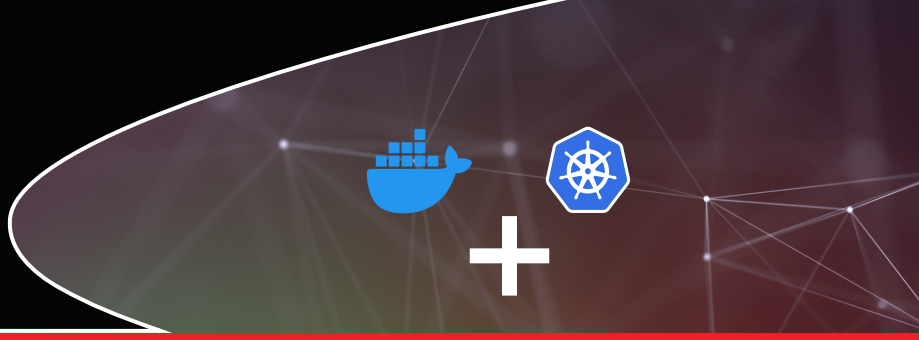
CENGN Academy
Docker & Kubernetes Advanced



Containerization and DevOps are revolutionizing the Information Technology world and introducing new paradigms into everything related to network-aware application development and deployment. This course continues on from CENGN's Docker & Kubernetes Basics course, taking the learner on an in-depth journey that starts with container creation, adds storage and networking, secures that container and configures monitoring for an Internet accessible Kubernetes-based service.

Audience:
- Software Engineer or Architect
- Network Engineer or Architect
- Cloud Engineer or Architect
- DevOps/DevSec Engineer

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

Duration: Learners will need approximately 25 hours to complete the course

Hands-on Labs: This course features hands-on labs using a 3-node cluster hosted on CENGN's multi-vendor cloud. The labs touch on all elements of container creation and deployment, leading to a full end-to-end lab that produces a proper web-accessible deployment using best practices.

Recommended Prequisites:
This course is best suited for learners with the following knowledge and skills
- Intermediate experience with networking
- Intermediate experience Linux, including command line interface
- Moderate level of understanding of virtualization
- Basic skills in containerization (CENGN Docker and Kubernetes Basics)
- Basic understanding of DevOps (CENGN Introduction to DevOps)

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

Course Objectives

After completing this course, the learner will be able to:
- Describe container runtime and network interface architectures.
- Explain how Docker, containerd, and Kubernetes interact with the Linux kernel
- Compare how Docker's Container Network Model (CNM) and Kubernetes' Container Network Interfaces (CNIs) build on core Linux networking functionality.
- Apply Docker/containerd functionality to Kubernetes concepts.
- Describe the Kubernetes Cluster architecture with all its components: Master Node, Worker Node, Pods, etc.
- Deploy different Kubernetes objects including workload resources, storage resources, access control resources and networking resources.
- Create a containerized application and deploy it in Kubernetes, using storage, networking, access control, and monitoring best practices
- Perform admin-level tasks for an existing application

Exam and Digital Badge

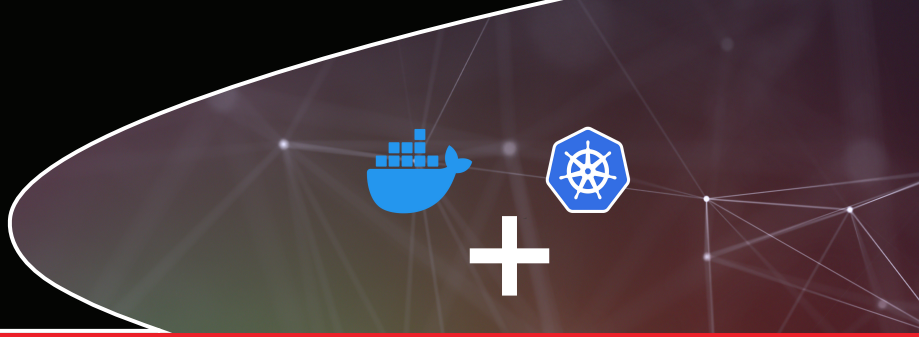
Learners who complete this course are ready for the CENGN Docker + Kubernetes Advanced exam. Those who successfully complete the exam will earn a CENGN Docker & Kubernetes Level 2 digital badge, which can be posted on LinkedIn and other social media

