CENGN Academy

Infrastructure as Code

As virtualization has become commonplace, various tools have emerged that treat virtualized infrastructure as code. This course introduces learners to the concepts behind Infrastructure as Code (IaC), the benefits of automating infrastructure deployment, and the tools that make it possible. Labs present real-world scenarios that build practical skills with Terraform, Chef, Puppet and Ansible. In particular, Terraform and Ansible are investigated and applied in depth.

Audience:
- Network Engineer, Team Lead or Manager
- Cloud Engineer, Team Lead or Manager
- Network or Product Manager

Delivery Mode: Learn on your own schedule with self-paced online training and labs

Duration: Learners will need approximately 20 hours to complete the course. Learners will have access to the online content and labs for 4 weeks

Hands-on Labs: This course features hands-on labs where learners provision and configure infrastructure within CENGN's private cloud infrastructure. Labs build on each other to demonstrate how multiple tools are used in real world scenarios

Recommended Prequisites:
- Basic understanding of DevOps (for example, CENGN Introduction to DevOps)
- Basic skills in programming and client-server-systems
- Intermediate level of experience with the Linux command line interface
- Intermediate understanding of virtualization
- Suggested – CENGN Docker & Kubernetes Basics

Learner Support: The CENGN Academy team of subject matter experts will be available to support you while you take this course. We will answer your questions, confirm your labs, and check in with you after your course to assist with your badge exam preparations

Exam and Digital Badge

Learners who complete this course are ready for the CENGN Infrastructure as Code exam. Those who successfully complete the exam will earn a CENGN Infrastructure as Code digital badge, which can be posted on LinkedIn and other social media

Course Objectives

After completing this course, the learner will be able to:
- Describe the Infrastructure as Code approach, along with its benefits and limitations
- Discuss IaC tool types and key characteristics
- List popular open-source and provider IaC tools, along with their typical use cases
- Explain fundamental Terraform concepts, including architecture, workflow model and state file
- Deploy a simple VM using Terraform
- Describe advanced Terraform features including meta-arguments, functions and backends
- Apply multiple Terraform features in a modular VM deployment
- Describe basic architecture, characteristics and features of Chef and Puppet
- Demonstrate basic hands-on skills with Chef and Puppet
- Explain fundamental Ansible concepts, including architecture, inventory, and modules
- Configure a virtualized web server using Ansible ad hoc commands
- Describe advanced Ansible concepts such as playbooks, facts, variables and roles
- Apply multiple Ansible concepts in a modular VM configuration task

Developed in collaboration with CloudOps

CENGN Academy

E: academy@cengn.ca
W: cengn.ca/services/cengn-academy

CENGN Academy

Infrastructure as Code

Course Content

Module 1 – Infrastructure as Code
- Recognize the problems with traditional infrastructure that led to the adoption of Infrastructure as Code
- Explain the core principles of Infrastructure as Code and how they aim to improve team workflow
- Describe the relationship between Infrastructure as Code and DevOps
- Describe common IT tasks that can be automated
- List popular Infrastructure as Code tools

Module 2 – Automation Tools
- Compare provisioning vs. configuration management tools
- Compare procedural vs. declarative languages
- Compare push/agent orientated tools vs. pull/agentless tools
- Explain other key IaC tool characteristics such as immutability, documentation and agnosticism
- Compare the architecture and functionality of popular open-source tools such as Terraform, Chef, Puppet and Ansible
- Describe cloud provider tools such as AWS CloudFormation, Azure ResourceManager, and Google Cloud Composer

Module 3 – Terraform
- Explain Terraform's development history and key characteristics
- List typical use cases for Terraform
- Summarize Terraform concepts such as modules, provisioners and resources
- Demonstrate Terraform's write/plan/apply workflow
- Apply Terraform's syntax, formatting and style
- Explain the purpose of Terraform's state file
- Create a simple VM deployment using Terraform

Module 4 – Terraform Advanced
- Describe advanced Terraform concepts such as meta-arguments and functions
- Explain the potential pain points of Terraform deployments
- Summarize how to effectively manage Terraform deployments with backends
- Create a modular Terraform deployment using advanced concepts and best practices

Module 5 – Chef
- Explain Chef's development history and key characteristics
- List typical use cases for Chef
- Define key concepts specific to Chef such as recipes, cookbooks and roles
- Define key Chef components such as the Chef Workstation and Server
- Apply Chef's syntax, formatting and style
- Give examples of how to effectively manage Chef deployments
- Implement Chef Workstation, Server and Client in a basic lab environment
- Configure a Chef Client as an Apache2 web server

Module 6 – Puppet
- Explain Puppets development history and key characteristics
- List typical use cases for Puppet
- Define key concepts specific to Puppet such as resources, manifests, and classes
- Define key Puppet components such as the Puppet Server and Client
- Apply Puppet's syntax, formatting and style
- Implement Puppet Server and Puppet Client in a basic lab environment
- Configure a Puppet Client with user accounts and selected services

Module 7 – Ansible
- Explain Ansible's development history and key characteristics
- List typical use cases for Ansible
- Identify concepts specific to Ansible such as inventory and modules
- Define key Ansible components such as the Ansible Controller
- Demonstrate Ansible ad-hoc commands to interact with Ansible Clients
- Configure an Ansible Client as an Apache2 web server using ad-hoc commands

Module 8 – Ansible Advanced
- Apply Ansible's syntax, formatting and style
- Describe advanced concepts such as playbooks and facts
- Describe Ansible variables such as dictionaries and lists
- Describe Ansible roles, including those publicly available through Ansible Galaxy
- Create a modular Ansible configuration using advanced concepts and best practices

CENGN Academy

E: academy@cengn.ca
W: cengn.ca/services/cengn-academy