Course Title: Dockers & Kubernetes (advanced)

Containerization and Kubernetes: Modern Application Deployment and Orchestration

Duration: 5 Day.

Course Description:

Learn about Application Modernisation using Cloud Native Design Principles by understanding containers and Orchestration of container-based workloads using Kubernetes. Overview of Containers, differences from virtualization and benefits of designing & developing loosely coupled, self-healing distributed applications which can dynamically scale based on real time workloads. Install Docker, understand its architecture, explore basic commands, build small & efficient Docker images, and launch containers. Understand container access and port forwarding techniques. Gain a deep understanding of Kubernetes, its architecture, and components. Overview of spinning up self-managed production grade Kubernetes clusters using various tools – Kubeadm, Kubespray, Rancher, Gardener etc, installation of single node Minikube cluster on Ubuntu 22.04 VM and in-depth hands-on sessions covering all aspects of container orchestration, management & observability.

Manage various workload Objects (Deployments, ReplicaSets, Jobs & CronJobs, DaemonSets) and scheduling in Kubernetes. Understand deployments, perform rolling updates and rollbacks. Use ConfigMaps and Secrets for secure application configuration. Scale applications within Kubernetes and learn about manifest management and templating tools.
Understand host networking configuration, connectivity between Pods, and different service types in Kubernetes (ClusterIP, NodePort, LoadBalancer). Use Ingress controllers and resources for external access to services. Configure CoreDNS within Kubernetes and choose a suitable container network interface plugin for effective networking.

Course Objective:

- Understand the fundamental concepts of containers and how they differ from virtualization.
- Learn about Docker and its role as a leading containerization platform.
- Explore various use cases for containers and the benefits they provide for software development and deployment.
- Gain hands-on experience by installing Docker and becoming familiar with basic Docker commands and flags for container management.
- Learn how to build Docker images using Dockerfiles and launch containers.
- Understand container access and port forwarding techniques to seamlessly interact with containers.

- Deeply comprehend the architecture of Kubernetes, a versatile container orchestration platform.
- Install and manage a basic Kubernetes cluster using Kubeadm.
- Explore different approaches to configure a Kubernetes cluster using single node Minikube cluster
- Acquire the skills to perform version upgrades on a Kubernetes cluster using Kubeadm.
- Understand the significance of Pods as a concept built on top of containers Kubernetes.
- Discover sidecar containers and init containers and their role in enhancing container functionality.
- Learn about deployments in Kubernetes and perform rolling updates and rollbacks effectively.
- Use ConfigMaps and Secrets to securely configure applications in Kubernetes.
- Gain insights into scaling applications within Kubernetes to meet changing demands.
- Understand how resource limits impact Pod scheduling in Kubernetes.
- Learn about manifest management and common templating tools used in Kubernetes.
- Understand services and networking in Kubernetes, including host networking configuration and connectivity between Pods.
- Configure different service types (ClusterIP, NodePort, LoadBalancer) in Kubernetes for routing traffic to application Pods.
- Utilize Ingress controllers and resources for managing external access to services in Kubernetes.
- Configure and utilize CoreDNS, a flexible and extensible DNS server, within Kubernetes.
- Understand the importance of choosing an appropriate container network interface plugin for effective networking in a Kubernetes environment.

Pre-Requisites:

1. Basic understanding of software development and deployment concepts.
2. Familiarity with Linux command line and basic system administration.
3. Knowledge of virtualization concepts and technologies.
4. Understanding of networking principles and protocols
5. Familiarity with containerization concepts (though not mandatory).
6. Basic understanding of cloud computing platforms (e.g., AWS, Azure, Google Cloud) is beneficial but not mandatory.
7. Access to a computer with administrative privileges to install software and perform hands-on exercises.
8. Willingness to learn and explore new technologies and concepts in the field of containers and container orchestration.

