Automating Networks Using Cisco Platforms v1.1 (200-901)

Exam Description: Automating Networks Using Cisco Platforms v1.1 (CCNAAUTO 200-901) is a 120-minute exam associated with the CCNA Automation Certification. This exam certifies a candidate's knowledge of software development and design including understanding and using APIs, application deployment and security, and infrastructure and automation on Cisco platforms. The course, Developing Applications and Automating Workflows using Cisco Core Platforms, helps candidates to prepare for this exam.

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.

15%	**1.0**	**Software Development and Design**
	1.1	Compare data formats (XML, JSON, and YAML)
	1.2	Describe parsing of common data format (XML, JSON, and YAML) to Python data structures
	1.3	Describe the concepts of test-driven development
	1.4	Compare software development methods (agile, lean, and waterfall)
	1.5	Explain the benefits of organizing code into methods / functions, classes, and modules
	1.6	Explain the advantages of common design patterns (MVC and Observer)
	1.7	Explain the advantages of version control
	1.8	Utilize common version control operations with Git
		1.8.a Clone
		1.8.b Add/remove
		1.8.c Commit
		1.8.d Push / pull
		1.8.e Branch
		1.8.f Merge and handling conflicts
		1.8.g diff
20%	**2.0**	**Understanding and Using APIs**
	2.1	Construct a REST API request to accomplish a task given API documentation
	2.2	Describe common usage patterns related to webhooks
	2.3	Describe the constraints when consuming APIs
	2.4	Explain common HTTP response codes associated with REST APIs
	2.5	Troubleshoot a problem given the HTTP response code, request and API documentation
	2.6	Interpret the parts of an HTTP response (response code, headers, body)
	2.7	Utilize common API authentication mechanisms: basic, custom token, and API keys
	2.8	Compare common API styles (REST, RPC, synchronous, and asynchronous)
	2.9	Construct a Python script that calls a REST API using the requests library

2025 Cisco Systems, Inc. This document is Cisco Public.

15%	**3.0**	**Cisco Platforms and Development**
	3.1	Construct a Python script that uses a Cisco SDK given SDK documentation
	3.2	Describe the capabilities of Cisco network management platforms and APIs (Meraki, Cisco Catalyst Center, ACI, Cisco Catalyst SD-WAN, and NSO)
	3.3	Describe the capabilities of Cisco compute management platforms and APIs (UCS Manager and Intersight)
	3.4	Describe the capabilities of Cisco collaboration platforms and APIs (Webex, Webex devices, Cisco Unified Communications Manager including AXL and UDS interfaces)
	3.5	Describe the capabilities of Cisco security platforms and APIs (XDR, Firepower, Secure Connect, Secure Endpoint, ISE, and Secure Malware Analytics)
	3.6	Describe the device level APIs and dynamic interfaces for IOS XE and NX-OS
	3.7	Describe the appropriate DevNet resource for a given scenario (Sandbox, Code Exchange, support, forums, Learning Labs, and API documentation)
	3.8	Apply concepts of model driven programmability (YANG, RESTCONF, and NETCONF) in a Cisco environment
	3.9	Construct code to perform a specific operation based on a set of requirements and given API reference documentation such as these:
		3.9.a Obtain a list of network devices by using Meraki, Cisco Catalyst Center, ACI, Cisco Catalyst SD-WAN, or NSO
		3.9.b Manage spaces, participants, and messages in Webex
		3.9.c Obtain a list of clients / hosts seen on a network using Meraki or Cisco Catalyst Center
15%	**4.0**	**Application Deployment and Security**
	4.1	Describe the benefits of edge computing
	4.2	Describe the attributes of different application deployment models (private cloud, public cloud, hybrid cloud, and edge)
	4.3	Describe the attributes of these application deployment types
		4.3.a Virtual machines
		4.3.b Bare metal
		4.3.c Containers
	4.4	Describe components for a CI/CD pipeline in application deployments
	4.5	Construct a Python unit test
	4.6	Interpret contents of a Dockerfile
	4.7	Utilize Docker images in local developer environment
	4.8	Describe application security issues related to secret protection, encryption (storage and transport), and data handling
	4.9	Explain the role of firewalls, DNS, load balancers, and reverse proxy in application deployment
	4.10	Describe top OWASP threats (such as XSS, SQL injections, and CSRF)
	4.11	Utilize Bash commands (file management, directory navigation, and environmental variables)
	4.12	Describe the principles of DevOps practices
20%	**5.0**	**Infrastructure and Automation**
	5.1	Describe the value of model driven programmability for infrastructure automation
	5.2	Compare controller-level to device-level management

2025 Cisco Systems, Inc. This document is Cisco Public.

	5.3	Describe the use and roles of network simulation and test tools (such as Cisco Modeling Labs and pyATS)
	5.4	Describe the components and benefits of CI/CD pipeline in infrastructure automation
	5.5	Describe the principles of infrastructure as code
	5.6	Describe the capabilities of automation tools such as Ansible, Terraform, and Cisco NSO
	5.7	Identify the workflow being automated by a Python script that uses Cisco APIs including ACI, Meraki, Cisco Catalyst Center, and RESTCONF
	5.8	Interpret the workflow being automated by an Ansible playbook (management packages, user management related to services, basic service configuration, and start/stop)
	5.9	Interpret the workflow being automated by a bash script (such as file management, app install, user management, directory navigation)
	5.10	Interpret the results of a RESTCONF or NETCONF query
	5.11	Interpret basic YANG models
	5.12	Interpret a unified diff
	5.13	Describe the principles and benefits of a code review process
	5.14	Interpret a sequence diagram that includes API calls
15%	**6.0**	**Network Fundamentals**
	6.1	Describe the purpose and usage of MAC addresses and VLANs
	6.2	Describe the purpose and usage of IP addresses, routes, subnet mask / prefix, and gateways
	6.3	Describe the function of common networking components (such as switches, routers, firewalls, and load balancers)
	6.4	Interpret a basic network topology diagram with elements such as switches, routers, firewalls, load balancers, and port values
	6.5	Describe the function of management, data, and control planes in a network device
	6.6	Describe the functionality of these IP Services: DHCP, DNS, NAT, SNMP, NTP
	6.7	Recognize common protocol port values (such as, SSH, Telnet, HTTP, HTTPS, and NETCONF)
	6.8	Diagnose application connectivity issues (NAT problem, Transport Port blocked, proxy, and VPN)
	6.9	Explain the impacts of network constraints on applications