

DevOps bootcamp for Expert

Duration: 12 days (8hrs/day)

Prerequisites:

- Basic knowledge of Linux Server Administration.

Course Objective: This comprehensive DevOps course, covering DevOps introduction, Continuous Integration (CI) and Continuous Deployment (CD) using Jenkins or Gitlab, – Infrastructure as Code (IaC) using Terraform, container management using Docker , container orchestration using Kubernetes, Collaboration and Communication using Git and GitHub and Observability using Prometheus is designed to equip learners with the skills needed to start working as DevOps engineer.

Tools: Latest version

Lab Requirement: Koenig-DC (<https://linuxlab.koenig-solutions.com>) + Student's Azure and GitHub account

Module 1 – Introduction to DevOps

Overview of DevOps

Introduction to DevOps

DevOps Tools and Ecosystem

Module 2 – Continuous Integration (CI) and Continuous Deployment (CD)

Ch:1 Jenkins

Introduction to Continuous Integration

Introduction to Jenkins

Jenkins Installation

Jenkins Management

Build Java Program

Run Jobs on Remote Machines

JUnit Testing

Graphical View of Tests

Saving Artifacts in Jenkins

Introduction to Jenkins Pipeline

Jenkins Pipeline with Maven

Lab: All above topics will be covered with Lab Demonstration

Module 3 – Infrastructure as Code (IaC): Terraform

Ch:1 - Getting Started & Setting Up Labs

Introduction to Infrastructure as Code and Terraform

Lab: Installation of Terraform on Windows

Comparison between Terraform and Ansible

Introduction to Azure CLI

Understanding Terraform Providers

Authenticate Azure with Terraform

Lab: Setting Up Terraform on Windows and Azure Authentication

Basic Terraform commands: init, plan, apply

Lab: Defining Provider & Using Basic Terraform commands

Ch:2 – Building Cloud Infrastructure with Terraform

Lab: Creating Resource Groups in Azure

Lab: Provisioning Virtual Networks, Subnets, Public IPs, and Network Interfaces

Lab: Deploying Windows and Linux VMs

Lab: Configuring Azure Storage, Security Groups, and Load Balancers

Understanding Terraform State file

Understanding Working of State file – Desired State & Current State

Terraform Provider Versioning

Lab: Methods to define Terraform Provider Versions

Ch:3 - Read, Generate, Modify Configurations

Understanding Attributes and Output Values in Terraform

Lab: Handling Terraform attributes and output values

Understanding Terraform Variables and Data Types – (List, Map)

Lab: Methods to Define Variables & Variable Arguments

Lab: Fetching Data from List & Map in Variables

Understanding Meta-Arguments – (count)

Lab: Using Meta-Arguments

Lab: Exploring debugging techniques in Terraform

Terraform Commands – validate, fmt

Lab: Using terraform validate and terraform fmt

Lab: Saving Terraform Plan to a file and apply from plan file

Ch:4 - Terraform Provisioners

Understanding provisioners in Terraform

Understanding Connection Block

Types of provisioners

Lab: Implementing file, remote-exec and local-exec provisioners

Ch:5 - Terraform Modules & Workspaces

Applying the DRY (Don't Repeat Yourself) principle

Understanding Usage of Terraform Modules

Standard Structure of Terraform Modules

Lab: Creating and Using local Modules

Lab: Utilizing Modules from Terraform Registry

Understanding and implementing Terraform workspaces

Lab: Working with Terraform Workspaces

Ch:6 – Terraform Cloud and Enterprise Overview

Introduction to Terraform Cloud

Creating infrastructure with Terraform Cloud

Overview of Sentinel Security in Terraform

Lab: Basic Deploying Infrastructure with Terraform Cloud and Sentinel Security

Module 4 – Containerization (Docker)

