Docker and Kubernetes

Duration
5 days

Audience
This workshop is intended for developers who need to understand Docker basics and start learning YML.

Course Objectives
After completing this workshop, the participants will be able to do

- Install Docker
- Work with Docker images
- Develop and Deploy small application using Docker
- Start working with YML
- Starting with Kubernetes

Pre-requisites
- Networking basics

Course Contents
- Shifting from Monolith to Micro Services
- Virtualization Vs Containerization
- Understand the architecture
  - What can I use Docker for?
  - What are the major Docker components?
  - What is Docker's architecture?
  - Inside Docker
  - What happens when you run a container?
  - The underlying technology
- Installing Docker
  - Installation from binaries
  - on Linux
  - on Windows
- Running a Docker Container
  - From the Docker quickstart terminal
  - Container port redirection
- Getting started with Docker Hub
  - Working with containers
  - Working with Docker images
  - Managing data in containers
  - Working with Docker Hub
- Creating docker image

- By modifying current container
- Using YML file

- Networking Docker images
  - Linking containers together
  - Working with default network
  - Creating your own bridge
- Volumes in Docker
  - Persistent / Shared Data
  - Mounting data volumes
  - Introduction to backup/restore/migrate data volumes
- Work with YML
  - image
  - build
  - dockerfile
  - command
  - extra_hosts
  - ports
  - links/external_links
- Overview of Docker Compose
  - Why docker compose
  - Installation and set-up
  - Create a Docker image
  - Define services
  - Build and run your app with Compose
  - Understanding networking in compose
- Container as a Service
  - Characteristics
  - Considerations
- Docker Implementation
  - On Premises
  - In the cloud
- Docker Clusters
  - Swarm
  - Kubernetes
  - Swarm Vs Kubernetes
- Kubernetes
  - What is Kubernetes?
  - Why Kubernetes?
  - Kubernetes Features
  - Kubernetes Architecture
  - Kubernetes Components
    - Etcd
    - API Server
    - Controller Manager Server

- Scheduler Server
- Minion Server Components

- Kubernetes for Container orchestration
  - Introduction and features
  - Kubernetes architecture and features
  - Kubernetes with container engine
  - Kubernetes Core Concepts
  - Minicube and kubectl tools
  - Using the VM Drivers
  - Start the basic cluster with minikube
  - The minikube commands overview
  - Basic objects
    - Pod
    - Service
    - Volume
    - Namespace
- The kubectl commands overview
  - The pod configuration in YML file
  - Create and manage pods with kubectl
  - The host network and host port mode
  - Run and monitor the pods with logs
  - Inspect the pods
  - Interact with pods
  - Pods with multiple containers: inspect, interact
  - Linking the containers in pod
  - Monitor and manage the cluster with dashboard
  - Minikube logging drivers
- Manage the container state with volume
  - The pods with volume
  - Sharing the data across containers with volume
  - Making the volume data persistent by mounting from host system.
  - Configure PersistentVolume and PersistentVolumeClaim objects
  - Using the PersistentVolumeClaim in the container
- The Controller objects
  - Deployment : to group multiple pod instances
  - Replication : To manage multiple replicas of pods
  - Statefulset: To manage the Stateful applications
  - DaemonSet : To manage the group of pods across hosts
  - Job :Grouping of pods
  - Create deployment
  - The deployment options
  - The labels and selectors in deployment
  - The service types

        - ClusterIP
        - NodePort
        - Loadbalancer
        - ExtrenalName
    - Expose the deployment with service
    - Service port mapping : fixed and dynamic
    - Inspect and log the service
    - Scale the service
    - Service Discovery
    - Default services and namespaces
    - Communication with outside world
    - The service load balancing
    - Ingress Controller for load balancing
- Kubernetes Cluster Administration and Monitoring
    - Monitoring the Kubernetes cluster
    - Kubernetes REST API for Control and analysis
    - Health Checks and liveness for pods/application
    - Resource Quotas
    - Working with Namespaces
    - User Management
- Kubernetes deployment on AWS