Practice Exam Questions

350-901

Developing Applications using Cisco Core Platforms & APIs

CISCO™

Total Question: 137 QAs

Question No: 1
A network operations team is using the cloud to automate some of their managed customer and branch locations. They require that all of their tooling be ephemeral by design and that the entire automation environment can be recreated without manual commands. Automation code and configuration state will be stored in git for change control and versioning. The engineering high-level plan is to use VMs in a cloud-provider environment then configure open source tooling onto these VMs to poll, test, and configure the remote devices, as well as deploy the tooling itself.
Which configuration management and/or automation tooling is needed for this solution?
A. Ansible
B. Ansible and Terraform
C. NSO
D. Terraform
E. Ansible and NSO
Answer: B

Question No: 2
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
Answer: A

Question No: 3
Refer to the exhibit.

```
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())
```

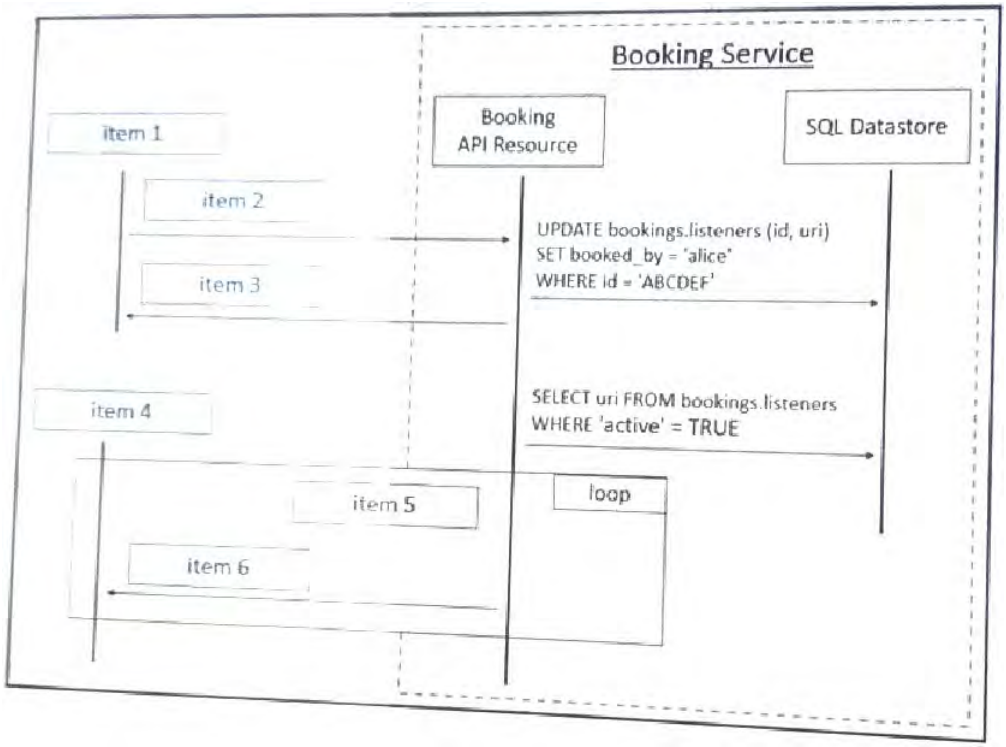
Drag and drop the code snippets from the left onto the item numbers on the right that match the missing sections in the exhibit to complete the script to implement control flow.

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>

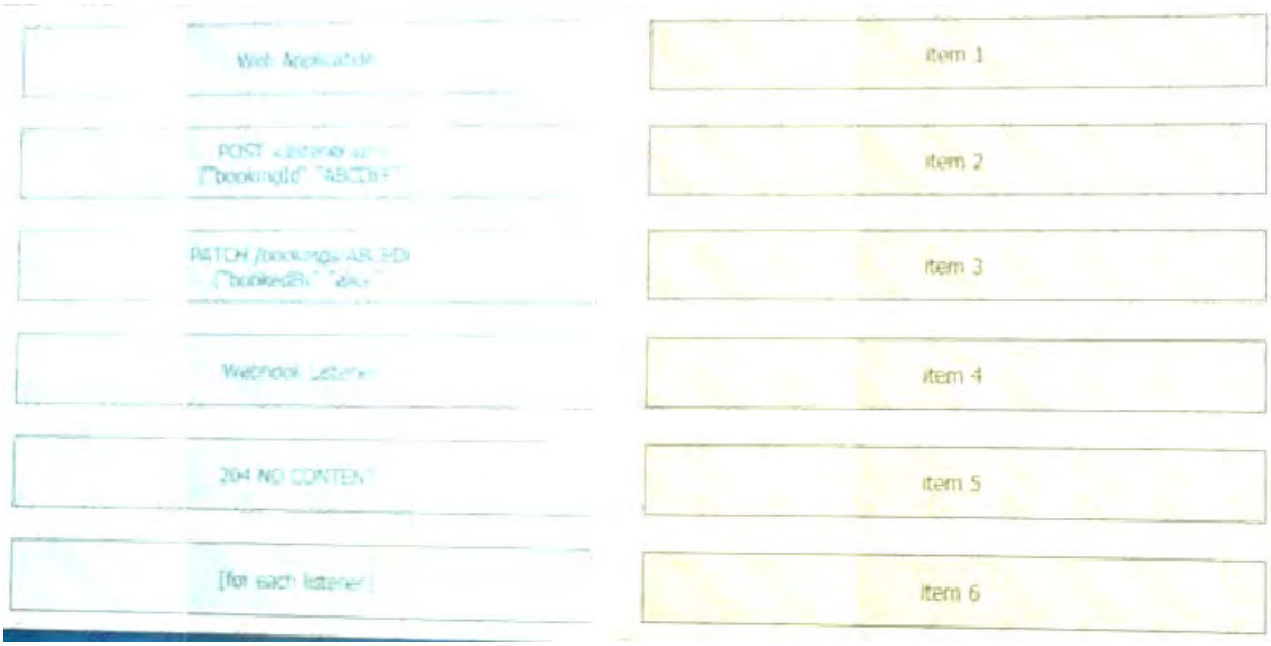
Answer:

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 No: 4



Refer to the exhibit above and click on the tab in the top left corner to view a diagram that describes the typical flow of requests involved when a webhook is created for a booking service. Drag and drop the requests from the left onto the item numbers on the right that match the missing sections in the sequence diagram to design the complete flow of requests involved as a booking is updated from a web application.

