Designing, Deploying and Managing Network Automation Systems v2.0 (350-901)

Exam Description: Designing, Deploying and Managing Network Automation Systems v2.0 (AUTOCOR 350-901) is a 120-minute exam associated with the CCNP and CCIE Automation Certification. This exam certifies a candidate's knowledge of network automation systems development and design including infrastructure as code, operations, and AI in automation. The course, Designing, Deploying and Managing Network Automation Systems, helps candidates 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.

30% 1.0 Network Automation

- 1.1 Construct a network automation solution with Ansible to manage configurations such as VLANs, OSPF, asset management, interface settings, and ACLs
- 1.2 Construct a network automation solution with Terraform to manage configurations such as VLANs, OSPF, asset management, interface settings, and ACLs
- 1.3 Construct a network automation solution with RESTCONF (RFC 8040), given the YANG model, to manage configurations such as VLANs, OSPF, asset management, interface settings, and ACLs
- 1.4 Construct a network automation solution with Python to manage configurations such as VLANs, OSPF, asset management, interface settings, and ACLs
- 1.5 Select the network automation approach to achieve technical and business requirements considering options such as infrastructure as code framework, low code/no code, and custom applications
- 1.6 Construct a network automation solution that consumes REST APIs including extended API attributes (such as pagination, complex authentication workflows, and rate limiting), error handling, and persistent authentication

30% 2.0 Infrastructure as Code

- 2.1 Use version control operations with Git
  - 2.1.a Merge a branch including squash and conflict resolution
  - 2.1.b git cherry-pick
  - 2.1.c git reset
  - 2.1.d git checkout
  - 2.1.e git revert
- 2.2 Diagnose a GitLab CE CI/CD pipeline failure such as missing dependency, incompatible versions of components, and failed tests
- 2.3 Construct a GitLab CE CI/CD pipeline to deploy a network automation solution including stages for:
  - 2.3.a build
  - 2.3.b prevalidation

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

2.3.c deploy
2.3.d post-validation

2.4 Construct a network simulation with Cisco Modeling Labs (CML) to test the network automation solution

2.5 Interpret a Docker Compose file including services, networks, volumes, and links

2.6 Integrate source of truth into a network automation solution

2.7 Construct a YAML or JSON representation of a network configuration given a YANG-based data model

20% 3.0 Operations

3.1 Describe architectural components of model-driven telemetry

3.2 Implement a logging strategy for a network automation solution targeting destinations such as syslog or webhooks

3.3 Diagnose problems with network automation given logs and output related to an event

3.4 Implement change validation for a network automation solution using pyATS CLI tools

3.5 Describe the process to obtain and deploy CA-signed TLS certificates

3.6 Implement secure coding practices into a network automation solution to meet input validation, authentication, and secret management requirements

20% 4.0 AI in Automation

4.1 Describe the benefits and risks of AI-assisted code development for network automation such as data privacy, IP ownership, and code validation

4.2 Interpret the security risks in a given AI-based network automation solution

4.3 Construct an MCP server to provide network information to an AI-agent using Python FastMCP

4.4 Construct a conversational agent that leverages LLMs for network automation

4.5 Evaluate the accuracy of AI recommendations on a network automation solution

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