Docker and Kubernetes – Course Outline

1 Duration

- 24 Hours

2 Objectives

At end of this workshop, participants will able to :

- Get understanding of Docker fundamentals, architecture, features and usage
- Get understanding of Kubernetes fundamentals, architecture, features and usage
- Containerize web applications / services using Docker and deploy into Kubernetes platform

Note: *This course is designed for beginner to intermediate level.*

3 Audience

Developers who are interested to learn how to containerize applications / services using Docker and manage the containers to handle scalability, fault tolerance, high availability using Kubernetes platform.

4 Pre-requisite

- Knowledge on Virtualization
- Knowledge on Distributed Computing
- Familiarity on Application Packaging and Deployment

5 Hardware & Network Requirements

- Desktop/Laptop with minimum 8GB RAM
- Open Internet connection (minimum 1 Mbps per user)

6 Software Requirements

- Windows / Linux / Mac OS
- Cloud Labs

7 Outline

Module-1: Introduction to Docker (12 hours)

- Why is Docker?
- What is Docker?
- What is Container?
- Virtual Machines vs Containers
- Benefits and Limitations of Docker
- Docker Architecture
    - Docker Client
    - Docker Server (Daemon)
- Docker Ecosystem
    - Docker Engine
    - Docker Registry
    - Docker Compose
    - Docker File
    - Image
    - Container
- Features Overview
    - Storage
    - Container Linking
    - Networking
- Docker Swarm Overview
- **Demo/Lab:** Verifying Docker Installation
- **Demo/Lab:** Pull and Run standard docker images
- **Demo/Lab:** Manage docker image and container life cycle
- **Demo/Lab:** Create Docker File for sample web application
- **Demo/Lab:** Build Docker Image for sample web application
- **Demo/Lab:** Run sample web application Docker Image locally
- **Demo/Lab:** Tag Docker Image build for sample web application
- **Demo/Lab:** Create DockerHub Account
- **Demo/Lab:** Upload (Push) Docker Image to DockerHub registry
- **Demo/Lab:** Download (Pull) Docker Image from DockerHub registry and run
- **Demo/Lab:** Store container data in the host file system using bind mount storage
- **Demo/Lab:** Store container data in the host file system using volume storage
- **Demo/Lab:** Store container data in the host system memory using tmpfs mount storage
- **Demo/Lab:** Link two containers and share data between them
- **Demo/Lab:** Create container networking with custom bridge network and share data between them
- **Demo/Lab:** Create and manage multi container applications using docker compose

Module-2: Introduction to Kubernetes (12 hours)

- Kubernetes Overview
- Kubernetes Architecture
- Kubernetes Setup and Configuration
- Components
    - Master Components
    - Node Components
    - Client Components
- Kubernetes Objects
- Kubernetes Containers
- Kubernetes Workloads
    - Pods
    - Deployments
    - Jobs
    - Replication
- Services and Load Balancing
- Storage Volumes
- Networking
- Creating and deploying an application in Kubernetes with Docker
- Configure Auto Scaling and High Availability
- Managing and accessing Kubernetes cluster with API and Kubectl
- Kubernetes Monitoring with Dashboard
- **Demo/Lab:** Verifying Kubernetes Installation
- **Demo/Lab:** Enable and access Kubernetes dashboard
- **Demo/Lab:** Create pod and deploy into K8s
- **Demo/Lab:** Create multi container pod and deploy into K8s
- **Demo/Lab:** Create deployment for sample web application with replication
- **Demo/Lab:** Create service to access the application internally
- **Demo/Lab:** Create service to access the application externally
- **Demo/Lab:** Create service to access the application with load balancing
- **Demo/Lab:** Store container data in the host file system with local path
- **Demo/Lab:** Store container data in the host file system with Persistent Volume Claim
- **Demo/Lab:** Verify load balancing and auto healing
- **Demo/Lab:** Create custom pod networking and share data between them
- **Demo/Lab:** Create and deploy sample application into K8s with auto scaling