CCNA 200-301 v1.1

New Topic Supplement

Free update for readers of

Cisco CCNA in 60 Days and CCNA Simplified

About this Supplement

Cisco updated the CCNA 200-301 exam to version 1.1 in 2024. This free supplement covers the new and updated exam topics not included in the current edition of *Cisco CCNA in 60 Days and CCNA Simplified*. Read this alongside your book — no other study material is required for the v1.1 additions.

New topics covered:

Topic	Exam Objective	Chapter
AI, Generative AI & Machine Learning in Network Operations	6.4	1
Cloud-Managed Network Device Access	2.8	2
Terraform for Configuration Management	6.6	3

Author: *Paul Browning* | ***Website:*** *www.howtonetwork.com* | ***Version:*** *2024*

Ch 1

AI, Generative AI & Machine Learning in Network Operations

Exam Objective: 6.4

Exam Objective: 6.4 — Explain AI (generative and predictive) and machine learning in network operations

Introduction

Artificial Intelligence and Machine Learning have moved into everyday network operations. Cisco added this topic to CCNA v1.1 because modern networks increasingly rely on AI-driven tools for monitoring, threat detection, and automation. For the exam you need to understand the concepts and how they apply to networking — not write code.

What is Artificial Intelligence (AI)?

AI refers to systems designed to simulate human reasoning and decision-making. Two categories are relevant to networking:

Narrow AI	Trained for specific tasks — facial recognition, spam filtering, anomaly detection in network traffic. Used in most current network management tools.
General AI	A theoretical concept: an AI system capable of any cognitive task a human can perform. Not yet achieved; not directly tested on the exam.

Machine Learning (ML)

ML is a subset of AI that enables systems to learn from data without being explicitly programmed for every scenario. Three main types:

Supervised Learning	Trains on labelled data to make predictions. Example: classifying traffic as normal or malicious based on past examples.
Unsupervised Learning	Finds patterns in unlabelled data. Example: clustering unusual traffic flows to detect anomalies.
Reinforcement Learning	Learns optimal actions through trial and error using rewards and penalties. Example: adaptive routing adjusting to network conditions.

A specialised form called **Deep Learning** uses multi-layered neural networks to process large volumes of unstructured data — enabling advanced traffic analysis and speech recognition.

■ Key Distinction for the Exam

ML is a subset of AI. Deep Learning is a subset of ML. Supervised = labelled data. Unsupervised = unlabelled data. Reinforcement = reward/penalty feedback.

ML Applications in Network Operations

Supervised Learning

• **Classification:** Identifies whether traffic is normal or malicious — supporting threat detection.

• **Regression:** Predicts future traffic volumes or latency for capacity planning.

Unsupervised Learning

• **Clustering:** Groups similar traffic patterns — identifies unusual behaviour or segments users.

• **Anomaly Detection:** Learns normal behaviour and flags deviations — security breaches, hardware faults, traffic spikes.

Reinforcement Learning

• **Adaptive Network Management:** Dynamically adjusts routing policies based on real-time feedback.

• **Load Balancing:** Automatically redistributes traffic to optimise performance and avoid congestion.

Predictive AI in Network Operations

Predictive AI analyses historical and real-time data to forecast future events — allowing teams to act before problems occur rather than reacting afterwards.

Predictive Maintenance	Analyses device performance trends to identify hardware about to fail — enabling proactive replacement and minimising downtime.
Traffic Forecasting	Predicts bandwidth demand over time — prevents congestion at peak periods.
Security Threat Prediction	Identifies traffic patterns that historically precede attacks — allows pre-emptive action.
Time-Series Forecasting	Evaluates data over time to predict future network load — used for capacity planning.

Generative AI in Network Operations

Generative AI creates new data or content that mirrors existing patterns. In networking it simulates scenarios, automates configuration, and generates test data.

GANs (Generative Adversarial Networks)	Two competing neural networks — a generator creates data; a discriminator evaluates it. Used to simulate realistic network traffic for testing and training.
VAEs (Variational Autoencoders)	Encode data into compressed form then reconstruct it — generating diverse realistic examples. Used for data compression and synthetic example generation.

Generative AI Applications in Networks:

• **Network Design:** Simulates multiple configurations to find optimal designs without physical testing.

• **Automated Configuration:** Generates device config scripts automatically — reduces errors, ensures consistency.

• **Scenario Simulation:** Creates realistic traffic patterns for testing and can simulate attacks to test defences.

Real-World Examples

Cisco Cognitive Threat Analytics	Uses ML to identify suspicious network flows and provide early warning of security threats.
Ericsson AI Tool	Managed 44 million daily call minutes and provided connectivity for 1.4 million visitors at a 2022 Qatar event.
Juniper AI (Philadelphia)	AI-driven network management reduced security risks and freed IT staff for higher-value tasks.

