Unified Assurance V6 - Microservice Fundamentals Ed 1 LVC

Oracle Communications

DURATION	MODULES	COURSE CODE
2 Days	**4 Lectures**	**—**

Course Overview

Microservices are the latest addition to the Unified Assurance Hyperscale Architecture. The central idea behind microservices is that some types of applications become easier to build and maintain when they are broken down into smaller, composable pieces which work together, and each component is continuously developed and separately maintained. This then makes the Unified Assurance application the sum of its constituent components, which contrasts with traditional, monolithic or service-oriented applications which are all developed all in one piece. The course will cover the following topics: Why Microservices? What are Microservices within Unified Assurance? Characteristics of Microservices Containers Legacy SOA Verses Microservices Architecture Microservice Horizontal Scaling Microservices Architecture Overview Docker Container Orchestration & Kubernetes Helm & Chart Museum Hyperscale Clustering Kubernetes Monitoring Pulsar Message Bus Microservice Data Pipelines, Trap and Syslog Installing Microservices Event Pipeline Microservice Troubleshooting

What You Will Learn

Module 1: Microservices Overview

- Objectives
- Why Microservices: The 12 Factors of Modern Application Development
- What are Microservices within Unified Assurance?
- Characteristics of Microservices
- Containers: Evolution of Computing
- From Servers to Containers
- Container vs. Virtual Machine
- Advantages of Containers
- Example Use Case
- Legacy SOA vs Microservices Architecture
- How SOA Apps Work
- Problems with SOA Applications and Scaling
- Scaling Solved with Microservices & Horizontal Scaling
- Recap

Module 2: Microservice Architecture & Software Components

- Objectives
- Oracle Communications Unified Assurance Documentation
- Microservices Architecture Overview
- Docker
- What is Docker?

Docker Registry, Docker Image Pipeline, Docker Image, Docker File, Docker Container

- Docker Linux Processes
- Container Orchestration: Kubernetes
- Key Roles and Tasks
- Kubernetes Architecture
- Cluster: Master Node & Worker Nodes
- Kubernetes Objects: Pods, Services, Namespaces
- Helm & ChartMuseum
- Hyperscale Clustering
- Unified Assurance KEDA Microservice
- Kubernetes Monitoring
- Pulsar Message Bus
- Architecture: Broker, Producers, Consumers, Topics
- Microservice Data Pipelines

Trap Pipeline, Trapd Extended Pipeline, Syslog Pipeline, Syslog Extended Pipeline

- Event Rules: trap-collector, js-generic-event-processor, syslog-collector
- Recap

Module 3: Implementing Microservices

- Objectives
- Microservice Cluster Setup: Single Server
- Command Line Aliases & Configuration CLI Commands
- Cluster Setup Dependencies & Roles
- Creating Kubernetes Cluster: Single Server
- Default Namespaces and Pods
- Helm Packages
- Updating Helm Repository

Installing Microservices: Event Pipeline, Trapd, Monitoring, Autoscaling (KEDA), Apache Pulsar, Event Sink, FCOM Processor, Trap Collector, Syslog, Syslog Collector, JavaScript Generic Event Processor

- Verification and Log Checks
- Extended Event Pipeline: JavaScript Generic Event Processor Setup
- Recap

Module 4: Troubleshooting Microservices

- Objectives
- Common Microservice Troubleshooting CLI Commands
- Node Issues

- describe nodes, describe node
- Pod Issues

get all pods, namespace pods, pod labels, list containers, logs, previous logs, shell, environment variables, delete pod

- Service Issues
- get services, describe service, get endpoints
- Resource Issues
- top nodes, top pods, describe pod
- Configuration Issues
- get all configmaps, get namespace configmaps, get YAML/JSON, get secrets
- Recap