIC+OPERATION, json=DATA, cookies.COOKIE, verify=False}
```

A.



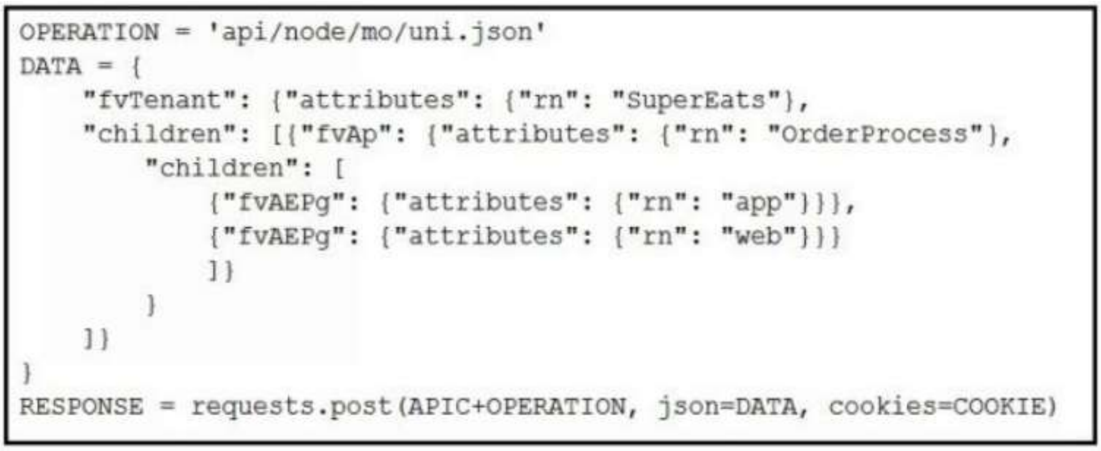
```
OPERATION = 'api/node/mo/uni.json'
DATA = {
    "FVTenant": {"attributes": {"name": "SuperEats"},
    "children": [{"FVAp": {"attributes": {"name": "OrderProcess"},
        "children": [
            {"FVAEPg": {"attributes": {"name": "app"}}},
            {"FVAEPg": {"attributes": {"name": "web"}}}
            ]}
        }
    ]}
}
RESPONSE = requests.post(APIC+OPERATION, json=DATA, cookies=COOKIE)
```

B.



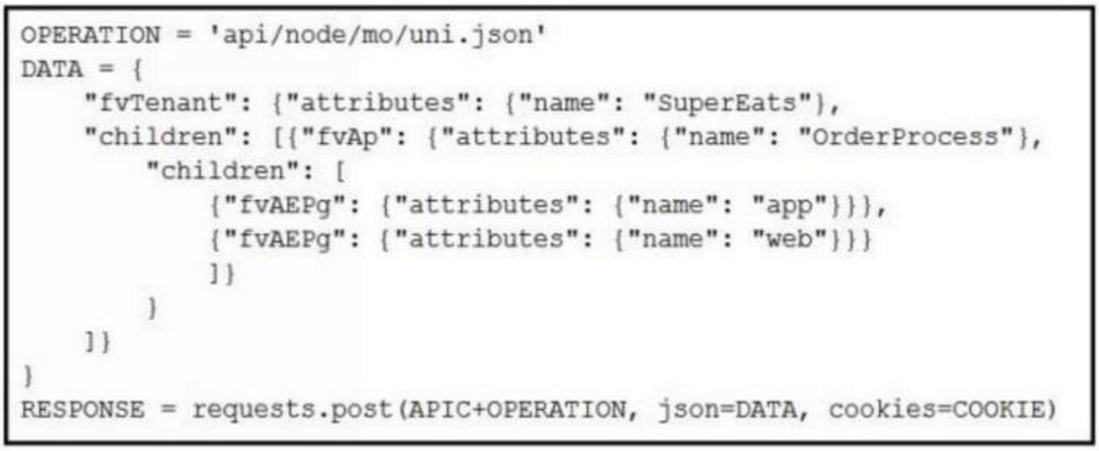
```
OPERATION = 'api/node/mo/uni.json'
DATA = {
    "fvTenant": {"attributes": {"name": "SuperEats"},
    "children": [{"fvAp": {"attributes": {"name": "OrderProcess"},
        "children": [
            {"fvAEPg": {"attributes": {"name": "app"}}},
            {"fvAEPg": {"attributes": {"name": "web"}}}
            ]}
        }
    ]}
}
RESPONSE = requests.get(APIC+OPERATION, cookies=COOKIE)
```

C.



