



Automating Cisco Data Center Solutions v1.1 (300-635)

Exam Description: Automating Cisco Data Center Solutions v1.1 (DCAUTO 300-635) is a 90-minute exam associated with the CCNP Data Center Certification and DevNet Professional Certification. This exam certifies a candidate's knowledge of implementing Data Center automated solutions, including programming concepts, orchestration and automation tools.

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. To better reflect the contents of the exam and for clarity purposes, the guidelines below may change at any time without notice.

- 10%** **1.0** **Network Programmability Foundation**
 - 1.1 Utilize common version control operations with git: add, clone, push, commit, diff, branching, merging conflict
 - 1.2 Describe characteristics of API styles (REST and RPC)
 - 1.3 Describe the challenges encountered and patterns used when consuming APIs synchronously and asynchronously
 - 1.4 Interpret Python scripts containing data types, functions, classes, conditions, and looping
 - 1.5 Describe the benefits of Python virtual environments
 - 1.6 Explain the benefits of using network configuration tools such as Ansible and Terraform for automating data center platforms

- 30%** **2.0** **Controller Based Data Center Networking**
 - 2.1 Describe the following:
 - 2.1.a ACI target policy
 - 2.1.b ACI application hosting capabilities
 - 2.1.c Implementation of an ACI application from the Cisco ACI Apps Center
 - 2.2 Leverage the API inspector to explore the REST API calls made by the ACI GUI
 - 2.3 Construct a Python script to create an application policy using the ACI REST API
 - 2.4 Construct a Python script to create an application policy using the ACI Cobra SDK
 - 2.5 Construct an Ansible playbook to create an application policy
 - 2.6 Describe the benefits of integrating Kubernetes infrastructure using the ACI CNI plugin
 - 2.7** Construct a Terraform plan to use an ACI or Nexus Dashboard Fabric Controller

- 30%** **3.0** **Data Center Device-centric Networking**
 - 3.1 Describe Day 0 provisioning with NX-OS
 - 3.1.a Cisco POAP
 - 3.1.b NX-OS iPXE
 - 3.2 Implement On-Box Programmability and Automation with NX-OS
 - 3.2.a Bash

- 3.2.b Linux containers (LXC and Docker using provided container)
- 3.2.c NX-OS guest shell
- 3.2.d Embedded Event Manager (EEM)
- 3.2.e On-box Python Scripting

- 3.3 Compare model-driven telemetry such as YANG Push and gRPC to traditional network monitoring strategies such as SNMP, Netflow, and SYSLOG
- 3.4 Construct Python script that consumes model-driven telemetry data with NX-OS
- 3.5 Implement Off-Box Programmability and Automation with NX-OS
 - 3.5.a Nexus NX-API (NX-API REST and NX-API CLI)
 - 3.5.b Nexus NETCONF using native and OpenConfig
 - 3.5.c Network configuration tools with NX-OS (Ansible)

- 30%** **4.0 Data Center Compute**
 - 4.1 Configure Cisco UCS with developer tools
 - 4.1.a UCS PowerTool
 - 4.1.b UCS Python SDK
 - 4.1.c Ansible

 - 4.2 Describe the capabilities of the Nexus Dashboard Fabric Controller API
 - 4.3 Identify the steps in the Cisco Intersight API authentication method
 - 4.4 Construct a Cisco Intersight API call given documentation to accomplish tasks such as manage server policies, service profiles, and firmware updates
 - 4.5 Interpret a Terraform plan to use the Cisco Intersight provider