■ **Exam Tip:** Know the difference between generative AI (creates new data/content) and predictive AI (forecasts future events). Know supervised = labelled data, unsupervised = unlabelled, reinforcement = reward/penalty. Anomaly detection and traffic forecasting are the most commonly tested ML applications.

Further Reading

• howtonetwork.com/certifications/ai/ai-and-machine-learning-explained/

• howtonetwork.com/certifications/ai/what-is-artificial-intelligence-ai/

• blogs.cisco.com/analytics-automation/cisco-ai-network-analytics

Ch 2

Cloud-Managed Network Device Access

Exam Objective: 2.8

Exam Objective: 2.8 — Describe network device management access (Telnet, SSH, HTTP, HTTPS, console, TACACS+/RADIUS, and cloud managed)

Introduction

The v1.1 update adds "cloud managed" to exam objective 2.8. You need to understand how cloud management platforms work alongside traditional access methods, and their key advantages and limitations.

Management Access Methods Compared

Method	Type	Key Characteristics
Console	Traditional	Physical direct connection via console cable. Used for initial setup or recovery.
Telnet	Traditional	Remote CLI — plaintext. Not recommended (no encryption).
SSH	Traditional	Encrypted remote CLI. Preferred method for secure management.
HTTP/HTTPS	Traditional	Web-based GUI. HTTPS encrypted; HTTP is not.
TACACS+/RADIUS	Traditional	AAA servers for centralised authentication and access control.
Cloud Managed	Cloud	Managed via cloud dashboard (e.g. Cisco Meraki). No direct CLI needed for day-to-day management.

How Cloud-Managed Networking Works

In a cloud-managed architecture, network devices (switches, routers, APs, firewalls) connect to a cloud-hosted management platform. Administrators use a web dashboard or API rather than logging into each device individually.

Cisco Meraki is the primary exam example:

• Devices managed entirely through the Meraki cloud dashboard

• Configuration changes pushed from the cloud to devices automatically

• Monitoring, alerts, and reporting centralised in one place

• No direct SSH or console access required for day-to-day management

• Devices "phone home" to the cloud to receive configuration updates

Advantages of Cloud-Managed Access

Centralised Management	Single dashboard manages all devices across multiple sites — no per-device login needed.
Zero-Touch Provisioning	New devices ship directly to branch sites, connect to the internet, and automatically pull their config from the cloud.
Auto Updates	Firmware and security patches pushed automatically — reduces maintenance overhead.
Visibility & Analytics	Real-time dashboards showing performance, client stats, and security alerts across the whole network.
Scalability	Adding new sites requires minimal on-site expertise — config managed centrally.
Remote Troubleshooting	Diagnose and resolve issues from anywhere with internet access.

Considerations

• **Internet Dependency:** Management plane requires cloud connectivity. If internet fails, management may be unavailable — but the data plane (forwarding traffic) continues.

• **Subscription Cost:** Cloud platforms typically require ongoing per-device licence fees.

• **Data Sovereignty:** Config and telemetry stored in the cloud — consider data privacy requirements.

> ■ **Exam Tip:** Know all methods listed in 2.8: Telnet, SSH, HTTP, HTTPS, console, TACACS+/RADIUS, and cloud managed. For cloud managed: know it uses a central dashboard, supports zero-touch provisioning, and that the management plane depends on internet connectivity while the data plane continues if the cloud connection is lost.

Further Reading

• howtonetwork.com/certifications/cloud/cloud-managed-network-devices/

• documentation.meraki.com

• developer.cisco.com/meraki/

Ch 3

Terraform for Configuration Management

Exam Objective: 6.6

Exam Objective: 6.6 — Recognize the capabilities of configuration management mechanisms such as Ansible and Terraform

Introduction

CCNA v1.1 shifted focus from Puppet and Chef to **Ansible and Terraform**. Ansible was in v1.0; Terraform is the new addition. Both automate infrastructure configuration but work differently and serve different purposes.

What is Terraform?

Terraform is an open-source Infrastructure as Code (IaC) tool by HashiCorp. It lets you define and provision infrastructure — network devices, cloud resources, VMs, services — using a declarative configuration language called HCL (HashiCorp Configuration Language).

■ **Declarative vs Imperative**

Declarative (Terraform): Describe the desired end state — "I want 3 routers with these settings" — Terraform figures out how to get there.

Imperative (scripting): Specify every step — create router, configure interface, add route... Declarative is simpler to maintain and less error-prone.