Course Contents:

Module 1: Introduction to Container and Docker

Day -1

- What are containers?
- Containerization vs Virtualization
- Container Use Cases
- Docker Installation on Ubuntu 22.04 VM
- Docker Architecture - Engine, Container, Docker host, Image, Registry
- Docker 101 commands, attaching & detaching, daemon containers, starting, stopping & removing containers
- Sample exercises for running various application containers – nginx, MySQL, tomcat, MongoDB etc
- Port forwarding & Port publishing, running various containers with popular open source images and connecting to published ports
- Docker logs & port forward, debugging containers by starting docker daemon in Debug mode

Day -2

- Building Docker Images, best practices for building small & secure images, Dockerfile detailed syntax - FROM, COPY, ADD, ENV, USER, WORKDIR, VOLUME, EXPOSE, RUN, CMD, ENTRYPOINT, .dockerignore
- Exercise to build real time web application facebooc (C with SQL Lite)Docker Image using two approaches and pushing them to DockerHub Registry
- Multi-Stage build exercise for a simple goLang image with reduced image size
- Concept of Private/Enterprise Registries, creating a local private registry and pushing newly built docker image to private registry – both Docker Registry2 and Sonatype Nexus OSS
- Concept of Docker Volumes, creating multiple docker volumes, sharing volumes across containers, read-only volumes
- Overview of Docker Networking, CNI/CNM concepts
- Popular container orchestration tools overview (Kubernetes, OpenShift etc.)
- Overview of Public Cloud offerings for container orchestration

Module 2: K8s Cluster Architecture, Installation & Configuration

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

DAY -3

- Kubernetes Architecture Overview – Control Plane components – kube-api-server, etcd, kube-scheduler, kube-controller-manager & Node components – kubelets & kube-proxy
- Overview of various popular solutions to deploy highly available Kubernetes clusters, both self-managed and public cloud managed service offerings – GKE, EKS & AKS
- Install single node Kubernetes cluster using Minikube & kubectl

- Pods, lifecycle of Pods, Single Container Pod v/s multi-container Pods, Sidecar Design Pattern, exercise to set-up multi-container Pod with volume sharing
- Need to Organize Pods, Labels, assigning Pods to specific Nodes using NodeSelector, Pod affinity & anti-affinity, Taints & Tolerations, exercises to set up Pods with some affinity or Toleration, check its status when it does not find any matching node for scheduling/execution

Module 3: Workloads & Scheduling

- Understand deployments and how to perform rolling update and rollbacks.
- Use ConfigMaps and Secrets to configure applications.
- Know how to scale applications.
- Understand the primitives used to create robust, self-healing, application deployments.
- Understand how resource limits can affect Pod scheduling.

Module 4: Organizing Pods, Health Probes, Rolling Updates, Services & Networking

Day-4

- Namespaces, organize Pods by running them in specific namespaces
- Health Checks & self-healing applications, Difference between Liveness & Readiness Probes, different types of Probes (httpGet, tcpSocket, exec) and arguments (initialDelaySeconds, periodSeconds, failureThreshold etc), exercise to set up probes for Pods and verify the restart in case of failures, best practices for setting up Application Probes
- Deep dive into release management using Deployment, setting up simple Tomcat deployment using RollingUpdate strategy, importance of maxSurge & maxUnavailable parameters, checking status of deployment using rollout and rolling back updates in case of any issues
- Services, need for having services, deep dive into types services – ClusterIP, NodePort, LoadBalancer and external (CNI/CNM), exercises to run a pod & expose it as Service, use case where to use NodePort
- Pod Networking, Introduction to Ingress, set up NGINX ingress for configuring two different Pods to different end points

Module 5: Storage, ConfigMaps & Secret, Pod Metrics, Horizontal Pod Autoscaling (HPA) & Monitoring

Day -5

- Introduction to PersistentVolume & PersistentVolumeClaim, exercise to define Pod Manifest which will mount a volume based on PVC that it binds to
- ConfigMaps & Secrets, need to inject run time configurations to Pods, exercise to set up ConfigMaps & Secrets and inject them into Pods as VolumeMounts and Environment Variables
- Advance Topics – resourceQuotas, assigning Pod resource (CPU & Memory) requests & limits, Pod monitoring using Metrics Server, exercise to check Pod resource usage using Metrics Server, Auto scaling in K8s (HPA & VPA)
- Helm Charts, demo to install Prometheus & Grafana helm charts and create a Monitoring Dashboard for cluster observability