Designing and Implementing Cisco Network Programmability (300-550)

Exam Description: The 300-550 exam tests a candidate's knowledge and skills related to Network programmability fundamentals, APIs and automation protocols, data models, operations, and controllers.

The following topics are general guidelines for the content likely to be included on the exam. However, other related topics may also appear on any specific delivery of the exam. In order to better reflect the contents of the exam and for clarity purposes, the guidelines below may change at any time without notice.

16% 1.0 Network Programmability Fundamentals
   1.1 Analyze and modify Python code to meet specified requirements
   1.2 Describe the use cases for an SDK
   1.3 Choose the appropriate Linux command to complete tasks:
      1.3.a Use RESTful APIs
      1.3.b Navigate the file system
      1.3.c Manage processes
      1.3.d Install and manage packages
      1.3.e Manage network configurations
   1.4 Describe the importance of securing management plane interfaces
   1.5 Describe the principle of least privilege and how it applies to programmatic interfaces
   1.6 Describe the methods of securing management plane interfaces
   1.7 Compare and contrast white-list vs. black-list security approaches

24% 2.0 APIs and Automation Protocols
   2.1 Identify the use cases for north-bound and south-bound APIs
   2.2 Describe the characteristics, benefits, capabilities, and functionalities of the APIs associated with the following operating systems:
      2.2.a NX-OS
      2.2.b IOS XR
      2.2.c IOS XE
      2.2.d ASA OS
   2.3 Compare and contrast the RESTful, RESTCONF, and NETCONF APIs
2.4 Describe the basic characteristics of OpenFlow

2.5 Describe the characteristics of the following protocols for APIs
2.5.a TLS
2.5.b HTTPS
2.5.c HTTP
2.5.d GRPC
2.5.e SSH

23% 3.0 Data models
3.1 Validate YANG based models for semantics and syntax
3.2 Describe the uses of YANG models in NETCONF and RESTCONF
3.3 Develop YANG models for specific use cases
3.4 Describe the use cases for platform-specific, vendor-specific, and standardized data models
3.5 Describe the Management Information Tree (MIT) on Nexus, ACI, and UCS
3.6 Explain how to use VISORE to navigate the MIT

16% 4.0 Operations
4.1 Describe the use cases for orchestration and automation
4.2 Compare and contrast agent-based with agent-less configuration management
Mechanisms
4.3 Describe the methods for enabling the following configuration management mechanisms:
4.3.a Puppet
4.3.b Chef
4.3.c Ansible
4.4 Describe how OpenStack Neutron interacts with a physical network infrastructure
4.5 Compare and contrast the software development methodologies Agile and Waterfall
4.6 Explain the DevOps concepts of:
4.6.a version control
4.6.b continuous integration
4.6.c configuration management
4.6.d automated validation testing
4.6.e continuous deployment
4.7 Compare and contrast the benefits of Virtual Network Functions (VNF)/Network Function Virtualization (NFV) with traditional network or service architectures
4.8 Describe the following platforms used to test network infrastructure automation
4.8.a VIRL
4.8.b APIC Simulator
4.8.c NX-OSv (via VIRL)
4.8.d DevNet Sandboxes
4.8.e ASA
4.8.f CSR1000v

4.9 Describe how VIRL can be used as an automated test tool

4.10 Describe how to setup a test environment using VIRL

4.11 Evaluate the impact of an automated task on a network

21% 5.0 Controllers
5.1 Deploy and configure the following:
   5.1.a APIC-EM
   5.1.b ACI
   5.1.c OSC

5.2 Troubleshoot the following:
   5.2.a APIC-EM
   5.2.b APIC
   5.2.c OSC

5.3 Describe the capabilities and architecture of:
   5.3.a APIC-EM
   5.3.b ACI
   5.3.c OSC

5.4 Construct an API request against:
   5.4.a APIC-EM
   5.4.b APIC
   5.4.c OSC

5.5 Describe how APIC cluster gets leveraged during an API call

5.6 Explain the use cases of Digital Networking Architecture in reference to the following technologies:
   5.6.a Network Service Orchestration (NSO)
   5.6.b Virtual Topology System (VTS)
   5.6.c Nexus Data Broker
   5.6.d WAN Automation Engine (WAE)
   5.6.e UCS Director
   5.6.f UCS Manager