Detailed 350-901 Answers Exam Pass Certify | 350-901: Developing Applications using Cisco Core Platforms and APIs (DEVCOR)

DOWNLOAD the newest Itcertkey 350-901 PDF dumps from Cloud Storage for free: https://drive.google.com/open?id=13BWt19YnIWJIPQTLhStsDFSLIy7t_Q95

If you want to get a good job, and if you are not satisfied with your present situation, if you long to have a higher station in life. We think it is high time for you to try your best to gain the 350-901 certification. Having our study materials, it will be very easy for you to get the certification in a short time. If you try purchase our study materials, you will find our 350-901 question torrent will be very useful for you. We are confident that you will be attracted to our 350-901 guide question.

Cisco 350-901: Developing Applications using Cisco Core Platforms and APIs (DEVCOR) exam is one of the most-awaited certifications to validate your skills and knowledge to develop applications using Cisco platforms and APIs. Developing Applications using Cisco Core Platforms and APIs (DEVCOR) certification exam is intended for software developers, network professionals, and solution architects who want to enhance their knowledge and expertise in developing applications using Cisco core platforms and APIs.

>> Detailed 350-901 Answers <<

350-901 Practice Test Engine | 350-901 Valid Test Notes

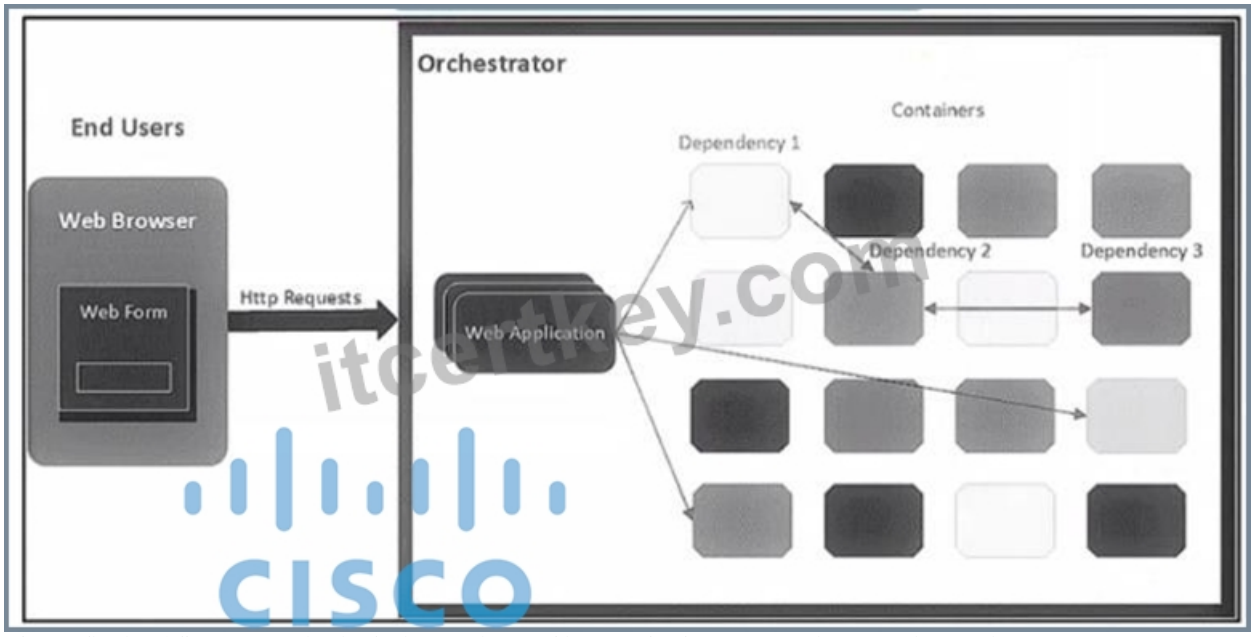
The clients can download our products and use our 350-901 study materials immediately after they pay successfully with their credit cards. Our system will send our 350-901 learning prep in the form of mails to the client in 5-10 minutes after their successful payment. The mails provide the links and if only the clients click on the links they can log in our software immediately to learn our 350-901 Guide materials. If there are something they can't understand, they can contact with our service and we will solve them right away.

Cisco 350-901 Exam is an essential certification for IT professionals who want to excel in developing applications using Cisco core platforms and APIs. Passing 350-901 exam demonstrates the candidate's ability to design, develop, and test applications using Cisco technologies, which is a highly sought-after skill in the IT industry. Developing Applications using Cisco Core Platforms and APIs (DEVCOR) certification can help the candidate to stand out in a competitive job market and increase their earning potential.

