DEVOPS TEST AUTOMATION COURSE CONTENT

Introduction to DevOps and Test Automation (6 hours LIVE + 2 hours RECORDED)

Overview of DevOps and its importance in software development
Overview of Test Automation and its role in DevOps
Key concepts, terminologies and tools used in DevOps Test Automation

Linux Fundamentals (6 hours LIVE + 2 hours RECORDED)

Overview of Linux and its importance in DevOps
Basic Linux commands and operations
Introduction to Linux Shell and Scripting
Hands-on exercises to reinforce learning
Takeaways: Familiarity with Linux environment, basic administration tasks

Maven (6 hours LIVE + 2 hours RECORDED)

Introduction to Maven and its role in software build management
Understanding Maven POM and goals, lifecycle, plugins and its structure
Setting up Maven Selenium and RestAssured projects, managing dependencies and plugins
Hands-on exercises with Maven testing build and deployment
Use case: Build Independent and Parent – Child Maven Selenium Projects
Takeaways: Understanding of Maven and its role in build automation for testing projects

Git (6 hours LIVE + 2 hours RECORDED)

Introduction to Git and its role in version control
Understanding Git concepts such as branching, merging and pull requests
Setting up a Git repository, committing and pushing changes
Hands-on exercises with Git operations and workflow
Use case: Collaborate with a team of developers in a Git environment
Takeaways: Understanding of Git and its role in version control and collaboration

Jenkins (12 hours LIVE + 4 hours RECORDED)

Introduction to Jenkins and its role in continuous integration and delivery
Setting up Jenkins servers and jobs
Integrating Jenkins with Git, Maven and other tools
Hands-on exercises with Jenkins pipelines and automation
Use case: Automate the build and deployment of a Selenium using Jenkins and Maven
Takeaways: Understanding of Jenkins and its role in continuous integration and deployment

Project 1 (6 hours)

The hands-on project uses Selenium for UI testing, REST Assured for API testing, Maven as the build tool, Git for version control, and Jenkins for continuous integration and deployment. This setup ensures the stability and security of the application to be tested by regularly testing and integrating code changes.

AWS (12 hours LIVE + 4 hours RECORDED)

Introduction to AWS and its role in cloud computing
Understanding AWS services and infrastructure
Setting up AWS instances and deploying applications
Hands-on exercises with AWS cloud computing and management
Use case: Setup test environment in AWS Cloud
Takeaways: Understanding of AWS Cloud and its role in setting up test infrastructure

Docker (12 hours LIVE + 4 hours RECORDED)

Introduction to Docker and its role in containerization
Understanding Docker images, containers and networking
Setting up Docker containers and deploying applications
Hands-on exercises with Docker operations and orchestration
Use case: Deploy selenium and framework in a Docker environment
Takeaways: Understanding of Docker and its role in containerization and deployment

Kubernetes (12 hours LIVE + 4 hours RECORDED)

Introduction to Kubernetes and its role in container orchestration
Understanding Kubernetes components such as pods, services and deployments
Setting up and managing Kubernetes clusters
Hands-on exercises with Kubernetes operations and management
Use case: Scale automated tests in Kubernetes environment
Takeaways: Understanding of Kubernetes and its role in orchestration and deployment

Prometheus (6 hours LIVE + 2 hours RECORDED)

Introduction to Prometheus and its role in monitoring and alerting
Understanding Prometheus data sources, metrics and alerts
Setting up Prometheus for monitoring applications and infrastructure
Hands-on exercises with Prometheus monitoring and alerting
Use case: Monitor automated test executions using Prometheus
Takeaways: Understanding of Prometheus and its role in monitoring and alerting

Terraform (6 hours LIVE + 4 hours RECORDED)

Introduction to Terraform and its role in infrastructure as code
Understanding Terraform configuration files and state management
Setting up Terraform for infrastructure provisioning and management
Hands-on exercises with Terraform operations and automation
Use case: Automate the creation and management of infrastructure resources in AWS
Takeaways: Understanding of Terraform and its role in Infrastructure as a code (IaC)

End to End Project(12 hours)

A cloud-based react application deployed on AWS. The automation project uses Selenium for UI testing, REST Assured for API testing, Maven as the build tool, Git for version control, and Jenkins for continuous integration and deployment. Terraform is used to manage infrastructure as code, while Prometheus is used for monitoring and logging. This setup provides a scalable, secure, and automated testing solution for the automation tests, ensuring seamless delivery and improved reliability.