Docker, Kubernetes and Terraform Course Contents (5 days)
By Dr. Vishwanath Rao

Prerequisites:

- **Basic Linux Command Line Skills:** Participants should have familiarity with the Linux command line interface,
- including basic commands for navigation, file manipulation, and text editing.
- **Understanding of Cloud Concepts:** A basic understanding of cloud computing concepts such as virtualization,
- networking, and storage will be beneficial.
- **Programming Basics:** Familiarity with at least one programming language (e.g., Python, JavaScript) will help in
- understanding scripting and automation aspects covered in the course.
- **Basic Knowledge of DevOps:** Participants should have a general understanding of DevOps principles and practices,
- including continuous integration and continuous deployment (CI/CD) concepts.

Course Objectives:

- **Understand Containerization Concepts:**
  - Explain the benefits of containerization and understand the differences between containers and virtual machines.
- **Master Docker Fundamentals:**
  - Demonstrate proficiency in building, running, and managing Docker containers.
  - Create Docker images and Dockerfiles for custom applications.
- **Learn Terraform for Infrastructure Provisioning:**
  - Understand Infrastructure as Code (IaC) principles and the role of Terraform.
  - Provision infrastructure resources using Terraform and manage state files effectively.
- **Deploy Kafka on Kubernetes:**
  - Deploy Apache Kafka clusters on Kubernetes using Helm charts.
  - Configure Kafka topics, brokers, and consumers within a Kubernetes environment.
- **Explore Helm Charts:**
  - Understand the role of Helm in Kubernetes application management.

  - Create, package, and deploy applications using Helm charts.
- **Master Advanced Kubernetes Concepts:**
  - Gain proficiency in advanced Kubernetes networking, scaling, and security features.
  - Learn deployment strategies like canary deployments and blue-green deployments.
- **Hands-on Experience:**
  - Gain practical experience through hands-on labs and exercises on Docker, Kubernetes, and Terraform.
  - Apply learned concepts to real-world scenarios and solve deployment challenges.
- **Troubleshooting and Debugging Skills:**
  - Develop skills for troubleshooting and debugging issues in Docker, Kubernetes, and Terraform deployments.
  - Understand best practices for maintaining and monitoring containerized environments.
- **Practical Application:**
  - Apply learned concepts to design, deploy, and manage infrastructure and applications in cloud-native environments.
  - Prepare participants for real-world DevOps roles involving Docker, Kubernetes, and Terraform.
- **Certification Preparation:**
  - Provide participants with the knowledge and skills required to pursue certifications in Docker, Kubernetes, and Terraform.
  - Equip participants with the confidence to tackle certification exams and excel in their DevOps careers.

Day 1: Introduction to Docker and Containerization

- **Understanding Containerization Concepts:**
  - Definition and benefits of containerization.
  - Comparison between virtual machines and containers.
- **Docker Fundamentals:**
  - Docker architecture overview: Docker Engine, images, containers, etc.
  - Dockerfile basics: writing Dockerfiles to create custom images.
- **Hands-on Docker Installation:**
  - Installing Docker on various operating systems.
  - Configuring Docker environment and basic settings.
- **Building Docker Images:**
  - Creating Docker images from Dockerfiles.
  - Tagging and versioning Docker images.
- **Running Containers:**
  - Starting, stopping, and managing Docker containers.

    - Understanding container lifecycle.
- **Docker Networking and Volumes:**
    - Networking concepts in Docker.
    - Managing data persistence with Docker volumes.
- **Docker Compose:**
    - Introduction to Docker Compose for multi-container applications.
    - Writing Docker Compose files to define application stacks.
- **Best Practices and Security Considerations:**
    - Docker security best practices.
    - Strategies for optimizing Docker images and containers.
- **Lab Session 1: Docker Basics:**
    - Exercises to practice Docker commands and container management.
- **Q&A Session:**
    - Addressing questions and clarifications on Docker fundamentals.
- **Group Discussion:**
    - Sharing experiences and insights on Docker adoption in real-world scenarios.
- **Review and Summary:**
    - Recap of key concepts covered during the day.
    - Preparation for the next day's topics.
- **Assignment:**
    - Prepare a Dockerfile for a sample application and create a Docker image.

Day 2: Terraform for Infrastructure Provisioning

- **Introduction to Infrastructure as Code (IaC):**
    - Overview of IaC principles and benefits.
    - Introduction to Terraform as an IaC tool.
- **Terraform Basics:**
    - Terraform configuration files (main.tf, variables.tf, etc.).
    - Terraform providers and resources.
- **Setting up Terraform Environment:**
    - Installing Terraform CLI.
    - Initializing a Terraform workspace.
- **Managing Terraform State:**
    - Understanding Terraform state files.
    - Using remote backends for state management.
- **Creating Infrastructure Resources:**
    - Defining infrastructure components using Terraform configuration.
    - Provisioning resources like virtual machines, networks, and storage.
- **Terraform Modules:**

    - Creating reusable infrastructure components with Terraform modules.
    - Organizing Terraform code for scalability and maintainability.
- **Variable Management in Terraform:**
    - Using variables and data sources in Terraform configurations.
    - Best practices for managing Terraform variables.
- **Terraform State Management and Locking:**
    - Implementing state locking for Terraform deployments.
    - Strategies for managing and sharing Terraform state files in a team environment.
- **Lab Session 2: Terraform Hands-on:**
    - Exercises to create and manage infrastructure resources using Terraform.
- **Q&A Session:**
    - Clarifying doubts and queries related to Terraform usage.
- **Group Discussion:**
    - Sharing experiences and challenges faced during Terraform deployments.
- **Review and Summary:**
    - Recap of Terraform fundamentals covered during the day.
    - Preparation for Kafka deployment on Kubernetes on the following day.
- **Assignment:**
    - Design and provision a sample infrastructure setup using Terraform.