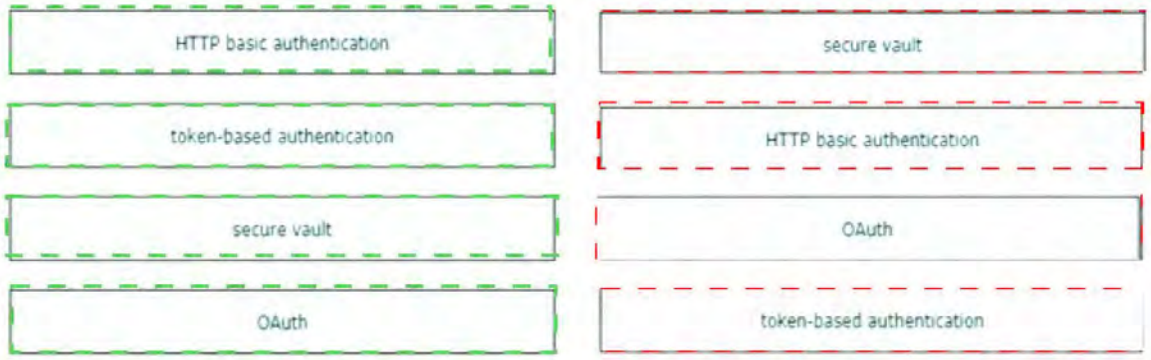


Answer:

Question No: 5

Drag and drop the REST API authentication method from the left to the description on the right

Answer:



Question No: 6

Into which two areas are AppDynamics APIs categorized? (Choose two.)

A. application-centric

B. analytics-events

C. database-visibility

D. platform-side

E. agent-side

Answer: D,E

Question No: 7

Refer to the exhibit.

```
open_file = open("text_file.txt", "r")
read_file = open_file.read()
print(read_file)
```

A developer created the code, but it fails to execute. Which code snippet helps to identify the issue?

A.
```
try:
  open_file = open("text_file.txt", "r")
  read_file = open_file.read()
  print(read_file)
except:
  print("File not there")
```

B.
```
try:
  print("File not there")
except:
  open_file = open("text_file.txt", "r")
  read_file = open_file.read()
  print(read_file)
```

C.
```
try:
  open_file = open("text_file.txt", "r")
  read_file = open_file.read()
  print(read_file)
except:
  print("File not there")
catch:
  error(read_file)
```

D.
```
  open_file = open("text_file.txt", "r")
  read_file = open_file.read()
try:
  print(read_file)
except:
  print("File not there")
```

A. Option A
B. Option B
C. Option C
D. Option D
Answer: A

Question No: 8
A local Docker image has an image ID of 386231131. Fill in the blanks to complete the command in order to tag the image into the "cisco" repository with "version1 0'".

$ docker tag [ ] [ ]

Answer:
386231131, cisco/386231131:version1.0

Question No: 9
A cloud native project is being worked on in which all source code and dependencies are written in Python, Ruby, and/or JavaScipt. A change in code triggers a notification to the CI/CD tool to run the CI/CD pipeline. Which step should be omitted from the pipeline?
A. Deploy the code to one or more environments, such as staging and/or production.
B. Build one of more containers that package up code and all its dependencies.
C. Compile code.
D. Run automated tests to validate the correctness.
Answer: C

Question No: 10
Which two design considerations should be considered when building a Cisco Meraki dashboard out of

available APIs? (Choose two,)

A. API call volume is rate-limited to five calls per second per organization.

B. The API version does not need to be specified in the URL.

C. Access to the API must first be enabled by using the settings for an organization.

D. The API requests require the key and the user credentials.

E. If the API key is shared, it cannot be regenerated

Answer: B,C

Question No: 11

Refer to the exhibit.

Drag and drop the steps from the left into the correct order of operation on the right for a successful OAuth2 three-legged authorization code grant flow.

Client initiates the flow	step 1
The authorization server authenticates the client, validates details sent, and responds with an access token.	step 2
The authorization server redirects the user-agent back to the client using the recirection URI provided.	step 3
The authorization server authenticates the resource owner.	step 4
The client requests an access token from the authorization server's token endpoint.	step 5

Answer:

Client initiates the flow	Client initiates the flow
The authorization server authenticates the client, validates details sent, and responds with an access token.	The authorization server authenticates the resource owner.
The authorization server redirects the user-agent back to the client using the redirection URI provided.	The authorization server redirects the user-agent back to the client using the redirection URI provided.
The authorization server authenticates the resource owner.	The client requests an access token from the authorization server's token endpoint.
The client requests an access token from the authorization server's token endpoint.	The authorization server authenticates the client, validates details sent, and responds with an access token.

Question No: 12

Where should distributed load balancing occur in a horizontally scalable architecture?

A. firewall-side/policy load balancing

B. network-side/central load balancing