QUESTION 28
A user is receiving a 429 Too Many Requests error. Which scheme is the server employing that causes this error?

A. rate limiting
B. time outs
C. caching
D. redirection

Correct Answer: A

QUESTION 29
Refer to the exhibit. As part of the Ansible playbook workflow, several new interfaces are being configured using the netconf_config module. The task references the interface variables that are unique per device. In which directory is the YAML file with these variables found?

```
- name: Configure Interfaces
  with_items: "{{interfaces}}"
  netconf_config:
    <<: *host_info
    xml: |
      <config>
        <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interefaces">
          <interface>
            <name>{{item.interface_type}}{{item.interface_id}}</name>
            <description>{{item.description}}</description>
            <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">ianaift:ethernetCsmacd</type>
            <enabled>true</enabled>
            <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
              <address>
                <ip>{{item.ip_address}}</ip>
                <netmask>{{item.subnet_mask}}</netmask>
              </address>
            </ipv4>
          </interface>
        </interfaces>
      </config>
```

A. host_vars directory
B. home directory
C. group_vars directory
D. current working directory

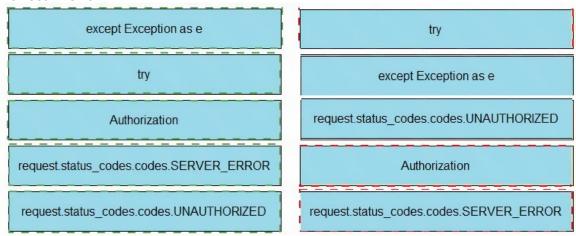**Correct Answer:** A

QUESTION 30
DRAG DROP
Refer to the exhibit. Drag and drop the code snippets from the left onto the item numbers on the right that match the missing sections in the exhibit to complete the script to implement control flow.

```
import request
import json
import sys

token = ""

def get_dnac_devices():
    <item 1>:
        url = "https://sanboxdnac.cisco.com/dna/intent/api/v1/network-device"

        print(token)
        payload = {}
        headers = {
        'Content-Type': 'application/json',
        "Accept": 'application/json',
        'x-auth-token': token
        }

        response = requests.request("GET", url, headers=headers, data = payload)
        response.raise_for_status()
        return response.text

    <item 2>:
        print(e)
        if str(<item 3>) in str(e):
            create_dnac_token()

def create_dnac_token():
    try:
        url = "https://sanboxdnac.cisco.com/dna/system/api/v1/auth/token"

        payload = {}
        headers ={
        '<item4>': 'Basic ZGV2bmV0dXNlcjpDaXNjbzEyMyE=',
        'Content-Type': 'application/json'
        }

        response = requests.request("POST", url, headers=headers, data = payload)
        response.raise_for_status()
        return response.json()["Token"]
    except Exception as e:
        print(e)
        if str(<item 5>) in str(e):
            sys.exit("DNAC Service is not reachable")

if ____name____ == "____main____":
    token = create_dnac_token()
    print(get_dnac_devices())
```

except Exception as e	<item 1>
try	<item 2>
Authorization	<item 3>
request.status_codes.codes.SERVER_ERROR	<item 4>
request.status_codes.codes.UNAUTHORIZED	<item 5>

Correct Answer:

except Exception as e	try
try	except Exception as e
Authorization	request.status_codes.codes.UNAUTHORIZED
request.status_codes.codes.SERVER_ERROR	Authorization
request.status_codes.codes.UNAUTHORIZED	request.status_codes.codes.SERVER_ERROR

QUESTION 31
Which type of file is created from issued intermediate, root, and primary certificates for SSL installation on a server?

A. DER
B. CSR
C. PEM
D. CRT

Correct Answer: C

QUESTION 32
DRAG DROP
Refer to the exhibit. The Python script is supposed to make an API call to Cisco DNA Center querying a wireless profile for the "ChicagoCampus" and then parsing out its enable FlexConnect value. Drag and drop the parts of the Python code from the left onto the item numbers on the right that match the missing sections in the exhibit.

GET	/dna/intent/api/v1/wireless/profile Get Wireless Profile

Gets either one or all the wireless network profiles if no name is provided for network-profile.

Parameters

Name	Description
profileName **string** *(query)*	*Default value:*

Responses

Code	Description
200	The request was successful. The result is contained in the response body.

Example Value Model

```
[
  {
    "profileDetails": {
      "name": "string",
      "sites": [
        "string"
      ],
      "ssidDetails": [
        {
          "name": "string",
          "type": "Guest",
          "enabledFabric": true,
          "flexConnect": {
            "enableFlexConnect": true,
            "localToVlan": 0
          },
          "InterfaceName": "string"
        }
      ]
    }
  }
]
```

```
import requests
import json

def get_dnac_wireless_profiles():
    try:
        url = "https://sandboxdnac2.cisco.com/dna/intent/api/v1" \
        + "/wireless/profile?<item1>=ChicagoCampus|"

        print(token)
        payload = {}
        headers = {
        'x-auth-token': token
        }

        response = requests.request("GET", url, headers=headers, data = payload)
        response.raise_for_status()
        return response.json()[0]['<item 2>']['<item 3>'] \
                          [<item 4>]['<item 5>']["<item 6>"]
    except Exception as e:
        print(e)

def create_dnac_token():
    try:
        url = "https://sandboxdnac2.cisco.com/dna/system/api/v1/auth/token"

        payload = {}
        headers = {
        'Authorization': 'Basic ZGV2bmV0dXNlcjpDaXNjbzEyMyE= ',
        'Content-Type': 'application/json'
        }

        response = requests.request("POST", url, headers=headers, data = payload)
        response.raise_for_status()
        return response.json()["Token"]
    except Exception as e:
        print(e)

if ____name____ == "__main__":
    token = create_dnac_token()
    print(get_dnac_wireless_profiles())
```

0	<item 1>
ssidDetails	<item 2>
profileDetails	<item 3>
profileName	<item 4>
flexConnect	<item 5>
enableFlexConnect	<item 6>