Day 3: Kafka Deployment on Kubernetes Cluster

- **Introduction to Apache Kafka:**
    - Overview of Apache Kafka and its use cases.
    - Kafka architecture: brokers, topics, partitions, etc.
- **Kubernetes Fundamentals:**
    - Recap of Kubernetes basics: nodes, pods, deployments, services.
    - Understanding Kubernetes networking and storage.
- **Installing Kubernetes Cluster:**
    - Setting up a Kubernetes cluster using tools like Minikube or kind.
    - Configuring Kubernetes cluster for development and testing.
- **Introduction to Kafka on Kubernetes:**
    - Overview of deploying Kafka on Kubernetes.
    - Challenges and considerations for Kafka deployment in Kubernetes.
- **Deploying Kafka with Helm Charts:**
    - Introduction to Helm for package management in Kubernetes.
    - Installing and configuring Kafka using Helm charts.

- **Configuring Kafka Topics and Brokers:**
  - Creating Kafka topics.
  - Scaling Kafka brokers and partitions in Kubernetes.
- **Kafka Consumer and Producer Setup:**
  - Deploying Kafka consumer and producer applications on Kubernetes.
  - Testing Kafka message publishing and consumption.
- **Monitoring and Logging Kafka on Kubernetes:**
  - Monitoring Kafka cluster health in Kubernetes.
  - Setting up logging for Kafka components.
- **Lab Session 3: Kafka Deployment Hands-on:**
  - Exercises to deploy and manage Kafka on Kubernetes.
- **Q&A Session:**
  - Addressing questions related to Kafka deployment and Kubernetes integration.
- **Group Discussion:**
  - Sharing experiences and insights on Kafka deployment challenges.
- **Review and Summary:**
  - Recap of Kafka deployment concepts covered during the day.
  - Preparation for Helm charts exploration on the following day.
- **Assignment:**
  - Deploy a Kafka cluster on a Kubernetes cluster and test producer-consumer interactions.

Day 4: Helm Charts

- **Introduction to Helm:**
  - Overview of Helm and its role in Kubernetes package management.
  - Benefits of using Helm for managing Kubernetes applications.
- **Understanding Helm Charts:**
  - Anatomy of a Helm chart: Chart.yaml, values.yaml, templates, etc.
  - Helm chart repositories and versioning.
- **Helm Installation and Configuration:**
  - Installing Helm CLI.
  - Initializing Helm client and configuring repositories.
- **Creating Custom Helm Charts:**
  - Building Helm charts for custom applications.
  - Defining chart dependencies and configurations.
- **Packaging and Distributing Helm Charts:**
  - Packaging Helm charts for distribution.
  - Publishing Helm charts to chart repositories.
- **Deploying Applications with Helm:**
  - Using Helm to deploy applications on Kubernetes.
  - Managing Helm releases and upgrades.

- **Helm Chart Templating and Values:**
  - Templating with Go templates in Helm charts.
  - Configuring Helm chart values for different environments.
- **Helm Chart Hooks and Post-Install Actions:**
  - Implementing hooks for executing actions during Helm chart lifecycle.
  - Post-install actions for configuring applications after deployment.
- **Lab Session 4: Exploring Helm Charts:**
  - Hands-on exercises to create, package, and deploy applications using Helm charts.
- **Q&A Session:**
  - Clarifying doubts and queries related to Helm chart usage.
- **Group Discussion:**
  - Sharing experiences and challenges faced in Helm chart usage.
- **Review and Summary:**
  - Recap of Helm chart fundamentals covered during the day.
  - Preparation for advanced Kubernetes topics on the final day.
- **Assignment:**
  - Create a Helm chart for deploying a sample application and test deployment on a Kubernetes cluster.

Day 5: Advanced Kubernetes Concepts

- **Advanced Kubernetes Networking:**
  - In-depth understanding of Kubernetes networking concepts.
  - Service mesh with tools like Istio.
- **Kubernetes Operators:**
  - Introduction to Kubernetes operators for managing complex applications.
  - Deploying and managing operators in Kubernetes.
- **Scaling and Autoscaling in Kubernetes:**
  - Horizontal and vertical scaling in Kubernetes.
  - Autoscaling based on metrics and resource usage.
- **Persistent Storage in Kubernetes:**
  - Persistent volume concepts in Kubernetes.
  - Using storage classes and dynamic provisioning.
- **Kubernetes Security Best Practices:**
  - Security considerations for Kubernetes deployments.
  - Implementing RBAC and network policies.
- **Advanced Deployment Strategies:**
  - Canary deployments and blue-green deployments in Kubernetes.
  - Traffic splitting and testing strategies.
- **Monitoring and Observability Tools:**
  - Exploring advanced monitoring and observability solutions for

Kubernetes.

  - Implementation of custom metrics and dashboards.
- **Troubleshooting and Debugging in Kubernetes:**
  - Strategies for troubleshooting common issues in Kubernetes.
  - Debugging techniques for Kubernetes applications.
- **Lab Session 5: Advanced Kubernetes Concepts:**
  - Hands-on exercises to explore advanced Kubernetes features and concepts.
- **Q&A Session:**
  - Addressing questions and clarifications on advanced Kubernetes topics.
- **Group Discussion:**
  - Sharing experiences and insights on advanced Kubernetes deployments.
- **Final Review and Summary:**
  - Recap of advanced Kubernetes concepts covered during the day.
  - Reflection on the entire course and key takeaways.
- **Course Conclusion and Certification:**
  - Certificate distribution and course feedback.
  - Next steps and resources for continuous learning in Docker, Kubernetes, and Terraform.