**DevOps Course outline 180 hours**

**Docker:**

Course Overview: This course provides an introduction to Docker, a popular containerization platform. Participants will learn the fundamentals of Docker, including containerization concepts, Docker architecture, Dockerfile creation, Docker networking, and container orchestration with Docker Compose.

Course Duration: 40 hours

Course Outline:

1. Introduction to Docker (2 hours)
   * Introduction to containerization and its benefits
   * Overview of Docker and its key features
   * Docker use cases and industry adoption
2. Docker Installation and Setup (3 hours)
   * Docker installation on different platforms (Windows, macOS, Linux)
   * Docker CLI basics and Docker engine architecture
   * Configuring Docker settings and managing Docker resources
3. Docker Images and Containers (5 hours)
   * Understanding Docker images and containers
   * Docker Hub and image repositories
   * Creating and running containers from Docker images
   * Container lifecycle management and troubleshooting
4. Docker Networking (5 hours)
   * Docker networking modes and concepts
   * Exposing container ports and mapping host ports
   * Creating and managing Docker networks
   * Linking containers and service discovery
5. Docker Volumes and Data Management (5 hours)
   * Managing data in Docker containers
   * Docker volumes and persistent storage
   * Sharing data between containers
   * Backing up and restoring Docker volumes
6. Dockerfile and Image Creation (5 hours)
   * Dockerfile syntax and best practices
   * Creating custom Docker images
   * Layer caching and optimizing image builds
   * Pushing and pulling Docker images to/from registries
7. Docker Compose and Container Orchestration (5 hours)
   * Introduction to container orchestration
   * Docker Compose and its capabilities
   * Defining multi-container applications with Docker Compose
   * Scaling and managing application services with Docker Compose
8. Docker Best Practices and Advanced Topics (5 hours)
   * Docker security best practices
   * Containerization strategies and microservices
   * Docker swarm mode and cluster management
   * Docker ecosystem and advanced tooling
9. Project - Flask Web Application with Docker (5 hours)
   * This project aims to containerize a Flask web application using Docker. The Flask web application will provide a simple API endpoint that returns a JSON response. Docker will be used to create a containerized environment for running the application.

**GitLab:**

Course Overview: This course provides a comprehensive introduction to GitLab, an open-source platform for version control, CI/CD pipelines, and collaboration. Participants will learn the core concepts and features of GitLab, including repository management, branching and merging, CI/CD configuration, issue tracking, and collaboration tools.

Course Duration: 25 hours

Course Outline:

1. Introduction to GitLab (2 hours)
   * Introduction to version control systems
   * Overview of GitLab and its key features
   * Understanding the GitLab interface and navigation
2. GitLab Installation and Setup (4 hours)
   * GitLab installation options (self-hosted or cloud-based)
   * Initial setup and configuration of GitLab
   * User management and permissions in GitLab
3. Git Basics and Repository Management (4 hours)
   * Introduction to Git and its fundamental concepts
   * Creating and cloning Git repositories in GitLab
   * Working with branches and tags in GitLab
4. Collaborating with GitLab (3 hours)
   * Managing merge requests and code reviews
   * Collaborating with teammates using issues and discussions
   * Using GitLab's integrated chat and activity feeds
5. GitLab CI/CD Fundamentals (4 hours)
   * Introduction to Continuous Integration and Continuous Deployment (CI/CD)
   * Configuring CI/CD pipelines in GitLab
   * Defining stages, jobs, and runners in GitLab CI/CD
6. GitLab CI/CD Configuration (4 hours)
   * Writing GitLab CI/CD configuration files (**.gitlab-ci.yml**)
   * Building, testing, and deploying applications with GitLab CI/CD
   * Integration with external services and environments
7. GitLab CI/CD Advanced Features (2 hours)
   * Caching and artifact management in CI/CD pipelines
   * Parallel and distributed testing with GitLab CI/CD
   * Customizing CI/CD workflows and pipeline triggers
8. GitLab Best Practices and Administration (2 hours)
   * GitLab workflows and best practices
   * Repository maintenance and backups
   * Monitoring and troubleshooting GitLab instances

**Ansible**

Course Overview: This course provides a comprehensive introduction to Ansible, an open-source automation platform. Participants will learn the core concepts of Ansible, including playbook creation, inventory management, task automation, roles, and Ansible Tower.

Course Duration: 40 hours

Course Outline:

1. Introduction to Ansible (2 hours)
   * Introduction to automation and Ansible's role
   * Overview of Ansible architecture and components
   * Understanding Ansible's YAML syntax and modules
2. Ansible Installation and Setup (3 hours)
   * Installing Ansible on different platforms (Linux, Windows, macOS)
   * Configuring the Ansible control node
   * Setting up SSH access and key-based authentication
3. Ansible Inventories (3 hours)
   * Understanding Ansible inventories and host groups
   * Inventory formats (INI and YAML)
   * Dynamic inventories and external inventory sources
4. Ansible Playbook Basics (5 hours)
   * Creating and running Ansible playbooks
   * Writing tasks and defining hosts
   * Ansible modules and common module examples
   * Using variables and facts in playbooks
5. Ansible Playbook Advanced Features (5 hours)
   * Conditionals and loops in playbooks
   * Handling errors and retries
   * Using roles for playbook organization and reusability
   * Ansible vault for securing sensitive data
6. Managing Configuration Files with Ansible (5 hours)
   * Managing configuration files using templates
   * Handling variables and dynamic content in templates
   * Deploying and managing configuration files with Ansible
7. Ansible Tower Overview (2 hours)
   * Introduction to Ansible Tower
   * Ansible Tower features and benefits
   * Installing and configuring Ansible Tower
   * Creating and managing projects, inventories, and job templates
8. Ansible Tower Advanced Features (4 hours)
   * Role-based access control (RBAC) and user management
   * Using workflows and job templates
   * Ansible Tower's API and integration options
   * Monitoring and reporting with Ansible Tower
9. Ansible Best Practices and Troubleshooting (3 hours)
   * Ansible best practices and style guide
   * Troubleshooting common issues and errors
   * Debugging playbooks and tasks
   * Monitoring and logging Ansible operations
10. Project (8 hours)
    * Project Title: Infrastructure Provisioning and Configuration Management with Ansible Project Overview: This project aims to automate infrastructure provisioning and configuration management using Ansible. Ansible will be used to define and manage the desired state of infrastructure resources and configure them according to specific requirements. The project will include the setup of a web server on multiple target machines and the configuration of a load balancer to distribute traffic.

