COURSE OUTLINE

Course Description

- The objective of the training is to help learners understand the role of container technology in packaging, shipping and running applications with container technologies like Docker/containerd and running containers at scale with orchestration solutions like Kubernetes.
- We help learners understand the changes in the SDLC lifecycle, DevOps practices when integrating a container based solution into the overall process.
- Learners are also made to understand the importance of microservices architecture, issues in developing/managing monolithic applications, common pitfalls in migrating from monolithic to microservices architectures and the challenges faced by dev & ops in managing such applications.
- We discuss case studies of how to build a Java/Python/Go based application that exposes a REST service. We first discuss how to build this using Docker and then in Kubernetes.
- Finally we also discuss some of the important developments in this space and what is to be expected in the next 6-12 months window.

Lab Setup

- Hardware requirements
  - Atleast 40GB of free disk space, 4GB RAM and Internet Connectivity
  - Operating System - Ubuntu 22.04 Desktop/Server Edition for Docker and Kubernetes
- Full network connectivity to the Internet.
- Lab VMs can be provided on a need basis – but will cost additional based on usage

Course Duration

- 5 days
  - Day 1 – Section and 2 – Overview of Distributed Systems, Cloud Computing, Containers and Orchestration
  - Day 2 – Section 3 to Section 6 – Docker and Containers deep dive
  - Day 3 – Section 7 to Section 10 – End to End Application with Docker and Kubernetes Deep Dive
  - Day 4 – Section 11 and Section 13 – Kubernetes Deep Dive
  - Day 5 – Section 14 to Section 16 – Microservices, Helm and ArgoCD

Pre-requisites

- Participants are expected to be familiar with a Linux environment and have at least some knowledge of virtualization and cloud technologies

Participant Profile

- Developers who will be moving a non-microservices application to a microservices architecture

Course Outline

Section 1 – Distributed Systems, Cloud Computing, Virtualization, Containers and Orchestration

- Distributed System design
- State and its importance in cloud architectures – handling operational issues of scalability and failure
- Overview of Virtualization technology
- Introduction to Cloud Computing
- Comparison of cloud v/s non-cloud deployments
- Cloud Deployment Models
- The Cloud Native Landscape
- Infrastructure As Code
- Role of containers and orchestration
- Avoiding vendor lock-in and achieving multi-cloud compatibility

Section 2 – Containerisation with Docker

- Installation of Docker – Linux/Windows/Mac
- Hello World - Running a simple BASH shell
- Docker images basics -- finding, listing, pulling and removing images
- Docker container basics - running, attaching, execing, listing, stopping/restarting containers
- Building images from containers

Section 3 – Working with custom Docker images

- Anatomy of a Dockerfile
- Building images from Dockerfiles

- Important instructions
- A few case studies
- Setting up a Docker registry
- Pushing images to the Docker registry

Section 4 – Introduction to CRI compatible runtimes

- Container lifecycle management and Container runtimes
- History and evolution of container runtimes
- The need for CRI specification
- Managing container lifecycle with containerd
- Use of nerdctl client

Section 5 – Networking in Docker (skim)

- Docker network types
- Multi-host networking (overview)
- Case studies

Section 6 – Managing state with Docker (skim)

- Working with data in containers
- Data volumes
- Case studies

Section 7 – End to end application building/deployment with Docker

- Stand-alone applications v/s Application Server based applications
- Analysing the sample 3-tier application
- Deploying the sample 3-tier application using Docker

Section 8 – Orchestration with Kubernetes

- How Kubernetes helps with orchestrating stateless and stateful components
- Comparison with Docker
- Introduction to declarative specifications
- Understanding the YAML file format
- Kubernetes architecture overview - master and workers

Section 9 – Installation of Kubernetes

- Use of kubeadm to setup a cluster
- Listing available nodes
- Troubleshooting cluster setup issues
- Dependency of Kubernetes on Docker/OCI runtimes
- Kubernetes Hello World
- A 50000 ft view of Kubernetes - running an nginx server and exposing it as a service
- Similarities between kubectl and docker commands

Section 10 – Containers and Pods

- Understanding the relationship between containers and pods
- Creating, listing and deleting pods via kubectl
- Interacting with pods - exec, log
- Running multiple containers in a pod
- Managing resources consumed by pods
- Node labels and selectors, affinity and anti-affinity

Section 11 – Deployments and ReplicaSets

- The role of ReplicaSets in managing pod replicas

- Deployments via Kubernetes
- Creating, listing and deleting Deployments via kubectl
- Understanding selectors
- How deployments work
- Deployment strategies

Section 12 – Kubernetes Networking, Services and Ingresses

- Multi-host Networking in Kubernetes (overview)
- Overview of Service discovery
- Creating, listing and deleting services via manifests
- Endpoint controller
- Listing endpoints of a service
- Communicating between multiple deployments
- Understanding how services work
- Virtual IPs and Service proxies
- Load balancer service
- Ingress
- Port forwarding during development
- Defining network policies

Section 13 – DevOps

- Software Development LifeCycle
- Devops toolchain
- Automating aspects of SDLC
- Continuous Integration/Continuous Delivery
- Types of Artifact Repositories – Language Specific v/s Container Registries when to use what
- Infrastructure as Code (IAC)

- Git feature branch workflow
- Blue green and Canary Deployments

Section 14 – Microservices Architecture

- Introduction to Microservices
- Comparing Microservices to Monolithic architectures
- Pros and cons of microservices architecture
- Complexities of microservice architectures
- Issue with debugging and development
- 12 factor application development
- Common pitfalls when migrating from Monolith to Microservices
- Microservices architectural patterns
    - Service discovery
    - Circuit breaker
    - Routing and proxies
    - Load balancing – internal and external
    - Ingress controllers
    - API gateway

Section 15 – Helm

- The need for Helm
- Installing Helm
- Installing charts
- Templating in Helm
- Creating your own charts

Section 16 – Continuous Delivery with ArgoCD

- Setting up ArgoCD
- Adding Git projects to Argo
- Working with Helm in Argo