Cisco Developing Applications using Cisco Core Platforms and APIs (DEVCOR) Sample Questions (Q556-Q561):

NEW QUESTION # 556

Refer to the exhibit.



The application follows a containerized microservices architecture that has one container per microservice.
The microservices communicate with each other by using REST APIs The double-headed arrows m the diagram display chains of synchronous HTTP calls needed for a single use/request Which action ensures the resilience of the application in the scope of a single user request'

- A. Redesign the application to be separated into these three layers Presentation. API. and Data
- B. Set up multiple instances of each microservice m active/active mode by using the Orchestrator
- C. Implement retries with exponential backoff during HTTP API calls
- D. Create two virtual machines that each host an instance of the application and set up a cluster

Answer: A

NEW QUESTION # 557
User report that they can no longer process transactions with the online ordering application, and the logging dashboard is displaying these messages.
Fri Jan 10 19:37:31.123 EST 2020 [FRONTEND] INFO: Incoming request to add item to cart from user 45834534858
Fri Jan 10 19:37:31 247 EST 2020 [BACKEND] INFO: Attempting to add item to cart Fri Jan 10 19:37:31 250 EST 2020 [BACKEND] ERROR: Failed to add item: MYSQLDB ERROR:
Connection refused
What is causing the problem seen in these log messages?

- A. The user is not authorized to add the item to their cart.
- B. The backend is not authorized to commit to the database.
- C. The database server container has crashed.
- D. The backend process is overwhelmed with too many transactions.

Answer: C

NEW QUESTION # 558
Refer to the exhibit.

```
import sys, requests

URL = "http://ios-xe-mgmt.cisco.com:9443"
USER - 'root'
PASS = 'C!isco0123'

url = URL + "/restconf/data/ietf-interfaces:interfaces-state"
headers = {'content-type': 'application/vang-data+json', 'accept':
          'application/yang-data+json'          }

try:
   result = requests.get(url, auth=(USER,PASS), headers=headers)
   r_json = result.json()
   flagDown = 0
   for record in r_json["ietf-interfaces:interfaces"]["interface"]:
        print("{0:<35}".format("interface: " + record["name"]), end="")
        print("{0:<5}".format("ip: "), end="")
        if('address' in record["ietf-ip:ipv4"]):
             print("{0:<15}".format(record["ietf-ip:ipv4"]["address"][0]["ip"]), end="")
        else:
             print("{0:<15}".format(record["No IPv4"], end="")
        print("{0:<9}".format("status: "), end="")
        print(str(record["enabled"]))
        if(record["enabled"]==False):
             flagDown=1
   print("")
   if(flagDown):
        print("At least one interface is down")
   else
        print("All interfaces are up")

except:
   print("Exception: " + str(sys.ecx_info()[0] + " " + str(sys.ecx_info()[1]))
   print("Error: " + str(result.status_code), result.text)
```

What is the output of this IOS-XE configuration program?

- A. interface administrative status in IPv6 addresses
- B. interface administrative status in IPv4 addresses
- C. interface operational status in IPv4 addresses
- D. interface operational status in IPv6 addresses

Answer: B

NEW QUESTION # 559
A developer has created an application based on customer requirements. The customer needs to run the application with the minimum downtime. Which design approach regarding high- availability applications, Recovery Time Objective, and Recovery Point Objective must be taken?

- A. Active/passive results in lower RTO and RPO.
  For RPO, data synchronization between the two data centers must be timely to allow seamless request flow.
- B. Active/active results in lower RTO and RPO.
  For RPO, data synchronization between the two data centers does not need to be timely to allow seamless request flow.
- C. Active/passive results in lower RTO and RPO.
  For RPO, data synchronization between the two data centers does not need to be timely to allow seamless request flow.
- D. Active/active results in lower RTO and RPO.
  For RPO, data synchronization between the two data centers must be timely to allow seamless request flow.