```
OPERATION = 'api/node/mo/uni.json'
DATA = {
    "fvTenant": {"attributes": {"rn": "SuperEats"},
    "children": [{"fvAp": {"attributes": {"rn": "OrderProcess"},
        "children": [
            {"fvAEPg": {"attributes": {"rn": "app"}}},
            {"fvAEPg": {"attributes": {"rn": "web"}}}
            ]}
        }
    ]}
}
RESPONSE = requests.post(APIC+OPERATION, json=DATA, cookies=COOKIE)
```

D.



```
OPERATION = 'api/node/mo/uni.json'
DATA = {
    "fvTenant": {"attributes": {"name": "SuperEats"},
    "children": [{"fvAp": {"attributes": {"name": "OrderProcess"},
        "children": [
            {"fvAEPg": {"attributes": {"name": "app"}}},
            {"fvAEPg": {"attributes": {"name": "web"}}}
            ]}
        }
    ]}
}
RESPONSE = requests.post(APIC+OPERATION, json=DATA, cookies=COOKIE)
```

Answer: D

QUESTION 26
Which management interface is selected by the Cisco APIC by default if an in band management interface and an out of-band management interface exist?

A. In-band is preferred
B. The first configured interface is selected
C. The interface that has the highest priority is selected
D. Out-of band is preferred

Answer: D
Explanation:

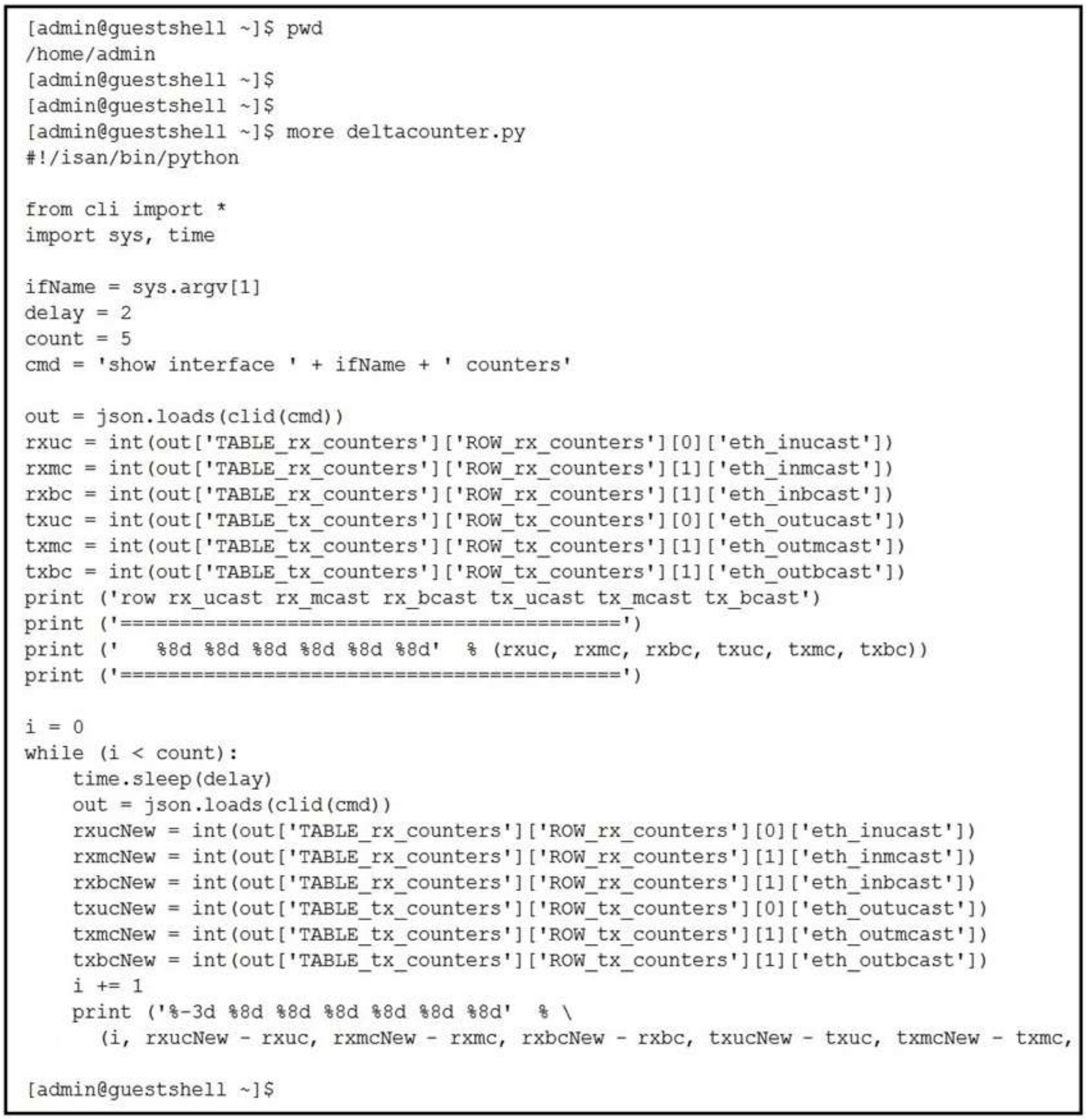
Guidelines and Limitations for In-Band and Out-of-Band Management There is no configuration available to leak the out-of-band management plane from the APIC into the data plane. This can only be accomplished by physically cabling out-of-band network devices directly into the data plane. Cisco does not recommend this setup. The preferred setup for this type of design would be to utilize in-band management.

QUESTION 27
Which tool can you use to convert XML/JSON REST code to Python code?

A. Postman
B. Cobra
C. Arya
D. API Inspector

Answer: C

QUESTION 28
Refer to the exhibit. The script is called deltacounters.py and it is currently inside a Guest Shell container running inside a Cisco NX-OS switch.



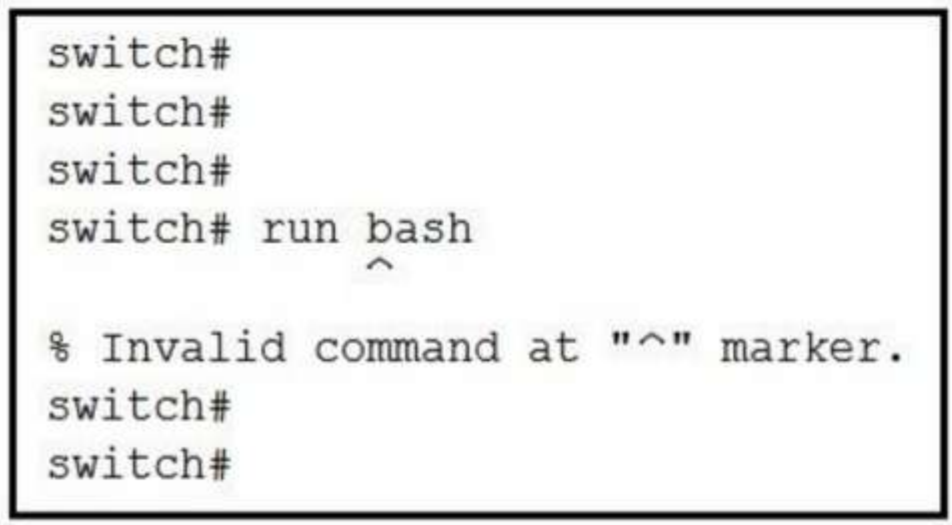
```
[admin@guestshell ~]$ pwd
/home/admin
[admin@guestshell ~]$
[admin@guestshell ~]$
[admin@guestshell ~]$ more deltacounter.py
#!/isan/bin/python

