DevOps Course for Cognyte
Microservices-Docker-Kubernetes

36 hours

Required Skills

- GIT
- Basic programming skills

Course Contents

Basic DevOps (12)

1. Introduction to DevOps

2. Docker

- Docker Architecture overview
- Docker concepts:
    - Volumes
        - Deme/lab
    - Networks
        - Demo/lab
- Docker Advance
    - Dockerfile best practises
    - Multi-Stage build
        - Demo/Lab
- Docker-compose.yml
    - Demo/Lab

3. CI/CD - Jenkins

- Introduction to CI/CD.
- Jenkins up and running (Docker based)
    - Demo/Lab

4. Security in DevOps

- Understanding the principles of DevSecOps.
- Implementing security best practices in the DevOps pipeline.
- Integrating security tools and practices such as vulnerability scanning, code analysis, and access control.

5. Cloud-Native

- Exploring cloud-native architectures and their relationship with DevOps.
- Containerization and microservices architecture using Docker and container orchestration tools like Kubernetes.
- Implementing serverless computing and leveraging cloud-native services.
- Guidelines\ how to migrate application to CN architecture \ best practices.
- Migration of data and storages tools and best practices

Advanced DevOps (24)

1. Kubernetes

- Kubernetes Architecture
    - Cluster Concepts, Design and Node Roles
    - Kubernetes Distributions comparison
- Kubernetes objects:
    - Pods
    - Services
    - Labels
    - Namespaces, Annotations
    - Deployments
    - DaemonSets
    - ReplicaSets
    - Jobs
    - ConfigMaps and Secrets
        - Demos/Labs (For each K8S object)
- Kubernetes Volumes
    - Demo/Lab
- Stateless Set vs Stateful Set
    - Demos/Labs
- Handling the Kubernetes Package Manager - Helm

    - Using Helm
        - Demos/Labs
    - Helm Commands
    - Charts
        - Demos/Labs
    - Templates
    - Best Practice
- Traffic management with istio.
    - Demos/Labs

2. **OpenShift \ Kubevirt**
3. **Observability and Monitoring**
    - Monitoring techniques using tools like Prometheus, Grafana, or Datadog.
    - Distributed tracing and advanced log analysis for troubleshooting and performance optimization.
    - Implementing anomaly detection and predictive analytics for proactive monitoring.
4. **Performance Optimization**
    - Performance testing and optimization techniques for DevOps environments.
    - Advanced techniques for optimizing application and infrastructure performance.
    - Implementing auto-scaling and load balancing strategies for high-traffic applications.