Answer: D

Explanation:
The question is mentioning the lower RTO using 'minimum downtime' so we can translate that way:
HA = Active/Active
RPO = synchronous data
RTO = minimum downtime
https://www.ibm.com/garage/method/practices/manage/hadr-on-premises-app/

NEW QUESTION # 560

1.5 Getting Started

1.5.1 Connecting Disconnecting

```
from ucsmsdk.ucshandle import UcsHandle

# Create a connection handle
handle = UcsHandle("192.168.1.1", "admin", "password")

# Login to the server
handle.login()

# Logout from the server
handle.logout()
```

Refer UcsHandle API Reference for detailed parameter sets to `UcsHandle`

This module contains the general information for ComputePooledSlot ManagedObject.

class **ucsmsdk.mometa.compute.ComputePooledSlot.ComputePooledSlot**(*parent_mo_or_dn, chassis_id, slot_id, **kwargs*) [source]

Bases: **ucsmsdk.ucsmo.ManagedObject**

This is ComputePooledSlot class.

consts = *<ucsmsdk.mometa.compute.ComputePooledSlot.ComputePooledSlotConsts instance>*

mo_meta = *<ucsmsdk.ucscoremeta.MoMeta object>*

naming_props = *set([u'chassisId', u'slotId'])*

prop_map = *{'dn': 'dn', 'status': 'status', 'sacl': 'sacl', 'slotId': 'slot_id', 'assigned': 'assigned', 'owner': 'owner', 'prevAssignedToDn': 'prev_assigned_to_dn', 'childAction': 'child_action', 'poolableDn': 'poolable_dn', 'chassisId': 'chassis_id', 'rn': 'rn', 'assignedToDn': 'assigned_to_dn'}*

prop_meta = *{'dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad250>, 'status': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad5d0>, 'sacl': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad4d0>, 'assigned_to_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x123392b10>, 'assigned': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x123392bd0>, 'owner': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad2d0>, 'child_action': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad1d0>, 'poolable_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad350>, 'chassis_id': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x123392ad0>, 'slot_id': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad550>, 'prev_assigned_to_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad3d0>, 'rn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1233ad450>}*

class ucsmsdk.mometa.compute.ComputePool.**ComputePool**(*parent_mo_or_dn, name, **kwargs*) [source]

Bases: **ucsmsdk.ucsmo.ManagedObject**

This is ComputePool class.

consts = *<ucsmsdk.mometa.compute.ComputePool.ComputePoolConsts instance>*
mo_meta = *<ucsmsdk.ucscoremeta.MoMeta object>*
naming_props = *set([u'name'])*
prop_map = *{'dn': 'dn', 'status': 'status', 'policyLevel': 'policy_level', 'assignmentOrder': 'assignment_order', 'sacl': 'sacl', 'policyOwner': 'policy_owner', 'assigned': 'assigned', 'intId': 'int_id', 'childAction': 'child_action', 'name': 'name', 'descr': 'descr', 'rn': 'rn', 'size': 'size'}*
prop_meta = *{'dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230f8f90>, 'status': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed3d0>, 'sacl': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed2d0>, 'assigned': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230f8d90>, 'int_id': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed050>, 'assignment_order': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230f8e10>, 'child_action': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230f8e90>, 'name': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed0d0>, 'descr': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230f8f10>, 'policy_owner': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed1d0>, 'policy_level': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed150>, 'rn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed250>, 'size': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x1230ed350>}*

1.5.2 Base APIs

The SDK provides APIs to enable CRUD operations.

- **C**reate an object - `add_mo`
- **R**etrieve an object - `query_dn,query_classid,query_dns,query_classids`
- **U**pdate an object - `set_mo`
- **D**elete an object - `delete_mo`

The above APIs can be bunched together in a transaction (All or None). `commit_mo` commits the changes made using the above APIs.

All these methods are invoked on a `UcsHandle` instance. We refer it by `handle` in all the examples here-after. Refer to the *Connecting Disconnecting* to create a new handle.

1.5.3 Creating Objects

Creating managed objects is done via `add_mo` API.
Example:
The below example creates a new Service Profile(`LsServer`) Object under the parent `org-root`

```
from ucsmsdk.mometa.ls.LsServer import LsServer

sp = LsServer(parent_mo_or_dn="org-root", name="sp_demo")
handle.add_mo(sp)
```

note: the changes will only be sent to server when `handle.commit()` is called.
Add Mo API reference

class ucsmsdk.mometa.ls.LsRequirement.**LsRequirement**(*parent_mo_or_dn*, ***kwargs*) [source]

Bases: **ucsmsdk.ucsmo.ManagedObject**

This is LsRequirement class.