**Kubernetes:**

Course Overview: This course provides a comprehensive introduction to Kubernetes, an open-source container orchestration platform. Participants will learn the core concepts and components of Kubernetes, including pod management, service discovery, scaling, deployments, and monitoring.

Course Duration: 45 hours

Course Outline:

1. Introduction to Kubernetes (3 hours)
   * Introduction to containerization and container orchestration
   * Overview of Kubernetes and its key features
   * Understanding Kubernetes architecture and components
2. Kubernetes Installation and Setup (3 hours)
   * Installing Kubernetes on different platforms (local, cloud, managed services)
   * Configuring a Kubernetes cluster
   * Managing cluster resources and access control
3. Kubernetes Pods and Containers (5 hours)
   * Understanding Kubernetes pods and containers
   * Pod management and lifecycle
   * Creating and configuring pods using YAML manifests
   * Pod scheduling and resource allocation
4. Kubernetes Services and Networking (5 hours)
   * Service discovery and load balancing
   * Cluster networking and communication
   * Configuring Kubernetes services
   * Ingress controllers and routing
5. Kubernetes Deployments and ReplicaSets (5 hours)
   * Managing application deployments
   * Scaling applications with replicas and ReplicaSets
   * Rolling updates and rollback strategies
   * Managing application state and data persistence
6. Kubernetes ConfigMaps and Secrets (3 hours)
   * Managing configuration data with ConfigMaps
   * Securing sensitive data with Secrets
   * Injecting configuration into pods and containers
7. Kubernetes Persistent Volumes and Storage (5 hours)
   * Storage options in Kubernetes
   * PersistentVolumes and PersistentVolumeClaims
   * Dynamic provisioning and storage classes
   * Managing data in stateful applications
8. Kubernetes Operators and Custom Resource Definitions (2 hours)
   * Introduction to Kubernetes Operators
   * Creating custom resource definitions (CRDs)
   * Implementing operators for application-specific workflows
9. Kubernetes Monitoring and Logging (5 hours)
   * Monitoring cluster and application health
   * Metrics, logging, and events in Kubernetes
   * Using Prometheus and Grafana for monitoring
   * Troubleshooting and debugging in Kubernetes
10. Kubernetes Best Practices and Advanced Topics (4 hours)
    * Kubernetes best practices and design considerations
    * Autoscaling and resource optimization
    * Deploying applications with Helm charts
    * Managing multi-cluster environments
11. Project Title: Kubernetes Deployment of a Node.js Application (5 hours)  
    Project Overview: This project aims to deploy a Node.js application using Kubernetes. The Node.js application will be containerized using Docker and deployed to a Kubernetes cluster. Kubernetes manifests will be created to define the deployment, service, and ingress resources for the application.

**Jenkins:**

Course Overview: This course provides a comprehensive introduction to Jenkins, an open-source automation server. Participants will learn the core concepts and features of Jenkins, including continuous integration, pipeline creation, job configuration, plugin usage, and distributed builds.

Course Duration: 15 hours

Course Outline:

1. Jenkins Installation and Setup (5 hours)
   * Introduction to continuous integration and Jenkins
   * Installing Jenkins on different platforms (Windows, macOS, Linux)
   * Initial setup and configuration of Jenkins
   * Managing Jenkins plugins and extensions
2. Building and Running Jobs in Jenkins (5 hours)
   * Creating and configuring Jenkins jobs
   * Configuring source code repositories and triggers
   * Running and scheduling Jenkins jobs
   * Analyzing build results and reports
3. Jenkins Pipeline Automation (5 hours)
   * Integrating Jenkins with version control systems (Git, SVN)
   * Implementing build automation and continuous integration
   * Implementing deployment automation and continuous delivery
   * Configuring Jenkins pipeline triggers and notifications

**Linux and Virtual Machine (VM):**

Course Duration: 5 hours

Module 1: Introduction and installation VM

* Overview of virtualization and its benefits
* Types of virtualization: full virtualization, paravirtualization, and hardware-assisted virtualization
* Installing and configuring a hypervisor (e.g., VMware ESXi, Microsoft Hyper-V)

Module 2: Linux installation and basic command

* Linux installation on VM
* Basic Linux commands and shell navigation
* Creating and managing user accounts
* User authentication and password policies
* Managing user groups and file permissions
* Sudo and privilege escalation
* Working with files and directories
* File permissions and ownership
* Package management systems (e.g., apt, yum)
* Installing, updating, and removing software packages

Complete CI/CD project (15 hours)

* Project Title: CI/CD Pipeline Implementation with Jenkins and GitLab  
  Project Overview: This project aims to implement a complete CI/CD pipeline using Jenkins and GitLab. The pipeline will automate the build, testing, and deployment processes for a software application. By integrating Jenkins with GitLab, developers will be able to trigger pipeline executions on code changes and ensure a seamless software delivery workflow.  
  Project requirements: VM installation, Ubuntu installation, Git lab installation, Jenkins installation, Php Laravel environment creation.