Developed in collaboration with CloudOps

CENGN Academy

To register for this course please email academy@cengn.ca
W: cengn.ca/services/cengn-academy

CENGN Academy
Docker & Kubernetes Advanced



Course Content

Module 1 – Container Recap
- Explain core Linux functionality used by container engines
- Describe container architecture and Docker engine capabilities
- Explain the relationship between container processes and host operating systems
- Compare and contrast runC, containerd and Docker
- Recognize the layers of abstraction in containerization
- Gain the ability to diagnose issues using the core functionality

Module 2 – Creating Containers
- Explain how a container is built, created, and executed
- Recognize how image layers are stored and used at runtime
- Explain how Linux kernel features such as cgroups and namespaces contribute in the process of creating containers
- Describe the kernel security in Docker and its support for namespaces and cgroups
- Build a base container image from scratch

Module 3 – Networking Containers
- Explain Docker's libnetwork and CNM constructs
- Set up networks using Docker and containerd

Module 4 – Kubernetes Recap
- Explain the basics of container orchestration and Kubernetes
- Describe the Kubernetes control plane, data plane, and their basic components
- Explain basic Kubernetes concepts including namespaces, Pods, labels, Services, etc.
- Describe the different Kubernetes Service types.
- Describe the Kubernetes Ingress resource type.
- Demonstrate essential ability working with kubectl, the Kubernetes CLI

Module 5 – Kubernetes Networking
- Define the fundamentals of the Kubernetes networking model
- Explain how the CNI model works and the role of each of its components
- Compare and contrast Kubernetes CNI and Docker CNM.
- Identify communication patterns within a Kubernetes Cluster: container-to-container, pod-to-pod, etc.

Module 6 – Storage and Workloads
- Explain the different storage options available for use in Kubernetes
- Create and deploy stateless and stateful applications with persistent storage
- Provision dynamic volumes using a StorageClass
- Describe the different Kubernetes Workload resources: Deployments, StatefulSets, ReplicaSets, DaemonSets, and Jobs

Module 7 – Network Policy and Service Mesh
- Identify the core components of networking in distributed systems
- Describe the Service Mesh architecture
- Describe the sidecar architecture and its capabilities

Module 8 – RBAC, Security, and Scheduling
- Identify tools for implementing RBAC in Kubernetes: Role, ClusterRole, RoleBinding, and ClusterRoleBinding
- Describe the mechanisms behind scheduling Pods on Nodes including Taints, Tolerations, labels, and selectors
- Identify methods to limit Cluster resources usage per namespace, resource, or deployment
- Explain when and how to implement CustomResourceDefinition resources
- Implement RBAC Roles and RoleBindings to apply permissions to ServiceAccounts

Module 9 – CI/CD with Kubernetes
- Describe different deployment strategies that can be implemented with Kubernetes including: RollingUpdate, Recreate, Blue-Green, and Canary
- Identify CI/CD tools available for use with Kubernetes
- Deploy a Kubernetes application using a Helm chart

Module 10 – Monitoring in Kubernetes
- Describe the challenges with monitoring in a distributed system, including aggregation and scaling
- Describe metrics, alerting, tracing, and logging
- Explain how distributed tracing helps in troubleshooting microservices
- Install a monitoring stack and a distributed tracing tool

Module 11 – End-to-end Lab

Module 12 – Course Quiz

CENGN Academy

To register for this course please email academy@cengn.ca
W: cengn.ca/services/cengn-academy