consts = *<ucsmsdk.mometa.ls.LsRequirement.LsRequirementConsts instance>*

mo_meta = *<ucsmsdk.ucscoremeta.MoMeta object>*

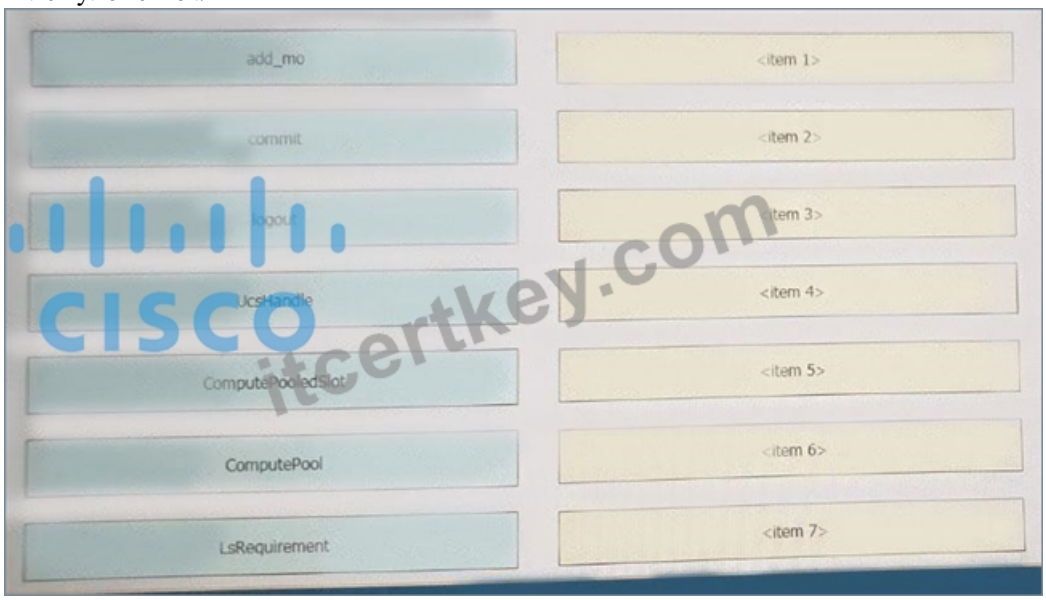
naming_props = *set([])*

prop_map = *{'dn': 'dn', 'status': 'status', 'operState': 'oper_state', 'qualifier': 'qualifier', 'sacl': 'sacl', 'pnDn': 'pn_dn', 'restrictMigration': 'restrict_migration', 'issues': 'issues', 'operName': 'oper_name', 'pnPoolDn': 'pn_pool_dn', 'name': 'name', 'computeEpDn': 'compute_ep_dn', 'rn': 'rn', 'childAction': 'child_action', 'assignedToDn': 'assigned_to_dn'}*

prop_meta = *{'dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x122cfbf10>, 'status': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e892790>, 'qualifier': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e892350>, 'sacl': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e892690>, 'pn_pool_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e8929d0>, 'assigned_to_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x122cfbd90>, 'oper_state': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e892a90>, 'issues': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e892450>, 'child_action': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x122cfb990>, 'name': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e8921d0>, 'oper_name': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e892a10>, 'rn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e892090>, 'restrict_migration': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e892110>, 'pn_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x12e8926d0>, 'compute_ep_dn': <ucsmsdk.ucscoremeta.MoPropertyMeta object at 0x122cfb350>}*

```
""" Create UCS Server Pool and associate to template """

from ucsmsdk.ucshandle import UcsHandle
from ucsmsdk.mometa.compute.ComputePool import ComputePool
from ucsmsdk.mometa.compute.ComputePooledSlot import ComputePooledSlot
from ucsmsdk.mometa.ls.LsRequirement import LsRequirement

HANDLE = <item 1>(
    "sandbox-ucsml.cisco.com",
    "admin",
    "password"
)
HANDLE.login()

SERVER_POOL = <item 2>(
    parent_mo_or_dn="org-root/org-devnet",
    name="devcore_pool"
)
HANDLE.<item 3>(SERVER_POOL, modify_present=True)

for blade in HANDLE.query_classid(
    "computeBlade",
    filter_str='(chassis_id, "7")'
    ):
    SERVER = <item 4>(
        parent_mo_or_dn=SERVER_POOL,
        chassis_id=blade.chassis_id,
        slot_id=blade.slot_id
    )
    HANDLE.add_mo(SERVER, modify_present=True)
HANDLE.commit()

SP_TEMPLATE = <item 5>(
    parent_mo_or_dn="org-root/org-devnet/ls-devcore_template",
    name="devcore_pool"
)
HANDLE.add_mo(SP_TEMPLATE, modify_present=True)
HANDLE.<item 6>()

HANDLE.<item 7>()
```