Ch:1 - Docker Administration

Introduction to Containers

Introduction to Docker

Docker Engine

Understanding Docker Images

Private Registry

Understand Storage Methods

Lab: Downloading and Installing Docker

Lab: Docker Essential Commands

Lab: Building Docker Images

Lab: Storing and Retrieving Docker Images from Docker Hub

Lab: Building Containers from Images

Lab: Networking Docker Containers

Lab: Data Persistence with Volumes

Lab: Linux Capabilities

Module 5 - Containerization Orchestration: Kubernetes

Ch:1 - Core Concepts

Overview of Container Orchestration

Introduction to Kubernetes

Kubernetes Architecture

Ch:2 – Installation, Configuration & Validation

Design a Kubernetes Cluster

Lab: Installation of Kubernetes 1-Master and 2-Nodes Cluster

Lab: Verify Installation with Kubectl command

Ch:3 - Managing Resources

Describe Pods

Describe Labels and Selector

Describe Replica Set

Describe Services

Describe Daemon Sets

Describe Namespaces

Lab: Managing Pods

Lab: Managing Labels & Selector

Lab: Managing Replica Set

Lab: Managing Cluster IP, Node Port

Lab: Installing Metal LoadBalancer and Using LoadBalancer Service

Lab: Managing Daemon Sets

Lab: Managing Namespaces

CH:4 - Application Lifecycle Management

Overview of Deployment

Deployment Strategies

Lab: Managing Deployment

Lab: Blue-Green Deployment Strategy

CH:5 - Environment Variable

Overview of Environment Variable

Lab: Plain Key as Variable in Pod

Lab: Config Map as Variable in Pod

Lab: Config Map as Volume

Lab: Secret as Variable in Pod

Lab: Secret as Volume

CH:6 - Storage

Describe Storage

Lab: Volumes

Lab: Creating Persistent Volume

Lab: Creating Persistent Volume Claim

CH:7 - Security

Kubernetes Authentication

Lab: Managing Users in Kubernetes

Lab: Service Account

Lab: Managing Roles and Role Binding

Lab: Managing Cluster Role and Cluster Role Binding

Lab: Basic Security Context

CH:8 - Logging and Monitoring

Understand how to Monitor all Cluster Components

Understand how to Monitor Applications

Lab: Read Cluster Component Logs

Lab: Using Elasticsearch and Kibana for Logging

Lab: Prometheus and Grafana Monitoring Tool

CH:9 - Networking in Kubernetes

Kubernetes Networking

Understand CNI

Lab: Configure and Manage Ingress Rule

Module 6 - Collaboration and Communication: Git and Github

Ch:1 Introduction to Git

Get started with Git

Install Git

Using command line

Configure Git

Git folder

Initialize git

Ch:2 Creating files/folder with Git

Add new files to Git

Git staging

Git commits

Git Help

Working with Git Branches

Ch:3 Introduction to GitHub

GitHub Account

Repository on GitHub

Push Local Repo to GitHub

Editing in GitHub

Pulling from GitHub

Pushing to GitHub

Lab: All topics will be covered with Lab demonstration

Module 7 – Observability: Prometheus and Grafana

Ch:1 Core Concepts of Prometheus:

Introduction to Prometheus

Metrics

Targets

Scraping

PromQL

Alerting

Lab: Install and explore Prometheus on Linux based machine

Ch:2 Core Concepts of Grafana:

Introduction to Grafana

Data Sources

Dashboards

Queries

Alerting

Lab: Install and explore Grafana on Linux based machine

Ch:3 Integrating Prometheus with Grafana:

Prometheus as a data source

visualize and analyze metrics collected by Prometheus.

create custom dashboards displaying metrics collected by Prometheus.

Lab: Integrating Prometheus with Grafana

Ch:4 Creating Dashboards in Grafana:

create custom dashboards in Grafana to visualize metrics using various visualization options such as graphs, charts, tables, and gauges.

customized with annotations, templating, and annotations to provide context and insights into the data.

Lab: Create and customize Grafana dashboard.