from cli import *
import sys, time

ifName = sys.argv[1]
delay = 2
count = 5
cmd = 'show interface ' + ifName + ' counters'

out = json.loads(clid(cmd))
rxuc = int(out['TABLE_rx_counters']['ROW_rx_counters'][0]['eth_inucast'])
rxmc = int(out['TABLE_rx_counters']['ROW_rx_counters'][1]['eth_inmcast'])
rxbc = int(out['TABLE_rx_counters']['ROW_rx_counters'][1]['eth_inbcast'])
txuc = int(out['TABLE_tx_counters']['ROW_tx_counters'][0]['eth_outucast'])
txmc = int(out['TABLE_tx_counters']['ROW_tx_counters'][1]['eth_outmcast'])
txbc = int(out['TABLE_tx_counters']['ROW_tx_counters'][1]['eth_outbcast'])
print ('row rx_ucast rx_mcast rx_bcast tx_ucast tx_mcast tx_bcast')
print ('=========================================')
print ('   %8d %8d %8d %8d %8d %8d'  % (rxuc, rxmc, rxbc, txuc, txmc, txbc))
print ('=========================================')

i = 0
while (i < count):
    time.sleep(delay)
    out = json.loads(clid(cmd))
    rxucNew = int(out['TABLE_rx_counters']['ROW_rx_counters'][0]['eth_inucast'])
    rxmcNew = int(out['TABLE_rx_counters']['ROW_rx_counters'][1]['eth_inmcast'])
    rxbcNew = int(out['TABLE_rx_counters']['ROW_rx_counters'][1]['eth_inbcast'])
    txucNew = int(out['TABLE_tx_counters']['ROW_tx_counters'][0]['eth_outucast'])
    txmcNew = int(out['TABLE_tx_counters']['ROW_tx_counters'][1]['eth_outmcast'])
    txbcNew = int(out['TABLE_tx_counters']['ROW_tx_counters'][1]['eth_outbcast'])
    i += 1
    print ('%-3d %8d %8d %8d %8d %8d %8d'  % \
      (i, rxucNew - rxuc, rxmcNew - rxmc, rxbcNew - rxbc, txucNew - txuc, txmcNew - txmc,

[admin@guestshell ~]$
```

Which Cisco NX-OS command results in a successful execution of this script?

A. python /home/admin/bootflash:deltacounters.py ethemet1/1
B. show python bootflash:deltacounters.py ethernet1/1
C. guestshell run python /home/admin/deltacounter.py ethernet1/1
D. guestshell execute python /home/admin/deltacounter.py ethernet1/1

Answer: C

QUESTION 29
Refer to the exhibit. Which configuration change command must be run on the Cisco NX-OS device to make this command work?



```
switch#
switch#
switch#
switch# run bash
             ^

% Invalid command at "^" marker.
switch#
switch#
```

A. enable bash-shell
B. bash-shell enable
C. service bash-shell
D. feature bash-shell

Answer: D

QUESTION 30
During the process of starting a Python network telemetry collector, which command starts the Cisco bigmuddy-network-telemetry-collector from GitHub?

A. model driven telemetry
B. telemetry_receiver.py --ip-address <addr> --port <port>
C. telemetry_receiver.py --destination <port> --url <url>
D. streaming telemetry

Answer: B

QUESTION 31
When the Cisco bigmuddy-network-telemetry-collector from GitHub is used, which command displays only the message headers?

A. --print
B. --all
C. --brief
D. --print-all

Answer: C

QUESTION 32
What are two differences between SNMP and model-driven telemetry? (Choose two.)

A. SNMP uses a continuous stream model.
B. SNMP uses a push model.
C. SNMP uses a pull model.
D. Model-driven telemetry uses a pull model.
E. Model-driven telemetry uses a push model.

Answer: CE