Refer to the exhibit above and click on the resource tabs in the top left corner to view resources to help with this question. Python code using the UCS Python SDK is creating a server pool named "devcore_pool" and populating the pool with all servers from chassis 7 and then the server pool is associated to existing service profile template "devcore_template". Drag and drop the code snippets from the left onto the item numbers on the right that match the missing sections in the python exhibit.
Refer to the above and click on the resource labs in the top left corner to view resources to help with this question.
Python code using the UCS Python SDK is creating a server pool named "devcore_pool" and populating the pool with all servers from chassis 7, and then the server pool is associated to existing Service Profile template "devcore_template" Drag and drop the code snippets from the left onto the item numbers on the right that match the missing sections in the Python exhibit.



Answer:

Explanation:

NEW QUESTION # 561

.....

350-901 Practice Test Engine: https://www.itcertkey.com/350-901_braindumps.html

- 350-901 Pass4sure Torrent - 350-901 Valid Pdf - 350-901 Testking Exam □ Open website 《 www.troytecdumps.com 》 and search for " 350-901 " for free download □Flexible 350-901 Testing Engine
- Real 350-901 Dumps □ Test 350-901 Testking ↖ Test 350-901 Testking □ Go to website ⇛ www.pdfvce.com ⇚ open and search for ➡ 350-901 □□□ to download for free □Valid 350-901 Real Test
- 350-901 Downloadable PDF □ Flexible 350-901 Testing Engine □ 350-901 New Study Materials □ Go to website " www.practicevce.com " open and search for （ 350-901 ） to download for free □350-901 Downloadable PDF
- Latest updated Detailed 350-901 Answers Spend Your Little Time and Energy to Clear 350-901 exam □ The page for free download of ➢ 350-901 □ on ➡ www.pdfvce.com □□□ will open immediately □350-901 Pass4sure Study Materials
- Exam Discount 350-901 Voucher □ Valid 350-901 Real Test □ 350-901 Latest Guide Files □ The page for free download of { 350-901 } on " www.examcollectionpass.com " will open immediately □Valid 350-901 Real Test
- Valid 350-901 Real Test □ Reliable 350-901 Dumps Ebook □ Valid 350-901 Real Test □ Search for □ 350-901 □ on ➽ www.pdfvce.com □ immediately to obtain a free download □350-901 Pass4sure Study Materials
- Cisco - 350-901 - Unparalleled Detailed Developing Applications using Cisco Core Platforms and APIs (DEVCOR) Answers ↗ Easily obtain free download of [350-901] by searching on ✔ www.examcollectionpass.com □✔□ ⌀350-901 New Study Materials
- Developing Applications using Cisco Core Platforms and APIs (DEVCOR) exam training solutions - 350-901 latest practice questions - Developing Applications using Cisco Core Platforms and APIs (DEVCOR) free download material □ Immediately open ➥ www.pdfvce.com □ and search for 《 350-901 》 to obtain a free download □350-901 Free Updates
- Detailed 350-901 Answers | 100% Free Excellent Developing Applications using Cisco Core Platforms and APIs (DEVCOR) Practice Test Engine □ Search for □ 350-901 □ and download it for free immediately on ✔ www.exam4labs.com □✔□ □Valid 350-901 Real Test
- 350-901 Pass4sure Torrent - 350-901 Valid Pdf - 350-901 Testking Exam □ Go to website □ www.pdfvce.com □ open and search for ➡ 350-901 □□□ to download for free □Valid 350-901 Real Test

- Cisco - 350-901 - Unparalleled Detailed Developing Applications using Cisco Core Platforms and APIs (DEVCOR) Answers □ Open 《 www.troytecdumps.com 》 enter " 350-901 " and obtain a free download □Valid Test 350-901 Tips
- myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, bbs.t-firefly.com, beinstatistics.com, ycs.instructure.com, ncon.edu.sa, www.stes.tyc.edu.tw, pastebin.com, www.stes.tyc.edu.tw, bbs.t-firefly.com, www.stes.tyc.edu.tw, Disposable vapes

What's more, part of that Itcertkey 350-901 dumps now are free: https://drive.google.com/open?id=13BWt19YnIWJIPQTLhStsDFSLIy7t_Q95