300-835 ^{Q&As}

Automating Cisco Collaboration Solutions (CLAUTO)

Pass Cisco 300-835 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.leads4pass.com/300-835.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Cisco Official Exam Center

⚙ **Instant Download** After Purchase

⚙ **100% Money Back** Guarantee

⚙ **365 Days** Free Update

⚙ **800,000+** Satisfied Customers

QUESTION 1

DRAG DROP

Drag and drop the XML snippets onto the AXL request to build a request that sets the cluster-wide Call Park Reversion Timer to 2 minutes. Not all options are used.

Select and Place:

Answer Area

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ns="http://www.cisco.com/AXL/API/11.5">
    <soapenv:Header/>
    <soapenv:Body>

        [                                                    ]

            [                                                ]

            [                                                ]

                <service>Cisco CallManager</service>
                <value>120</value>

        [                                                    ]

    </soapenv:Body>
</soapenv:Envelope>
```

```
<name>CallParkDisplayTimer</name>
```

```
<name>CallParkReversionTimeout</name>
```

```
<ns:updateProcessNodeService>
```

```
<processNodeName>Default</processNodeName>
```

```
</ns:updateServiceParameter>
```

```
<ns:updateServiceParameter>
```

```
</ns:updateProcessNodeService>
```

```
<processNodeName>EnterpriseWideData</processNodeName>
```

Correct Answer:

Answer Area

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ns="http://www.cisco.com/AXL/API/11.5">
        <soapenv:Header/>
        <soapenv:Body>

            <ns:updateServiceParameter>

            <processNodeName>EnterpriseWideData</processNodeName>

                <name>CallParkReversionTimeout</name>

            <service>Cisco CallManager</service>
            <value>120</value>

                </ns:updateServiceParameter>

        </soapenv:Body>
</soapenv:Envelope>
```

```
                <name>CallParkDisplayTimer</name>
```

```
                <ns:updateProcessNodeService>
```

```
            <processNodeName>Default</processNodeName>
```

```
                </ns:updateProcessNodeService>
```

QUESTION 2

Which two files must be downloaded from a Cisco Unified Communications Manager server in order to build requests for the AXL API? (Choose two.)

A. AXLEnums.xsd

B. AXL_CUCM.xsd

C. AXLSoap.xsd

D. CUCM_AXL.json

E. AXLWSDL.xml

Correct Answer: AC

QUESTION 3

DRAG DROP

Drag and drop the elements from the left onto the correct descriptions on the right. Not all options are used.
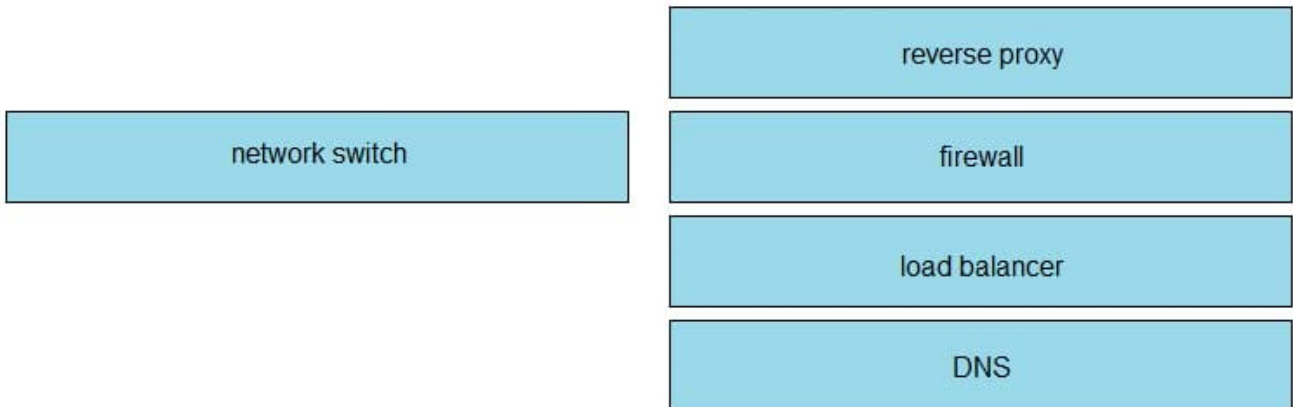
Select and Place:

Answer Area

reverse proxy	sits in front of web servers and retrieves resources from the web servers on behalf of clients; used to increase security
network switch	controls incoming traffic based on defined rules
load balancer	device that distributes application traffic across a number of servers; used to increase capacity
firewall	translates domain names to IP addresses so browsers can load Internet resources
DNS	

Correct Answer:

Answer Area

| network switch |

| reverse proxy |

| firewall |

| load balancer |

| DNS |

QUESTION 4

DRAG DROP

A small Python script is constructed that creates a Webex Meeting for John Doe scheduled for the current time. Drag and drop the code snippets at the bottom onto the areas of the source code exhibit to create a program. Not all options are used.

Select and Place:

Answer Area

```
import requests, datetime
timestamp = datetime.datetime.now()
startDate = timestamp.strftime('%m/%d/%Y %H:%M:%S')
xml = f'''<message><header><securityContext>
```

```
</securityContext></header>
<body><bodyContent xsi:type="java:com.webex.service.binding.meeting.CreateMeeting">
```

```
<siteName>apidemoeu</siteName>
<webExID>johndoe@example.com</webExID>
<token>AAABbSKM1UYAOgAKEkdfUOhBMjU2XOFMR09SSVRITV8=</token>
```

```
<schedule><startDate>(startDate)</startDate></schedule>
</bodyContent></body></message>'''
print(requests.post('https://api.webex.com/WBXService/XMLService', xml))
```

```
<metaData>
<confName>Sample Meeting</confName><meetingType>105</meetingType>
</metaData>
```

```
<siteName>apidemoeu</siteName>
<webExID>johndoe@example.com</webExID>
<sessionTicket>AAABbSKM1UYAOgAKEkdfUOhBMjU2XOFMR09SSVRITV8=</sessionTicket>
```

```
<metaData>
<confName>Sample Meeting</confName><meetingType>Training Center</meetingType>
</metaData>
```

Correct Answer:

Answer Area

```
import requests, datetime
timestamp = datetime.datetime.now()
startDate = timestamp.strftime('%m/%d/%Y %H:%M:%S')
xml = f'''<message><header><securityContext>

    <siteName>apidemoeu</siteName>
    <webExID>johndoe@example.com</webExID>
    <sessionTicket>AAABbSKM1UYAOgAKEkdfUOhBMjU2XOFMR09SSVRITV8=</sessionTicket>

</securityContext></header>
<body><bodyContent xsi:type="java:com.webex.service.binding.meeting.CreateMeeting">

    <metaData>
    <confName>Sample Meeting</confName><meetingType>105</meetingType>
    </metaData>

    <schedule><startDate>(startDate)</startDate></schedule>
</bodyContent></body></message>'''
print(requests.post('https://api.webex.com/WBXService/XMLService', xml))
```

```
    <siteName>apidemoeu</siteName>
    <webExID>johndoe@example.com</webExID>
    <token>AAABbSKM1UYAOgAKEkdfUOhBMjU2XOFMR09SSVRITV8=</token>
```

```
    <metaData>
    <confName>Sample Meeting</confName><meetingType>Training Center</meetingType>
    </metaData>
```

QUESTION 5

DRAG DROP

The Python script executes a REST API to retrieve information from the Cisco Meeting Server about the call with an identifier of "987654321". Drag and drop the code onto the snippet to complete the Python script. Not all options are used.

Select and Place:

Answer Area

```
import requests

url = 'https://myserver:445/api/v1/ [          ] /987654321'

headers = {
    ' [          ] ': 'Basic YXBpOmRDbG91ZDEyMyE='
}

response = requests.request(' [          ] ', url, headers = headers)

print(response.text)
```

call	GET
Authorization	Authentication
calls	PUT
POST	callid

Correct Answer:

Answer Area

```
import requests

url = 'https://myserver:445/api/v1/ [ calls ] /987654321'

headers = {
    ' [ Authorization ] ': 'Basic YXBpOmRDbG91ZDEyMyE='
}

response = requests.request(' [ POST ] ', url, headers = headers)

print(response.text)
```

call

GET

Authentication

PUT

callid