

Implementing DevOps Automatic CI/CD Pipeline with OpenAI, GitLab, Harbor, Jenkins, Kubernetes, Argo and Ansible

Course Outline

Overview

This is a comprehensive training course that provides a detailed understanding of the DevOps methodology and tools used in building efficient and automated Continuous Integration/Continuous Deployment (CI/CD) pipelines. The course is designed for IT and DevOps professionals who want to acquire the knowledge and skills to utilize the latest DevOps tools to automate the deployment, delivery, and testing of software applications.

In this course, students will learn how to use VS Code with OpenAI to create a web application, then use GitLab, Harbor, Jenkins, Ansible, Kubernetes, and Argo to automate the deployment of software applications. The course covers important topics such as version control systems, containerization, container orchestration, and infrastructure as code.

The course will provide students with the required knowledge and skills to automate the end-to-end software development lifecycle, from development to testing to deployment, using a combination of DevOps tools. By the end of the course, students will have the ability to design and implement DevOps pipelines using GitLab, Harbor, Jenkins, Ansible, Kubernetes, and Argo.

When you completed this training, you will be able to take the [Certified Kubernetes Application Developer \(CKAD\)](#) exam.

Duration: 5 days

Knowledge/Skills Requirements

For a successful learning experience, basic Linux command line and file editing skills are required. Completed Kenfil's [Kubernetes Administration – Part 1](#) and [Kubernetes Administration – Part 2](#) training.

About the Exam

CERTIFIED KUBERNETES APPLICATION DEVELOPER (CKAD)

The online, proctored, performance-based test consists of a set of performance-based items (problems) to be solved in a command line and is expected to take approximately two (2) hours to complete.

This exam curriculum includes these general domains and their weights on the exam:

DOMAIN	WEIGHT
Application Design and Build	20%
Application Deployment	20%
Application Observability and Maintenance	15%
Application Environment, Configuration and Security	25%
Services and Networking	20%

The cost is US\$395 and includes one free retake.

Quarterly exam updates are planned to match Kubernetes releases. Please see [the FAQ](#) for the current exam environment version.

Implementing DevOps Automatic CI/CD Pipeline with OpenAI, GitLab, Harbor, Jenkins, Kubernetes, Argo and Ansible

Course Outline

Course Outline

Module 1 - Understanding DevOps

- Automation in DevOps
- Types of automation used in DevOps
- DevOps Automation Tools
- Best practices for successful DevOps automation implementation

Module 2 - Jenkins

- Installing and configuring Jenkins
- Plugins and extensions
- Creating a Jenkins Job
- Running a job
- Introduction to Jenkins Pipeline
- Creating a Pipeline
- Pipeline syntax
- Pipeline stages and steps
- Jenkins Dashboard and Reporting
- Customizing Jenkins dashboard
- Jenkins reporting
- Jenkins metrics and analytics

Module 3 - Harbor for CI/CD Pipeline

- Introduction to Harbor
- Installation and Configuration of Harbor
- Create TLS for Harbor
- Harbor features and functionalities

Module 4 - Customize and Optimize Container Image

- Adding your own programs to the image
- Case study: Use OpenAI/Chatgpt4 with Python and add the APP into container image
- Passing arguments to an application in an image
- Running an application by default
- Optimizing builds

Module 5 - Kubernetes for DevOps

- Managing Multi-Container Pods
- Configure a Local registry server
- Using ConfigMaps to configure Containers
- Rolling Updates and Rollbacks to update application
- Configure SecurityContext for a Pod and Container
- Create and consume Secrets
- Working with ServiceAccounts
- Implement a NetworkPolicy for Pod Security
- Implement Service Mesh and Ingress Controller
- Add your application into Mesh
- Monitor Applications

Module 6 - GitLab DevOps

- Install GitLab Server
- GitLab connect Kubernetes
- Create a project containing a .gitlab-ci.yml file and register a GitLab Runner
- Create a basic CI configuration.
- Display pipeline environment info
- CI/CD Variable hierarchy
- Job policy pattern
- Using artifacts
- Accessing harbor self-hosted registry server

Module 7 - Visual Studio Code

- Using Git source control in VS Code
- DevOps GitLab Workflow
- DevOps GitLab Workflow with Merge Request
- Git integration
- Create two users in GitLab and assign permission
- Branching and merging
- Committing changes
- Pushing and pulling
- Installing extensions

Module 8 - ArgoCD

- Introduction to Argo
- Installation and Configuration of Argo
- Argo features and functionalities
- Argo CI/CD Pipeline

Module 9 - Ansible for Infrastructure as Code (IaC)

- Installation and Configuration of Ansible
- Setup Ansible Control Node
- Managed Nodes
- Ansible Playbooks
- Definition of Playbooks