Key Terraform Concepts

IaC	Infrastructure as Code — config written in version-controlled files for repeatable, consistent deployments.
Provider	Plugin enabling Terraform to talk to a specific platform — AWS, Azure, GCP, Cisco, VMware, etc.
Resource	Individual infrastructure component — a router, subnet, VM, or firewall rule.
State File	Tracks what Terraform has deployed. Used to calculate what needs to change when config is modified.
Plan	Preview of changes before applying — shows exactly what will be created, modified, or destroyed.
Apply	Executes the plan and makes actual changes to infrastructure.
Destroy	Removes all infrastructure defined in the config — useful for labs and temporary environments.

Terraform vs Ansible

Both are agentless and automate configuration, but serve different primary purposes:

Feature	Terraform	Ansible
Primary Use	Provisioning infrastructure	Configuring existing devices/servers
Approach	Declarative	Procedural (YAML playbooks)
State File	Yes — tracks deployed state	No — runs tasks each time
Cloud Focus	Strong (AWS, Azure, GCP)	Good — more network/server focused
Language	HCL	YAML
Best For	Building infra from scratch	Managing and configuring existing systems

Terraform in Network Environments

• **Cloud networking:** Provisioning VPCs, subnets, routing tables, security groups in AWS/Azure/GCP.

• **Hybrid environments:** Managing on-premises and cloud resources with the same toolset.

• **Network devices:** Cisco Terraform providers enable configuration of physical and virtual devices.

• **Lab environments:** Spin up and destroy complete topologies on demand — "terraform destroy" removes everything cleanly.

Simple HCL Example (Recognition Only)

You will not write Terraform code on the exam, but recognising the syntax helps:

```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
  tags = { Name = "MainVPC" }
}

resource "aws_subnet" "primary" {
  vpc_id     = aws_vpc.main.id
  cidr_block = "10.0.1.0/24"
}
```

Defining a VPC and subnet in AWS using Terraform HCL — declarative style.

■ **Exam Tip:** Know that Terraform is declarative IaC, uses a state file, and is primarily for provisioning. Ansible is for configuring existing systems. Both are agentless. Terraform uses HCL; Ansible uses YAML. Puppet and Chef are no longer the exam focus.

Further Reading

- howtonetwork.com/technical/network-management/what-is-terraform/
- developer.hashicorp.com/terraform/docs
- registry.terraform.io/providers/CiscoDevNet

Quick Reference — CCNA v1.1 New Topics

Chapter 1 — AI & ML (Exam Obj 6.4)

Term	Definition
AI	Systems simulating human reasoning. Narrow AI = specific tasks; General AI = theoretical.
Machine Learning	AI subset — systems learn from data without explicit programming.
Supervised Learning	Trains on labelled data. Used for traffic classification and prediction.
Unsupervised Learning	Finds patterns in unlabelled data. Used for clustering and anomaly detection.
Reinforcement Learning	Learns via reward/penalty. Adaptive routing and load balancing.
Deep Learning	Multi-layer neural networks. Handles large unstructured data sets.
Predictive AI	Forecasts future events — maintenance, traffic volumes, security threats.
Generative AI	Creates new data/content — configs, test traffic, scenario simulation.
GANs	Generator + Discriminator networks producing realistic synthetic data.
Anomaly Detection	Learns baseline behaviour; flags deviations. Key ML security application.

Chapter 2 — Cloud-Managed Access (Exam Obj 2.8)

Term	Definition
Cloud Managed	Devices managed via cloud dashboard, not direct CLI. Example: Cisco Meraki.
Zero-Touch Provisioning	Device receives config automatically from cloud on first connection.
Management Plane	Depends on cloud/internet connectivity — may be unavailable if internet fails.
Data Plane	Continues forwarding traffic even if cloud management is unreachable.
Meraki Dashboard	Cisco's cloud management platform for switches, APs, firewalls, and cameras.

Chapter 3 — Terraform (Exam Obj 6.6)

Term	Definition
Terraform	Open-source IaC tool by HashiCorp. Declarative — describe desired state; Terraform provisions it.
IaC	Infrastructure as Code — version-controlled config files defining infrastructure.

HCL	HashiCorp Configuration Language — used to write Terraform configs.
Provider	Plugin enabling Terraform to interact with a platform (AWS, Azure, Cisco, etc.).
State File	Tracks deployed infrastructure. Used to calculate required changes.
Plan	Preview changes before applying — shows what will be created/modified/destroyed.
Declarative	Specify desired end state; the tool determines how to achieve it.
vs Ansible	Terraform = provision infrastructure. Ansible = configure existing systems. Both agentless.

© Paul Browning | www.howtonetwork.com | www.101labs.net
Free supplement for readers of Cisco CCNA in 60 Days and CCNA Simplified