Introduction to Docker containerization and Openshift for build team and administrators

Docker is an open-source tool that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating-system-level virtualization on Linux. The Docker supports independent "containers" to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines.

OpenShift is the containerization software developed by Red Hat.
The **OpenShift Container Platform** is an on-premises platform as a service built around Docker containers orchestrated and managed by Kubernetes engine.
OpenShift Container Platform formerly known as OpenShift Enterprise is Red Hat's on-premises private platform as a service product, built around a core of application containers powered by Docker, with orchestration and management provided by Kubernetes, on a foundation of Red Hat Enterprise Linux.

Objectives

This course provides the complete practical learning and hands-on experience to the team to start using docker containers for angular java web application deployments and the OpenShift platform to manage the docker containers at high level.

Prerequisite Knowledge

Participants must have

- A general understanding of virtualization concepts
- Have experience with Linux command line
- Experience with networking
- Understanding of the software development lifecycle
- Understanding of web application communication with dependent layers

Target Audience

Maximum 20 participants matching the prerequisites as above with every one working on a separate system having the set up as below.

Course Material

The course martial including presentations, sample exercises, documents and exercise case studies is shared in soft format.

Training Methodology

The theoretical topics are discussed interactively and technical details are demonstrated with practical examples. The participants work on the hands on exercises which strengthen the concepts learned.
Each topic is supplemented with practical demonstrations and hands on exercises for the participants.

Class Room Setup
The Intel dual/quad core compatible **CPU** with minimum 8 GB RAM and 500 GB HDD with Red Hat Enterprise Linux ver7.5 64 bit system with following software packages to be installed and configured.

1. Docker CE latest version and docker-compose for Linux.
2. Openshift cluster tools for Linux
3. Adobe Acrobat Reader for Linux
4. Oracle VirtualBox 5.1.12

Every participant must have a user account registered on Docker-Hub web site.
All the machines must be well connected in the LAN.
The participants must have admin rights on their machines.

Open live internet connection with reasonable speed and download permissions must be provided in the class room.

Training Duration: - 4 Days.

Instructor: Prakash Badhe.

Course plan

Day1
The VM and the Container

- Development and Deployment platforms
- Manage deployment across platforms
- Deployment in Virtual Machine environment
- The Virtual MC usage and limitations
- The Container architecture
- The container vs. virtual machine
- The isolated containers
- Manage the deployment environment with container
- Application dependencies in the container
- Making it highly available with clustering
- Scaling for performance
- Load balancing and instance management with orchestration
- Deployment monitoring and instance management.

Introduction to Docker Container Engine

- Docker architecture
- Installing the Docker Engine
- Docker daemon and docker client
- Docker containers for application deployments
- Docker commands overview
- Docker container and image
- Docker Hub registry for images
- Create and run first Docker container
- Monitor the docker container

Manage Docker containers and images

- Inspect docker process and containers
- Start, Stop and remove containers
- Interactive and detached containers
- Pull more docker images from Docker Hub
- Docker container Logging and monitoring
- Define and Build images with Dockerfile
- Create the images for angular-java web applications
- Testing the images with docker containers
- Access the application running in the container from outside
- Commit the docker container as image
- Tag the images
- Publish and share images with docker-hub
- Manage the docker images locally
- The Docker container life cycle

Docker container networking

- Docker networks
- Inspect the docker container properties
- Docker container isolation
- Default bridge network
- Container IP address
- Host network
- Port forwarding and mapping
- Create custom network
- Manage the docker networks
- Linking the docker containers such as UI and Rest service
- Execute commands on docker containers
- Connect to docker container through bash terminal
- Use private local Docker Registry

Day2

Docker container State management

- Docker container state across restarts

- Sharing the data across containers with volume
- Make the container state data persistent across container restarts with Volume
- Mount volumes to containers from host system
- Copy files/dirs from container to host and vice versa

Docker-compose usage

- Container configuration with YML files
- Configure and Deploy dependent containers
- Install and configure Docker-compose
- Docker-compose commands
- Start and stop the docker-compose containers
- Manage the volumes in docker-compose

More with Docker Containers

- Docker secrets
- Container resource utilization limits
- Docker container Logging and monitoring
- Health monitoring of docker containers

Day3

OpenShift Container Platform

- Overview of features
- Container orchestration
- Red Hat OpenShift Online
- The on-demand access build, deploy and manage scalable containerized applications
- The command line client tools
- The web management console UI

Docker Container Orchestration requirements

- High availability with Cluster
- The nodes in cluster
- Load balancing and instance management
- Deploy an application across multiple docker host nodes
- Application updates (rolling updates) and roll back
- Scaling the containers
- Monitor the containers
- Inspect the services
- Using logging drivers
- Docker swarm overview

The openshift container platform is built on top of Kubernetes to manage the containers. Hence it is imperative to understand the Kubernetes deployment objects in brief.

This topic introduces the Kubernetes cluster configuration and deployment objects.

Kubernetes for Container orchestration and Cluster

- Introduction and features
- Kubernetes architecture
- Kubernetes with container engine
- Kubernetes Core Concepts
- Kubernetes cluster tools
- Minikube overview
- The Kubeadm, kube-proxy with kubelet
- The kubectl for application deployment and monitoring
- Configure the cluster in local LAN.
- Basic objects
    - Pod
    - Service
    - Volume
    - Namespace
- The kubectl commands overview

Manage PODs in Kubernetes Cluster

- 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 and logs
- Linking the containers in pod
- Monitor and manage the cluster objects with dashboard
- Scale up/down the pods in the cluster

POD 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 POD

- Overview on custom volumes with storage drivers

Deploying services in the cluster

- Create deployment units
- The labels and selectors in deployment
- Expose the deployment units with service
- Service port mapping-fixed, dynamic
- Service types
    - ClusterIP
    - NodePort
    - Loadbalancer
    - ExtrenalName
- Inspect and log the service
- Scale the service
- Service Discovery
- Default services and namespaces
- Communication with outside world
- The service updates(rolling update) and rollbacks
- Dashboard to monitor the cluster objects
- Log management and troubleshooting the cluster
- Rolling update of applications

Day4

OpenShift Container Platform

- Overview of features
- Container orchestration
- Red Hat OpenShift Online
- The on-demand access build, deploy and manage scalable containerized applications
- Sign up to use OpenShift Online service
- The command line client and web console
- The access from Eclipse IDE

The Cluster Administration and Monitoring

- Create a project
- Deploy docker image
- Manage pods from console
- Scaling the containers
- Resource usage monitoring
- Auto provisioning and self healing
- Working with Namespaces

- Scheduling the apps and pods
- Configure storage and volumes

Service Objects in OpenShift platform

- Deploy services as wrapper around deployment units
- Configure service scaling
- OpenShift router to allow access
- Create new route
- Monitor the routes

Explore Openshift with Minishift

- Download and install
- The minishift and OpenShift client tools
- Eclipse IDE using the JBoss tools.
- Start the Minishift with virtual driver
- Deploy sample application in minishift
- The web console
- Expose a route to the service
- Access the application
- Monitor the openshift origin containers and pods

****